ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Διδακτορική διατριβή του Υ/Δ του τμήματος

Μηχανικών Παραγωγής και Διοίκησης

Ματθαίου Πρωτόπαπα

με θέμα

«Γενετικοί Αλγόριθμοι στο παίγνιο Cournot»

Επιβλέπων: Επικ. Καθηγητής Ηλίας Κοσματόπουλος

ΠΕΡΙΕΧΟΜΕΝΑ

Ευχαριστίες

Η παρούσα διατριβή δεν θα ήταν δυνατόν να έχει γραφτεί χωρίς την αμέριστη υποστήριξη, την πολύτιμη βοήθεια και την ασίγαστη συνεργασία από την πλευρά του επιβλέποντα Επίκουρου Καθηγητή Ηλία Κοσματόπουλου, στον οποίο θα ήθελα να απευθύνω τις θερμότερες ευχαριστίες μου. Ο κ. Κοσματόπουλος αποτελεί, με την αψεγάδιαστη συμπεριφορά του, παράδειγμα καθηγητή προς μίμηση και αποδεικνύει σε κάθε ευκαιρία ότι, εκτός από λαμπρός επιστήμονας είναι και ένας εξαιρετικός άνθρωπος και φίλος.

Θα ήθελα να ευχαριστήσω επίσης θερμά τον προϊστάμενό μου στο τμήμα Στατιστικής του Πανεπιστημίου “La Sapienza” της Ρώμης, Prof. Francesco Battaglia, για την κατανόηση που έδειξε για τις ανάγκες της παρούσας έρευνας και τις πολύτιμες παρατηρήσεις που έκανε κατά τη διάρκειά της. Θα ήθελα επί της ευκαιρίας να ευχαριστήσω και όλους εκείνους που απαρτίζουν την ακαδημαϊκή κοινότητα του δικτύου ευρωπαϊκών πανεπιστημίων COMISEF και ιδιαίτερα τον συντονιστή του δικτύου Prof. Peter Winker, για την υψηλών προδιαγραφών υποστήριξη που παρέχουν σε όλους τους ερευνητές που συμμετέχουν στο πρόγραμμα Marie Curie του δικτύου, με υψηλού επιπέδου σεμινάρια, workshops, εξειδικευμένα μαθήματα αγγλικών πάνω στην συγγραφή ακαδημαϊκού χαρακτήρα, διαλέξεις και συνέδρια, η οποία αποτελεί αναμφίβολα ένα σημαντικότατο εφόδιο στην καριέρα του κάθε ερευνητή.

Ευχαριστώ επίσης τους καθηγητές που συμμετείχαν στην τριμελή επιτροπή αξιολόγησης της ερευνητικής μου δουλειάς, κ.κ. Μάρκο Παπαγεωργίου και Αθανάσιο Μυγδαλά, για τις υποδείξεις και τις διορθώσεις που έκαναν στην παρούσα διατριβή, καθώς και τα υπόλοιπα μέλη της επταμελούς επιτροπής κ.κ. Εμμανουήλ Πετράκη, Νικόλαο Ματσατσίνη, Νικόλαο Βλάσση και Νικόλαο Τσουρβελούδη - ο οποίος αντικατέστησε επίσης τον κ. Μυγδαλά στην τριμελή επιτροπή- που είχαν την καλοσύνη να αξιολογήσουν το παρόν έργο.

Τέλος θα ήθελα να ευχαριστήσω όλα τα μέλη του εργαστηρίου Δυναμικών Συστημάτων και Προσομοίωσης του τμήματος Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης, καθώς και τους συνεργάτες μου στο Πανεπιστήμιο της Ρώμης, για την πολύ δημιουργική συνεργασία και την φιλική ατμόσφαιρα στην οποία έχω την ευχαρίστηση να εργάζομαι.

Η παρούσα έρευνα χρηματοδοτήθηκε από την Ευρωπαϊκή Ένωση, μέσω του δικτύου COMISEF (www.comisef.eu).

Abstract

We study the Agent Computational Economics' problem of simulating agents' behavior in a Cournot oligopoly model, by the use of co-evolutionary learning Genetic Algorithms. We try to discover algorithms of that kind, which conclude to the Nash Equilibrium outcome, and hence, can be used as heuristics for discovering the Nash Equilibrium quantities, as well. In order to quantify the difference between a given state of the co-evolutionary genetic algorithms and the goal outcome, we introduce a measure that is defined on the lumped states of the corresponding Markov Chain. We finally introduce an optimization algorithm that is based on the convergence of sequential best replies, and uses a genetic algorithm to identify the best reply at any given situation, which convergences to the Nash Equilibrium, under the aforementioned requirement.

Περίληψη

Στην διατριβή αυτή μελετάται η προσομοίωση της συμπεριφοράς των παικτών μιας ολιγοπωλιακής αγοράς Cournot, μέσω συν-εξελικτικών, γενετικών αλγορίθμων μάθησης. Γίνεται προσπάθεια για την εύρεση αλγορίθμων που θα επιτυγχάνουν τη σύγκλιση της παρατηρούμενης συμπεριφοράς των πρακτόρων στην ισορροπία κατά Nash, οπότε θα μπορούν να χρησιμοποιηθούν και σαν ευρετικοί αλγόριθμοι προσδιορισμού της ισορροπίας. Για την εκτίμηση της απόκλισης των πληθυσμών των αλγορίθμων από την επιθυμητή κατάσταση, εισάγουμε ένα «μέτρο» που βασίζεται στην έννοια των ομαδικών καταστάσεων των αλυσίδων Markov. Εισάγεται τέλος και ένας αλγόριθμος βελτιστοποίησης που, εκμεταλλευόμενος την σύγκλιση της θεωρητικής ακολουθίας βέλτιστων απαντήσεων, χρησιμοποιεί έναν γενετικό αλγόριθμο για να βρει την βέλτιστη απάντηση του κάθε παίκτη και οδηγείται έτσι στην σύγκλιση στην ισορροπία κατά Nash.

1. Εισαγωγή

Στην μελέτη αυτή θα αναλύσουμε την εφαρμογή γενετικών αλγορίθμων στο παίγνιο που εισήχθη το 1838 από τον Γάλλο μηχανικό και οικονομολόγο A.A.Cournot (Cournot 1838) και το οποίο ενέπνευσε τον Nash στην ανάπτυξη της θεωρίας παιγνίων μη-μηδενικού αθροίσματος (Nash 1950a, 1950b) . Το παίγνιο αυτό περιγράφει μια ολιγοπωλιακή αγορά με ορισμένο αριθμό επιχειρήσεων, οι οποίες προσδιορίζουν την ποσότητα (και μόνο) που θα παράγουν, με σκοπό βέβαια την μεγιστοποίηση του ιδίου τους οφέλους. Η τιμή στην αγορά προσδιορίζεται από την τομή των καμπυλών ζήτησης (εξωγενής) και προσφοράς, η οποία αποτελείται από το σύνολο των ποσοτήτων που προσφέρουν οι επιχειρήσεις.

Η ποσότητα ισορροπίας που υπολόγισε ο Cournot για το παίγνιο αυτό, δεν είναι άλλη από την ποσότητα της ισορροπίας κατά Nash σε καθαρές στρατηγικές. Στην ανάλυσή του, ο Γάλλος οικονομολόγος υπέθεσε ότι οι επιχειρήσεις επιλέγουν καθορισμένες ποσότητες κάθε φορά, εξού και η ισορροπία σε καθαρές στρατηγικές. Η ανάλυση του Cournot ενέπνευσε τον Nash στην διατύπωση του περιώνυμου θεωρήματός του για την ύπαρξη ισορροπίας (σε μεικτές εν γένει) στρατηγικές σε οποιοδήποτε παίγνιο μη-μηδενικού αθροίσματος με πεπερασμένο αριθμό καθαρών στρατηγικών και επίσης το παίγνιο Cournot ήταν από τα πρώτα στα οποία εφαρμόστηκε η θεωρία αυτή του Nash.

Η συμβολή της έρευνάς μας έγκειται στο γεγονός ότι αναλύσαμε με χρήση της θεωρίας των αλυσίδων Markov τις κυριότερες από τις ήδη υπάρχουσες μελέτες συν-εξελικτικών γενετικών αλγορίθμων για το παίγνιο Cournot και αφού διαπιστώσαμε ότι οι πιθανότητες της στάσιμης κατάστασης (πρόκειται για εργόδικες αλυσίδες Markov) των ομαδοποιημένων καταστάσεων των οποίων ορίσαμε με χρήση της απόστασης Hamming είναι τέτοιες ώστε να μην συνάγεται το συμπέρασμα της σύγκλισης των αλγορίθμων αυτών στην ισορροπία κατά Nash, τους τροποποιήσαμε ώστε, με την εισαγωγή κοινωνικής μάθησης, να έχουμε την σύγκλιση (με την στοχαστική πάντα έννοια) των στρατηγικών των παικτών στην ισορροπία κατά Nash σε καθαρές στρατηγικές. Δεδομένου ότι οι αλγόριθμοι κοινωνικής μάθησης που εισάγουμε, απαιτούν την ύπαρξη συμμετρικών συναρτήσεων κόστους, κρίναμε αναγκαία και την δημιουργία ενός επαναληπτικού αλγόριθμου που βασίζεται στην σύγκλιση της αλληλουχίας των διαδοχικών βέλτιστων απαντήσεων και στους γενετικούς αλγόριθμους, ώστε να επιτύχουμε την σύγκλιση και στην περίπτωση μη συμμετρικών παιγνίων Cournot στην ισορροπία κατά Nash σε καθαρές στρατηγικές. Ο επαναληπτικός αλγόριθμος που εισάγουμε δεν έχει το πρόβλημα άλλων επαναληπτικών αλγορίθμων που συγκλίνουν –λανθασμένα- σε τοπικά ελάχιστα χωρίς να βρίσκουν την ισορροπία κατά Nash. Τέλος δίνουμε οδηγίες για τον τρόπο με τον οποίο μπορούν οι αλγόριθμοι αυτοί να χρησιμοποιηθούν για την εύρεση της ισορροπίας κατά Nash σε καθαρές στρατηγικές, σε παίγνια στα οποία η ισορροπία κατά Nash δεν είναι γνωστή.

Στην 2η ενότητα του 2ου κεφαλαίου θα κάνουμε μια εκτεταμένη παρουσίαση του παιγνίου Cournot και της ισορροπίας κατά Nash, των διαφόρων μορφών του, καθώς και μια επισκόπηση των συνθηκών εκείνων που επιτρέπουν την ύπαρξη ισορροπίας σε καθαρές στρατηγικές.

Στην 3η υπο-ενότητα της 2ης ενότητας του κεφαλαίου 2, θα αναλύσουμε την σύγκλιση των ακολουθιών διαδοχικών απαντήσεων στην ισορροπία αυτή. Εφόσον θεωρήσουμε ότι έχουμε μια δεδομένη κατάσταση στην αγορά (ένα δεδομένο σύνολο ποσοτήτων που έχουν επιλεγεί από την κάθε επιχείρηση), μπορούμε να διερευνήσουμε την εξέλιξη της κατάστασης αυτής, καθώς ο κάθε παίκτης -διαδοχικά ή ταυτόχρονα- επιλέγει τη βέλτιστη ποσότητα, με δεδομένη την κατάσταση αυτή. Καθώς δηλαδή προσδιορίζει τη βέλτιστη απάντησή του στις -δεδομένες- επιλογές των άλλων παικτών. Οι θεωρίες αυτές θα μας φανούν πολύ χρήσιμες στο κεφάλαιο 7, όπου θα εισάγουμε έναν αλγόριθμο, που θα βασίζεται σε γενετικούς αλγόριθμους βελτιστοποίησης, για να βρει αυτήν τη βέλτιστη απάντηση.

Στο μεγαλύτερο μέρος της μελέτης μας αυτής, πάντως, θα επικεντρωθούμε στους συν-εξελικτικούς γενετικούς αλγορίθμους μάθησης. Οι γενετικοί αλγόριθμοι αυτοί διαχωρίζονται από τους κλασικούς γενετικούς αλγόριθμους βελτιστοποίησης στο ότι η συνάρτηση προσαρμοστικότητας, εξαρτάται κάθε φορά, από την κατάσταση στην οποία βρίσκεται ο πληθυσμός του αλγορίθμου. Αντί δηλαδή να αξιολογείται το κάθε χρωμόσωμα του αλγορίθμου με βάση την καταλληλότητά του σε σχέση με μια δεδομένη και εξωγενώς καθοριζόμενη συνάρτηση (μια συνάρτηση προς ελαχιστοποίηση λ.χ.), η προσαρμοστικότητα του κάθε χρωμοσώματος εξαρτάται από τα άλλα χρωμοσώματα που βρίσκονται στον πληθυσμό (ή στους πληθυσμούς, αφού θα χρησιμοποιήσουμε και αλγόριθμους με πολλούς πληθυσμούς- έναν για κάθε παίκτη). Οι αλγόριθμοι αυτού του είδους παρουσιάζονται εκτεταμένα στην 4η ενότητα του κεφαλαίου 2.

Η χρήση αυτού του είδους των αλγορίθμων υποδηλώνει ότι αντιμετωπίζουμε το πρόβλημα σαν πρόβλημα μάθησης. Οι παίκτες δηλαδή προσδιορίζουν την στρατηγική τους, η οποία καθορίζει την κατάσταση στην αγορά και στην συνέχεια η κατάσταση αυτή (η τιμή πιο συγκεκριμένα) καθοδηγεί τους παίκτες στο να αλλάξουν την συμπεριφορά τους. Η σχετική συζήτηση αναπτύσσεται και αυτή στις τελευταίες ενότητες του 2ου κεφαλαίου, ενώ στην 1η ενότητα του κεφαλαίου γίνεται μια σύντομη αναφορά στον ορισμό της έννοιας της Υπολογιστικής Οικονομικής Πολλαπλών Πρακτόρων (Agent Computational Economics - ACE). Δεδομένου ότι η προσέγγισή μας χρησιμοποιεί αλγόριθμους μάθησης και πράκτορες που μαθαίνουν μέσω αυτών των αλγορίθμων, εμπίπτει σαφώς στην γενικότερη αυτή κατηγορία.

Η έννοια των εξελικτικά σταθερών στρατηγικών, σχετίζεται άμεσα με το θέμα μας. Η έννοια αυτή, αναπτύχθηκε ειδικά για την εξελικτική θεωρία παιγνίων, για καταστάσεις δηλαδή σαν αυτή που περιγράφουμε εδώ, για να περιγράψει τις στρατηγικές εκείνες που αναμένεται να επικρατήσουν, μέσω της εξελικτικής διαδικασίας μάθησης. Στις διαδικασίες εξελικτικής μάθησης, οι παίκτες δεν μεγιστοποιούν per se, αλλά μαθαίνουν κατά τη διάρκεια των διαπραγματεύσεων. Συνεπώς η κλασική αιτιολόγηση ενός αποτελέσματος της θεωρίας παιγνίων, ότι όλοι οι παίκτες είναι «έξυπνοι» και κατανοούν ότι και οι αντίπαλοί τους είναι το ίδιο, ότι δηλαδή μεγιστοποιούν το αναμενόμενο κέρδος τους με την παραδοχή ότι και οι συμπαίκτες τους επιλέγουν την στρατηγική εκείνη που μεγιστοποιεί το δικό τους, δεν έχει θέση εδώ. Η αιτιολόγηση της ισορροπίας πρέπει να προκύψει από τις ιδιότητες της διαδικασίας μάθησης και γι αυτό εισήχθη η έννοια των εξελικτικά σταθερών στρατηγικών οι οποίες, εκτός του ότι είναι ισορροπίες κατά Nash του παιγνίου, έχουν το πρόσθετο πλεονέκτημα, ότι δεν μπορούν -μέσω κάποιας συγκεκριμένης- εξελικτικής διαδικασίας να «εκτοπιστούν» από τον πληθυσμό, από άλλες «μεταλλαγμένες» στρατηγικές. Αργότερα διαπιστώθηκε ότι -για πεπερασμένους πληθυσμούς στρατηγικών, όπως συμβαίνει στους γενετικούς αλγορίθμους- υπάρχει η δυνατότητα για μια στρατηγική που δεν αποτελεί ισορροπία κατά Nash να έχει την ιδιότητα της ευστάθειας, κάτι που οδήγησε σε αναθεώρηση του αρχικού ορισμού για τις εξελικτικά ευσταθείς στρατηγικές σε διαδικασίες μάθησης με πεπερασμένους πληθυσμούς στρατηγικών. Η συζήτηση για τις εξελικτικά ευσταθείς στρατηγικές αναπτύσσεται στις ενότητες 2.2.4, 2.3 και στην παρουσίαση των εργασιών των Vega-Redondo (1997), Vriend (2000), Alos-Ferrer και Ania (2005) και Valee και Yildizoglou (2007) στο κεφάλαιο 3.

Από τα παραπάνω γίνεται φανερό ότι το σημαντικότατο ερώτημα που ανακύπτει είναι εάν ένας δεδομένος γενετικός αλγόριθμος μάθησης, επιτρέπει στους πράκτορες να «μάθουν» την σωστή στρατηγική (με τον όρο σωστή εννοούμε την στρατηγική κατά Nash) και συνεπώς τη σύγκλιση της αγοράς στην κατάσταση που προβλέπεται από την θεωρία παιγνίων μη-μηδενικού αθροίσματος. Όπως θα δούμε στο κεφάλαιο 3, στο οποίο παρουσιάζουμε τα πρότερα ευρήματα άλλων ερευνητών, οι περισσότεροι γενετικοί αλγόριθμοι μάθησης που έχουν εφαρμοστεί στο πρόβλημα μέχρι στιγμής, δεν καταφέρνουν κάτι τέτοιο. Αντίθετα, οι πράκτορες που λειτουργούν με βάση αυτούς τους αλγόριθμους, «μαθαίνουν» τελικά μια άλλη στρατηγική, η οποία είναι κατάλληλη για την περιγραφή ανταγωνιστικών και όχι ολιγοπωλιακών αγορών. Μιλάμε βέβαια για την ισορροπία Walras (ή ανταγωνιστική ισορροπία) στην οποία η κάθε επιχείρηση επιδιώκει την μεγιστοποίηση του κέρδους της, θεωρώντας την τιμή στην αγορά, δεδομένη. Στην περίπτωση μάλιστα γενετικών αλγορίθμων με έναν κοινό πληθυσμό για όλους τους παίκτες, στον οποίο το κάθε χρωμόσωμα αντιστοιχεί στην στρατηγική επιλογή μιας συγκεκριμένης επιχείρησης οι μελέτες των Vega-Redondo (1997) και Alos-Ferrer και Ania (2005) δίνουν και μια θεωρητική τεκμηρίωση της σύγκλισης στην ισορροπία αυτή, για μια αρκετά γενική κατηγορία εξελικτικών αλγορίθμων.

Στο κεφάλαιο αυτό θα αναφερθούμε περιορισμένα και σε μια μελέτη που αφορά την εφαρμογή νευρωνικού δικτύου στο πρόβλημα. Το νευρωνικό δίκτυο μοντελοποιεί το «περιβάλλον» μέσα στο οποίο καλείται να αποφασίσει ο πράκτορας, οπότε αντί να «μαθαίνει» εξελικτικά ο ένας πράκτορας από τη συμπεριφορά του άλλου, προσπαθούν όλοι να εκτιμήσουν τη μη γραμμική συνάρτηση που συνδέει την κατάσταση στην αγορά με το κέρδος τους. Όπως έδειξαν οι Barr και Saraceno (2005) που έκαναν την συγκεκριμένη μελέτη, η προσέγγιση του νευρωνικού δικτύου ήταν επιτυχής στην ανακάλυψη και στη σύγκλιση της συμπεριφοράς των πρακτόρων στην ισορροπία κατά Nash.

Οι μοναδικοί γενετικοί αλγόριθμοι που οδήγησαν στην σύγκλιση της μέσης ποσότητας, και συνεπώς της ποσότητας και της τιμής στην αγορά, στις τιμές που προβλέπει η ισορροπία κατά Nash, ήταν ο αλγόριθμος πολλών πληθυσμών και ατομικής μάθησης του Vriend (2000) και ο αλγόριθμος κοινωνικής μάθησης (και ενός πληθυσμού, δεδομένου ότι δεν έχει χρησιμοποιηθεί ακόμα γενετικός αλγόριθμος πολλών πληθυσμών και κοινωνικής μάθησης, στο παίγνιο) των Alkemade et al. (2007). Στο κεφάλαιο 5, θα εκτελέσουμε μια σειρά προσομοιώσεων των δύο αυτών αλγορίθμων, μαζί με τον πρώτο γενετικό αλγόριθμο μάθησης που χρησιμοποιήθηκε στο πρόβλημα Cournot (Arifovic 1994) -ο οποίος όμως συγκλίνει στην ισορροπία κατά Walras- καθώς και του αλγορίθμου του co-evolutionary programming, που χρησιμοποιήθηκε με ανάλογα αποτελέσματα με εκείνα των δύο πρώτων αλγορίθμων (σύγκλιση της μέσης ποσότητας, της τιμής και της ποσότητας στην αγορά στην ισορροπία κατά Nash) σε ένα παρόμοιο παίγνιο αγορών ηλεκτρικής ενέργειας (Price 1997), σε μια σειρά υποδειγμάτων Cournot με ισορροπία σε καθαρές στρατηγικές, ώστε να διερευνήσουμε τι πραγματικά συμβαίνει με τη διαδικασία μάθησης καθεαυτή, το πώς δηλαδή εξελίσσονται και το πού συγκλίνουν οι ποσότητες που επιλέγουν οι μεμονωμένοι παίκτες. Για να καταφέρουμε να έχουμε μια πιο πλήρη εικόνα της κατάστασης, εισάγουμε ένα τρόπο «μέτρησης» της απόκλισης των καταστάσεων στις οποίες βρίσκεται το σύστημα κάθε στιγμή, που βασίζεται στην έννοια των ομαδικών καταστάσεων (lumped states- Kemeny and Snell 1960), της αλυσίδας Markov που περιγράφει την στοχαστική εξέλιξη των αλγορίθμων (και των γενετικών αλγορίθμων μάθησης γενικότερα - περισσότερα για τις ιδιότητες των γενετικών αλγορίθμων μάθησης σαν αλυσίδες Markov, στη σχετική υποενότητα του κεφαλαίου 2, στην σχετική ενότητα του κεφαλαίου 5, καθώς και σε κάποιες από τις παρουσιάσεις των πρότερων ερευνητικών ευρημάτων του κεφαλαίου 3).

Η διαπίστωση της μη σύγκλισης των μεμονωμένων ποσοτήτων στην ποσότητα που προβλέπει η θεωρία παιγνίων μη-μηδενικού αθροίσματος και συνεπώς, η μη σύγκλιση της τεχνητής αγοράς στην ισορροπία κατά Nash, στην περίπτωση των αλγορίθμων του Vriend (2007) και του co-evolutionary programming (Price 1997) μας οδηγεί -στο κεφάλαιο 6- στην τροποποίησή τους ώστε να είναι εφικτή η θεώρηση των συμπεριφορών των συμπαικτών κατά τη διάρκεια διαμόρφωσης των εναλλακτικών του κάθε συγκεκριμένου παίκτη (κοινωνική μάθηση), εισάγοντας έτσι τους πρώτους αλγόριθμους κοινωνικής μάθησης πολλών πληθυσμών, που εφαρμόστηκαν στην ανάλυση του παιγνίου Cournot. Οι τροποποιήσεις που προτείνουμε στο κεφάλαιο 6, όχι μόνο οδηγούν στην σύγκλιση των αναμενόμενων ποσοτήτων των μεμονωμένων παικτών στην λύση Nash, αλλά και στη συχνή σύγκλιση ολόκληρου του συνόλου των χρωμοσωμάτων των διαφόρων πληθυσμών, στην ποσότητα αυτή, ενώ τέλος σε πολλά από τα παίγνια που πραγματώνονται κατά τη διάρκεια της προσομοίωσης έχουμε ισορροπία κατά Nash. Όλα αυτά τα αποτελέσματα ήταν αδύνατα στην αρχική έκδοση των αλγορίθμων αυτών.

Δυστυχώς η «κοινωνικοποίηση» των αλγορίθμων ατομικής μάθησης προϋποθέτει την συμμετρία των συναρτήσεων κέρδους των παικτών ή ισοδύναμα τη συμμετρία των συναρτήσεων κόστους, αφού στο συγκεκριμένο παίγνιο η συνάρτηση κέρδους εξαρτάται αποκλειστικά από την τιμή στην αγορά -που εξ΄ ορισμού είναι κοινή- κι από το κόστος του κάθε παίκτη. Για την περίπτωση μη συμμετρικών παιγνίων καταφεύγουμε -όπως προαναφέραμε- στην έννοια της σύγκλισης των διαδοχικών απαντήσεων. Στο κεφάλαιο 7 εισάγουμε έναν αλγόριθμο που προσδιορίζει τις βέλτιστες απαντήσεις του κάθε παίκτη ευρετικά, μέσω ενός γενετικού αλγορίθμου, και αναλύοντας τις ιδιότητες της αλυσίδας Markov που τον χαρακτηρίζει, αλλά και μέσω προσομοιώσεων τόσο σε παίγνια Cournot με συγκλίνουσα ακολουθία διαδοχικών βέλτιστων απαντήσεων όσο και σε παίγνια με αποκλίνουσα τέτοια ακολουθία, προσπαθούμε να διαπιστώσουμε την επιτυχία ή μη του αλγόριθμου αυτού. Απ΄ ότι διαπιστώνουμε, πράγματι μπορούμε να βρούμε την ισορροπία κατά Nash, βελτιώνοντας παράλληλα την ακρίβεια των σημαντικών ψηφίων, με τη χρήση κατάλληλου εξωτερικού βρόχου, με την προϋπόθεση ότι η θεωρητική ακολουθία συγκλίνει. Αυτό συμβαίνει ακόμα και σε παίγνια με συναρτήσεις κέρδους με τοπικά ελάχιστα, στα οποία παρόμοιοι αλγόριθμοι που βασίζονται σε μεθόδους τοπικής αναζήτησης (Son and Baldacci 2004, Weber and Over bye 1999, Hobbs et al. 2000), αποτυγχάνουν.

2. Θεωρητικό Υπόβαθρο

2.1. Τι είναι η Υπολογιστική Οικονομική Πολλαπλών Πρακτόρων (Agent Computational Economics)

Οι οικονομίες της αγοράς αποτελούνται από έναν μεγάλο αριθμό «πρακτόρων» που αλληλεπιδρούν μεταξύ τους τοπικά, δηλαδή ο καθένας από αυτούς αλληλεπιδρά με ένα περιορισμένο υποσύνολο άλλων πρακτόρων. Αυτές οι τοπικές αλληλεπιδράσεις, αν εξεταστούν μακροσκοπικά, αποκαλύπτεται ότι οδηγούν σε μακροοικονομικές «κανονικότητες», όπως συστήματα συναλλαγών, χρηματοοικονομικές δομές (νόμισμα-τράπεζες), συγκεκριμένες «εθιμοτυπικές» συμπεριφορές όσον αφορά τις συναλλαγές (για παράδειγμα σε λιγότερο ανεπτυγμένες οικονομίες παρατηρείται το φαινόμενο του «παζαριού», κάτι που εκλείπει σε πιο ανεπτυγμένες αγορές λιανικής), τεχνολογικές μεταβολές κλπ. που με τη σειρά τους επηρρεάζουν τη δομή και την αλληλεπιδραστική λειουργία των ίδιων των πρακτόρων. Το αποτέλεσμα είναι ένα πολύπλοκο δυναμικό σύστημα με τις αιτίες και τα αιτιατά να αλλάζουν ρόλους διαχρονικά, σε μια αέναη μεταβολή όλων των στοιχείων του συστήματος.

Αυτή η αμφίδρομη αλληλεπίδραση μεταξύ των μικρότερων συστατικών ενός οικονομικού συστήματος και των μακροοικονομικών του χαρακτηριστικών είναι ευρύτατα γνωστή [1,2,3,4]. Ωστόσο τα μέσα για την μοντελοποίηση αυτών των αλληλεπιδράσεων ήταν ανύπαρκτα μέχρι την έλευση των Η/Υ, λόγω της τεράστιας πολυπλοκότητας του όλου εγχειρήματος. Στην «κλασική» οικονομική ανάλυση, τα μοντέλα βασίζονται σε έξωθεν κανόνες αναφορικά με τον τρόπο λήψης αποφάσεων (υποτίθεται ότι όλοι μεγιστοποιούν το ίδιον όφελος για παράδειγμα), με την πληροφόρηση (ότι για παράδειγμα όλες οι πληροφορίες είναι διαθέσιμες σε όλους), με τη γενικότερη συμπεριφορά των πρακτόρων (κατά κανόνα λαμβάνεται μια συγκεκριμένη συμπεριφορά σαν ενδεικτική της συμπερίφορας των μελών μιας ομάδας-έχουμε δηλαδή αυτό που λέμε «αντιπροσωπευτικό πράκτορα») και τέλος υποθέσεις σχετικά με την επίτευξη ισορροπίας στις αγορές. Προφανώς όλες αυτές οι υποθέσεις ισχύουν μόνο προσεγγιστικά, κάτι που υποβαθμίζει την ισχύ των οικονομικών μοντέλων. Από την άλλη όμως, τέτοιες υποθέσεις είναι απαραίτητες για να είναι τα μοντέλα επιλύσιμα με σχετικά απλές -αναλυτικές μεθόδους. Τα τελευταία χρόνια ωστόσο, η ανάπτυξη των Η/Υ έδωσε τη δυνατότητα ανάπτυξης νέων εργαλειών για να «διεισδύσουμε» στο εσωτερικό των τοπικών αλληλεπιδράσεων των επιμέρους οικονομικών πρακτόρων [5,6,7,8,9,10,11,12].

Σε τελική ανάλυση η Υπολογιστική Οικονομική Πολλαπλών Πρακτόρων (Agent Computational Economics- ACE) είναι ακριβώς η δημιουργία μοντέλων δικτύων πρακτόρων στον Η/Υ και η εκτέλεση προσομοιώσεων, ουτωσώστε να γίνει περισσότερο κατανοητή η λειτουργία των πραγματικών δικτύων που συνδέουν καταναλωτές, εργαζόμενους, επιχειρήσεις και θεσμούς σε μια πραγματική οικονομία. Ο σκοπός είναι διπλός: αφενός να κατανοήσουμε και να περιγράψουμε το γιατί παρατηρούνται συγκεκριμένες δομές στις οικονομικές συμπεριφορές και αφετέρου να σχεδιάσουμε νέους αποτελεσματικούς μηχανισμούς, που να επηρρεάζουν την οικονομία αποτελεσματικότερα αφού θα λαμβάνεται υπόψη η επιρροή τους στις συμπεριφορές των μεμονομένων πρακτόρων και το αντίστροφο. Τα πειραματικά δεδομένα δε, συμφωνούν σε μεγαλύτερο βαθμό με τα μοντέλα Υπολογιστικής Οικονομικής Πολλαπλών Πρακτόρων (Agent Computational Economics), παρά με τα κλασσικά μοντέλα [13].

2.2. Το παίγνιο Cournot

2.2.1. Το παίγνιο Cournot και η ισορροπία κατά Nash

Το παίγνιο Cournot, δεν είναι άλλο από το μοντέλο ολιγοπωλίου που παρουσίασε ο Cournot (1838), σε όρους θεωρίας παιγνίων. Στο υπόδειγμα ολιγοπωλίου του Cournot υπάρχουν δύο (δυοπώλιο) ή περισσότερες επιχειρήσεις οι οποίες παράγουν ένα ομογενές προϊόν, αντιμετωπίζουν μια εξωγενώς καθορισμένη ζήτηση και επιλέγουν ανεξάρτητα και χωρίς συνεργασία μεταξύ τους την ποσότητα που θα προσφέρουν στην αγορά. Οι συναρτήσεις κόστους $c_i(q_i)$ είναι συνήθως κοινές για όλες τις επιχειρήσεις. Η θεμελιώδης υπόθεση είναι βέβαια, ότι επιχειρήσεις επιδιώκουν την μεγιστοποίηση των κερδών τους.

Οι επιχειρήσεις βασίζουν την επιλογή τους στην τιμή, η οποία βεβαίως προσδιορίζεται από την τομή της προσφοράς και της ζήτησης. Η συνάρτηση ζήτησης, προσδιορίζεται εξωγενώς, ενώ η συνολική προσφορά ορίζεται ως το άθροισμα των ποσοτήτων που προσφέρουν οι n επιχειρήσεις. Αν λοιπόν υποθέσουμε ότι η συνάρτηση ζήτησης είναι της μορφής

$$Q_D = a + bP$$

με Q_D την ζητούμενη ποσότητα για τιμή P, με a,b σταθερές παραμέτρους και λύσουμε την παραπάνω εξίσωση ως προς την τιμή, έχουμε την τιμή σαν συνάρτηση της ποσότητας. Η τιμή λοιπόν μπορεί να οριστεί σαν η αντίστροφη συνάρτηση ζήτησης, η οποία υποτίθεται πάντα ότι είναι αμφιμονοσήμαντη.

$$P = b^{-1}(Q_D - a)$$

Η ακριβής τιμή και ποσότητα ισορροπίας προσδιορίζονται από τη συνολική ποσότητα που προσφέρουν οι επιχειρήσεις Q_S, σε συνδυασμό με την αντίστροφη συνάρτηση ζήτησης $(Q_S = Q_D)$. Ο προσδιορισμός της προσφερόμενης ποσότητας μιας επιχείρησης, εξαρτάται από το κέρδος που αποδίδει η ποσότητα αυτή, το οποίο εξαρτάται με τη σειρά του από την τιμή ισορροπίας και το κόστος παραγωγής της συγκεκριμένης ποσότητας. Σε όλα σχεδόν τα μοντέλα που θα μας απασχολήσουν στη μελέτη μας, οι συναρτήσεις κόστους θεωρούνται κοινές για όλες τις επιχειρήσεις.

Οι καθαρές στρατηγικές για κάθε επιχείρηση, συνίστανται στην επιλογή μιας συγκεκριμένης ποσότητας $s_i = q_i$. Η τελική ισορροπία προσδιορίζεται, στα πλαίσια της θεωρίας παιγνίων, με την έννοια της ισορροπίας κατά Nash. Η ισορροπία κατά Nash υπάρχει πάντα στην περίπτωση παιγνίων N παικτών με πεπερασμένο αριθμό στρατηγικών (Nash 1950). Εφόσον χρησιμοποιούμε γενετικούς αλγορίθμους για την μοντελοποίηση των επιλογών των παικτών, το παίγνιο έχει πεπερασμένο αριθμό καθαρών στρατηγικών (ποσοτήτων). Θα επανέλθουμε στο θέμα αυτό παρακάτω. Προς το παρόν δίνουμε κάποιους βασικούς ορισμούς για την ισορροπία κατά Nash.

Όπως είπαμε οι συγκεκριμένες ποσότητες που επιλέγει μια επιχείρηση αποτελούν τις καθαρές στρατηγικές της. Αν συμβολίσουμε με $s_i = q_i$ την καθαρή στρατηγική της επιχείρησης i το σύνολο των επιλογών όλων των παικτών $s = (s_1,...,s_N)$ λέγεται «προφίλ στρατηγικών» (Fundenberg and Tirole 1991). Αν από την άλλη οι παίκτες χρησιμοποιούν μεικτές στρατηγικές $\sigma_\iota = (p_{i1}, p_{i2},..., p_{iJ})$, επιλέγοντας με την πιθανότητα p_{ij} (ο παίκτης i) την στρατηγική j, το σύνολο των στρατηγικών $\sigma = (\sigma_1,...,\sigma_N)$ ονομάζεται ανάλογα «προφίλ μεικτών στρατηγικών». Συμβολίζουμε επίσης με $\sigma_{-\iota}$ το σύνολο των επιλογών (μεικτών στρατηγικών) των υπολοίπων παικτών, εκτός του ι. Ανάλογα ορίζουμε το s_{-i}. Το κέρδος του παίκτη i εξαρτάται βέβαια από τις επιλογές όλων των παικτών. Στην περίπτωση μεικτών στρατηγικών, υπολογίζουμε το αναμενόμενο κέρδος του παίκτη, σε κάθε δεδομένο προφίλ στρατηγικών και το συμβολίζουμε με $\pi_\iota(\sigma)$ ή $u_i(\sigma)$. Ανάλογα συμβολίζουμε τα κέρδη των παικτών στην περίπτωση που περιοριζόμαστε σε καθαρές στρατηγικές.

Ισορροπία κατά Nash. Το προφίλ μεικτών στρατηγικών σ^* είναι ισορροπία κατά Nash αν και μόνο αν, για κάθε παίκτη ι ισχύει

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(s_i, \sigma_{-i}^*) \qquad \forall s_i \in S_i \text{ και } i = 1,2,...,n \qquad \text{(Fundenberg and Tirole 1991)}$$

Σημειώνουμε ότι η παραπάνω εξίσωση συνεπάγεται την

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*) \qquad \forall \sigma_i \neq \sigma_i^* \text{ και } i = 1,2,...,n$$

Δεδομένου ότι το κέρδος από μια μεικτή στρατηγική είναι ένα σταθμισμένο άθροισμα των κερδών των μεικτών στρατηγικών, και εφόσον το κέρδος μιας στρατηγικής είναι μεγαλύτερο από το κέρδος όλων των καθαρών στρατηγικών, θα είναι μεγαλύτερο κι από το κέρδος όλων των υπολοίπων μεικτών στρατηγικών. Αν πάλι το s^* είναι ισορροπία κατά Nash και $\forall s_i \neq s_i^*$, έχουμε

$$u_i\left(s_i^*, s_{-i}^*\right) > u_i\left(s_i, s_{-i}^*\right)$$

τότε το s^* ονομάζεται *αυστηρή ισορροπία κατά Nash* (strict Nash equilibrium). Εξ' ορισμού, μόνο μια καθαρή στρατηγική μπορεί να είναι αυστηρή ισορροπία κατά Nash.

Το σ^* είναι λοιπόν ισορροπία κατά Nash, αν ο κάθε παίκτης, «απαντά» με το βέλτιστο τρόπο, μεγιστοποιώντας δηλαδή το αναμενόμενο κέρδος του, στη δεδομένη επιλογή των υπολοίπων παικτών. Αν λοιπόν όλες οι μεικτές στρατηγικές όλων των παικτών είναι τέτοιες «βέλτιστες απαντήσεις», τότε το προφίλ στρατηγικών είναι ισορροπία κατά Nash. Η ισορροπία κατά Nash είναι μια «συνεπής» πρόβλεψη για την εξέλιξη του παιγνίου, με την έννοια ότι εφόσον όλοι οι παίκτες προβλέπουν ότι οι αντιπάλοι τους θα παίξουν με αυτόν τον τρόπο, τότε δεν υπάρχει κίνητρο για αυτούς να παίξουν κάποια διαφορετική μεικτή στρατηγική. Αυτό δε σημαίνει βέβαια ότι σε κάθε περίπτωση θα καταλήξουμε σε ισορροπία κατά Nash. Τόσο στα περισσότερα μοντέλα που θα μελετήσουμε εδώ, όσο και στις περισσότερες επίσης πειραματικές έρευνες με πραγματικούς παίκτες, οι αποφάσεις των παικτών σπάνια συγκλίνουν σε μια ισορροπία κατά Nash.

Ο Nash (1950), απέδειξε ένα από τα θεμελιώδη θεωρήματα της θεωρίας παιγνίων. Απέδειξε ότι όλα τα παίγνια N παικτών με πεπερασμένο αριθμό καθαρών στρατηγικών, έχουν ισορροπία κατά Nash (όχι κατ΄ ανάγκη μοναδική). Οι Lemke και Howson (1964) εισήγαγαν έναν αλγόριθμο για την εύρεση σημείου ισορροπίας για παίγνια δύο παικτών, που συγκλίνει αποδεδειγμένα σε σημείο ισορροπίας κατά Nash. Έχουν επίσης προταθεί πολλοί ευρετικοί αλγόριθμοι για την επίλυση παιγνίων n παικτών (βλ. Sandholm et al 2005, Porter et al 2004). Αλλά και οι γενετικοί αλγόριθμοι που θα παρουσιάσουμε εδώ αποτελούν επίσης ευρετικούς αλγόριθμους που επιδιώκουν την εύρεση της ισορροπίας κατά Nash, εφόσον ερμηνευτούν κάτω από το κατάλληλο πρίσμα.

Το ολιγοπώλιο Cournot ήταν ίσως το πρώτο παίγνιο οικονομικού τύπου που μελετήθηκε στα πλαίσια της θεωρίας παιγνίων μη μηδενικού αθροίσματος. Ο Nash έκανε εκτεταμένη αναφορά σε αυτό στη διδακτορική του διατριβή (Nash 1950), αλλά και η λύση που είχε προτείνει ο Cournot, για το δυοπώλιο, δεν ήταν παρά μια πρώιμη μορφή της ισορροπίας κατά Nash. Η ισορροπία του αυτή εξαρτάται από τη μορφή των συναρτήσεων ζήτησης και κόστους. Στην περίπτωση που οι παίκτες χρησιμοποιούν καθαρές στρατηγικές, $q_i \in Q_i = [0, \infty)$ έχουμε για την περίπτωση δύο παικτών (Fundenberg and Tirole 1991):

$$u_i(q_1, q_2) = q_i p(q_1, q_2) - c_i(q_i) \qquad i = 1,2$$

με $p(q_1, q_2)$ την τιμή ισορροπίας όταν η προσφορά είναι $Q = q_1 + q_2$ (προσδιορίζεται από την αντίστροφη συνάρτηση ζήτησης) και $c_i(q_i)$ το κόστος της επιχείρησης i από την παραγωγή ποσότητας q_i. Οι «συναρτήσεις αντίδρασης» $r_1 : q_2 \to q_1$ και $r_2 : q_1 \to q_2$, δείχνουν τη βέλτιστη επιλογή του κάθε παίκτη, για δεδομένη την επιλογή του αντιπάλου. Αν οι u_i είναι παραγωγίσιμες και αυστηρά κοίλες και ισχύουν οι κατάλληλες συνοριακές συνθήκες[1], μπορούμε να λύσουμε αυτές τις συναρτήσεις αντίδρασης, χρησιμοποιόντας τις συνθήκες πρώτης τάξης. Για παράδειγμα η $r_2(\cdot)$ ικανοποιεί την

$$p(q_1 + r_2(q_1)) + p'(q_1 + r_2(q_1))r_2(q_1) - c'_2(r_2(q_1)) = 0$$

Η τομή των δύο συναρτήσεων αντίδρασης -αν υπάρχει- μας δίνει την ισορροπία κατά Nash, του δυοπωλίου Cournot.

Για παράδειγμα, αν η αντίστροφη συνάρτηση ζήτησης είναι $p(q) = \max(0, 1 - q)$ και οι συναρτήσεις κόστους είναι οι ίδιες για τις δύο επιχειρήσεις $c(q) = cq$ με $0 \leq c \leq 1$, τότε

$$r_2(q_1) = (1 - q_1 - c)/2$$

και λόγω συμμετρίας,

$$r_1(q_2) = (1 - q_2 - c)/2$$

Η ισορροπία κατά Nash είναι $q_1^* = q_2^* = \frac{1}{3}$.

Εφόσον θεωρήσουμε δεδομένη τη μοντελοποίηση μέσω γενετικών αλγορίθμων, κάθε παίκτης έχει εκ των πραγμάτων έναν περιορισμένο αριθμό καθαρών στρατηγικών, οπότε έχει εφαρμογή το θεώρημα του Nash (1950) περί ύπαρξης ισορροπίας σε μεικτές στρατηγικές. Αν πάλι θεωρήσουμε ότι ο κάθε παίκτης έχει τη δυνατότητα να επιλέξει μεταξύ άπειρων ποσοτήτων, τότε το γεγονός ότι τα κέρδη είναι συνεχείς συναρτήσεις των ποσοτήτων ($u_i = p(q_i, Q_{-i}) - c_i(q_i)$) μπορεί να αποβεί ιδιαίτερα χρήσιμο, στην διερεύνηση ύπαρξης ισορροπίας κατά Nash. Όπως απέδειξαν οι Debreu (1952), Glicksberg (1952) αν το σύνολο των εναλλακτικών καθαρών στρατηγικών S είναι ένα μη

[1] Οι «κατάλληλες συνοριακές συνθήκες» είναι οι ικανές συνθήκες που πρέπει να ικανοποιούνται ώστε η λύση να ανήκει στο εφικτό σύνολο Q_i. Έτσι αν $q \in [0, \infty)$, αρκεί $p(q) - c_2'(q) > 0 \;\; \forall q$ (κάτι που συνεπάγεται γενικά ότι $c_2'(0) = 0$), ώστε η $r_2(q_1)$ να είναι αυστηρά θετική για όλα τα q_1 και $\lim_{q \to \infty} p(q) + p'(q) - c_2(q) < 0$

κενό, συμπαγές[2] και κυρτό[3] υποσύνολο (non empty, compact and convex) του R^N, και οι συναρτήσεις κέρδους είναι συνεχείς στο S και οιωνεί-κοίλες[4] (quasi-concave) για κάθε s_i, το παίγνιο έχει ισορροπία σε καθαρές στρατηγικές. Μπορούμε εύκολα να εξασφαλίσουμε ότι η συνθήκη για το χώρο των στρατηγικών θα ισχύει, ορίζοντας τους κατάλληλους περιορισμούς. Αν υποθέσουμε για παράδειγμα ότι οι επιχειρήσεις επιλέγουν μια ποσότητα μεταξύ 0 και $Q_{\max}$, τότε το $S = [0, Q_{\max}]^N$ είναι συμπαγές και κυρτό σύνολο στον R^N. Η δεύτερη συνθήκη από την άλλη είναι περισσότερο περιοριστική, αφού απαιτεί αυστηρότερες συνθήκες στις δεύτερες παραγώγους των συναρτήσεων κέρδους. Βέβαια η μη ισχύς του θεωρήματος δεν συνεπάγεται κατ΄ ανάγκη την μη ύπαρξη ισορροπίας σε καθαρές στρατηγικές, δεδομένου ότι οι συνθήκες είναι ικανές. Ωστόσο, όπως έδειξαν οι Roberts and Sonnenschein (1976), η ισορροπία σε καθαρές στρατηγικές είναι ανέφικτη ακόμα και αν έχουμε αρκετά απλές συναρτήσεις κέρδους.

Ο Glicksberg (1952) προχώρησε ακόμα παραπέρα καλύπτοντας και την περίπτωση μη οιωνεί-κοίλων στρατηγικών. Έδειξε ότι στην περίπτωση που οι χώροι των επιτρεπτών καθαρών στρατηγικών S_i είναι μη κενά συμπαγή υποσύνολα, ενός μετρικού χώρου και οι συναρτήσεις κέρδους είναι απλά συνεχείς, τότε υπάρχει ισορροπία κατά Nash σε μεικτές στρατηγικές. Δεδομένου ότι οι συναρτήσεις κέρδους στο παίγνιο Cournot είναι συνεχείς, αρκεί μια «συμβατική» επιλογή του χώρου των δυνατών στρατηγικών, για να ισχύει το θεώρημα αυτό και να έχουμε ισορροπία σε μεικτές στρατηγικές.

Ο McManus (1962, 1964) έδειξε ότι κάθε συμμετρικό ολιγοπώλιο Cournot (με κοινές συναρτήσεις κόστους), με τη συνάρτηση κόστους κυρτή, έχει ισορροπία σε καθαρές στρατηγικές, συμμετρική μάλιστα για τους Ν παίκτες. Ο Novshek (1985) έδειξε ότι κάθε παίγνιο Cournot Ν παικτών, έχει ισορροπία σε μεικτές στρατηγικές, εφόσον το οριακό έσοδο της κάθε επιχείρησης είναι φθίνουσα συνάρτηση του συνολικού εσόδου των υπολοίπων παικτών. Ο Kukushkin (1992) επέκτεινε την ανάλυσή του Novshek, για διαφορετικά άνω φράγματα στο σύνολο παραγωγικών δυνατοτήτων της κάθε επιχείρησης. Ο Vives (1990) έδειξε ότι, στην περίπτωση του δυοπωλίου, η συνθήκη αυτή για το οριακό έσοδο, συνεπάγεται ότι η συνάρτηση κέρδους είναι υπό-τμηματική (sub-modular) [5] τόσο

[2] Συμπαγές (compact) υποσύνολο ενός μετρικού χώρου, ονομάζεται κάθε κλειστό (όλα τα οριακά του σημεία ανήκουν στο σύνολο) και φραγμένο υποσύνολο του χώρου (βλ. και Wikipedia: "Compact Space")

[3] Ένα $S \subset R^n$ ονομάζεται κυρτό (convex) αν για κάθε $x, y, \in S$, $\lambda x + (1-\lambda) y \in S$, $\forall \lambda \in R$.

[4] Μια συνάρτηση $f : X \to R$ λέγεται οιωνεί κοίλη, αν $\forall x \forall y \in X$ και $\lambda \in (0,1)$, $f(\lambda x + (1-\lambda) y) \geq \min\{f(x), f(y)\}$. Στην περίπτωση μάλιστα που $X \subset R$ η f είναι οιωνεί κοίλη (quasi concave) αν είναι μονότονη, ή αρχικά αύξουσα και στη συνέχεια φθίνουσα (Wilson 2005)

[5] Μια συνάρτηση $F : R_+^2 \to R$ είναι υπό ή υπέρ τμηματική (sub-modular ή super-modular) αν για κάθε $x_1 \geq x_2$ και $y_1 \geq y_2$ ισχύει $F(x_1, y_1) - F(x_2, y_1) \geq (\leq)\ F(x_1, y_2) - F(x_2, y_2)$. Βλ. επίσης Alos Ferer, Ania (2005) στο

σε σχέση με την ποσότητα της επιχείρησης, όσο και σε σχέση με τη συνολική ποσότητα όλων των παικτών. Αυτό συνεπάγεται ότι το δυοπώλιο είναι υπέρ-τμηματική (super-modular) και έχει ισορροπία σε καθαρές στρατηγικές (Milgrom and Roberts 1990), η οποία είναι μοναδική, εφόσον οι συναρτήσεις κόστους είναι κυρτές. Στην περίπτωση τριών οι περισσότερων παικτών, η υπερ-τμηματικότητα (super-modularity) δεν ισχύει πιά. Θα πρέπει να ληφθεί υπόψη όμως ότι το κέρδος της κάθε επιχείρησης εξαρτάται μονάχα από το άθροισμα των κερδών των συμπαικτών της. Ο Amir (1996), χρησιμοποίησε την ιδιότητα αυτή, που εισήγαγε ο Selten (1970) και έδειξε ότι ένα συμμετρικό n-οπώλιο Cournot με γραμμική συνάρτηση κόστους, φραγμένες επιλογές ποσοτήτων και κυρτή συνάρτηση ζήτησης έχει φυσικό λογάριθμο super-modular και συμμετρική ισορροπία σε καθαρές στρατηγικές. Στην ίδια μελέτη έδειξε για την περίπτωση του δυοπωλίου, ότι η υπερ-τμηματικότητα (super-modularity) ισχύει και στην περίπτωση απλά αυξουσών συναρτήσεων κόστους, αρκεί ο φυσικός λογάριθμος της συνάρτησης ζήτησης να είναι κοίλη συνάρτηση. Τέλος οι Dubey et al. (2006) έδειξαν ότι αν $\forall\, q'_{-i} < q_{-i}$ και

$$0 > \frac{q' - q}{q'_{-i} - q_{-i}}$$

(με q την βέλτιστη απάντηση της επιχείρησης i στην συνολική ποσότητα των υπολοίπων παικτών q_{-i}) ισχύει $\forall\, i$, το n-οπώλιο Cournot είναι «παίγνιο ψευδο-δυναμικού»[6] (pseudo-potential game) και έχει ισορροπία κατά Nash σε καθαρές στρατηγικές, με μόνη προϋπόθεση, ο χώρος των στρατηγικών να είναι συμπαγής. Αυτή η προϋπόθεση συμπεριλαμβάνει και τη διακριτή περίπτωση στην οποία ο αριθμός των εναλλακτικών στρατηγικών είναι πεπερασμένος. Αν μάλιστα

$$\frac{q' - q}{q'_{-i} - q_{-i}} > -1$$

ο Vives (2000) δείχνει ότι η ισορροπία αυτή είναι μοναδική. Οι Dubey et al. (2006) έδειξαν και ότι ένα παίγνιο Cournot n παικτών, με συμπαγή χώρο στρατηγικών (ή για $s_i \in R_+$, Amir 1996), με το λογάριθμο της αντίστροφης συνάρτησης ζήτησης να είναι κοίλη (log-concave) και γνησίως φθίνουσα συνάρτηση, συναρτήση κόστους για την κάθε επιχείρηση c_i γνησίως αύξουσα και συνεχή από τα αριστερά και με το κέρδος της κάθε επιχείρησης $\pi_i = Pq_i - c_i$ να γίνεται αρνητικό για αρκετά μεγάλο q_i, τότε το παίγνιο έχει ισορροπία κατά Nash σε καθαρές στρατηγικές.

Για την περίπτωση παιγνίων Cournot με άπειρο αριθμό παικτών, που δε θα καλύψουμε εδώ, βλ. Khan (1985), Khan και Παπαγεωργίου (1987), Γιανέλλης και Prabhhakar (1983) και Kim et al. (1985).

κεφάλαιο 3

[6] Για τον ορισμό του pseudo-potential game βλ. Dubey et al. (2006).

2.2.2. Ασυμπτωτική ευστάθεια της ισορροπίας κατά Nash

Μέχρι τώρα δικαιολογήσαμε την σύγκλιση στην ισορροπία κατά Nash, χρησιμοποιώντας επιχειρήματα που αναφέρονται στην λογική και προβλεπτική ικανότητα των παικτών. Μπορούμε όμως να θεωρήσουμε την ισορροπία κατά Nash, σαν ένα προϊόν εξελικτικής αναθεώρησης των στρατηγικών των παικτών, με το χρόνο. Να υποθέσουμε δηλαδή ότι οι παίκτες αποκλίνουν, με κάποιο προκαθορισμένο πρότυπο μάθησης που εξαρτάται από τα κέρδη τους, από τις παρελθοντικές τους επιλογές, καθώς το παίγνιο επαναλαμβάνεται και τελικά το προφίλ στρατηγικών συγκλίνει στην ισορροπία κατά Nash.

Ήδη ο Cournot (1838), περιέγραψε μια τέτοια διαδικασία προσαρμογής, στην περίπτωση του δυοπωλίου Cournot. Αν υποθέσουμε ότι ο πρώτος παίκτης καθορίζει την ποσότητά του και στη συνέχεια ο δεύτερος «απαντά» με τον καλύτερο τρόπο σε αυτήν την επιλογή, για να απαντήσει με τη σειρά του ο πρώτος με τον βέλτιστο για αυτόν τρόπο στην επιλογή του δεύτερου κ.ο.κ., ο Cournot έδειξε ότι μπορούμε να συγκλίνουμε στην ισορροπία Nash (η λύση του Cournot είναι ταυτόσημη με την ισορροπία κατά Nash). Διατυπώνοντας τα παραπάνω με τη βοήθεια των «συναρτήσεων αντίδρασης» έχουμε

$$q_2^1 = r_2\left(q_1^0\right),\ q_1^1 = r_1\left(q_2^1\right),\ q_2^2 = r_2\left(q_1^1\right), \ldots$$

Αν αυτή η σειρά συγκλίνει σε μια στάσιμη κατάσταση στην οποία οι ποσότητες παραμένουν σταθερές, q_1^* και q_2^*, τότε λέμε ότι είναι **ασυμπτωτικά σταθερή** (Fundenberg and Tirole 1991). Σε αυτήν την περίπτωση θα είναι

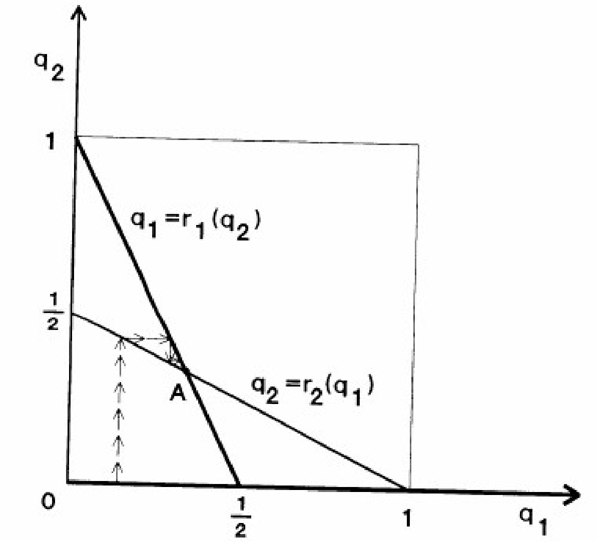
$$r_1\left(q_2^*\right) = q_1^* \text{ και } r_2\left(q_1^*\right) = q_2^*$$

δηλαδή η στάσιμη κατάσταση θα είναι ισορροπία κατά Nash (βλ. σχ. 2.1). Εφόσον τώρα η ασυμπτωτικά σταθερή κατάσταση είναι ισορροπία σε καθαρές στρατηγικές, λέγεται και *σταθερή κατά Cournot* (Vives 2000). Αν τώρα υπάρχουν περισσότερες ισορροπίες κατά Nash στο παίγνιο και η δυναμική διαδικασία των «βέλτιστων απαντήσεων» που περιγράψαμε παραπάνω, συγκλίνει σε μια ισορροπία κατά Nash, λέμε ότι έχουμε ασθενή ευστάθεια Cournot (weak Cournot stability). Θα μπορούσαμε για παράδειγμα να έχουμε περισσότερα από ένα σημεία ισορροπίας κατά Nash και το παίγνιο να συγκλίνει σε διαφορετικό κάθε φορά, ανάλογα με το σημείο εκκίνησης (βλ. σχ. 2.2).

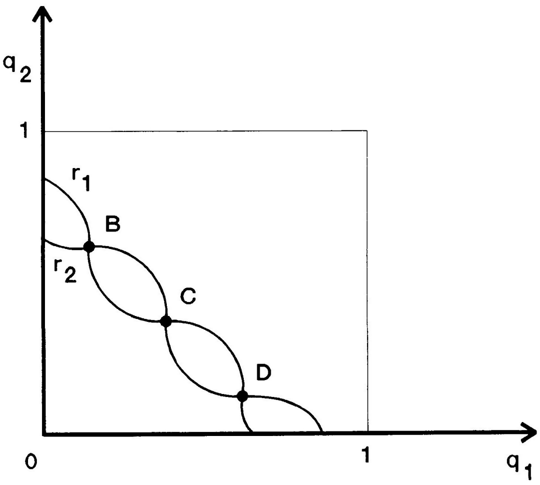
Όσον αφορά την περίπτωση του δυοπωλίου, και εφόσον οι συναρτήσεις αντίδρασης είναι δύο φορές παραγωγίσιμες, οι Fundenberg and Tirole (1991) δίνουν μία ικανή συνθήκη για να είναι ένα σημείο ισορροπίας κατά Nash, ασυμπτωτικά σταθερό, εφόσον το σημείο εκκίνησης της διαδικασίας βέλτιστων απαντήσεων βρίσκεται αρκετά κοντά του. Η συνθήκη είναι

$$\left|\frac{dr_1}{dq_2}\right|\left|\frac{dr_2}{dq_1}\right| < 1 \text{ ή ισοδύναμα } \frac{\partial^2 u_1}{\partial q_1 \partial q_2}\frac{\partial^2 u_2}{\partial q_1 \partial q_2} < \frac{\partial^2 u_1}{\partial^2 q_1^2}\frac{\partial^2 u_2}{\partial^2 q_2^2}$$

σε μια ανοικτή γειτονιά του σημείου ισορροπίας.



σχήμα 2.1



σχήμα 2.2

Σημειώνουμε ότι ακόμα και στην περίπτωση που οι παίκτες αντιδρούν ταυτόχρονα στις προηγούμενες επιλογές του συμπαίκτη τους, οι ιδιότητες που απαιτούνται για ευστάθεια παραμένουν οι ίδιες. Ας υποθέσουμε ότι οι δύο παίκτες επιλέγουν ταυτόχρονα την βέλτιστη απάντησή τους στην επιλογή του συμπαίκτη τους στην προηγούμενη περίοδο. Η δυναμική αυτή διαδικασία περιγράφεται ως

$$q^t = \left(q_1^t, q_2^t\right) = \left(r_1\left(q_2^{t-1}\right), r_2\left(q_1^{t-1}\right)\right) \equiv f\left(q^{t-1}\right)$$

Από τη θεωρία των δυναμικών συστημάτων (Hirsch and Smale 1974), ξέρουμε ότι ένα δεδομένο σημείο q^* είναι ασυμπτωτικά σταθερό, αν οι ιδιοτιμές της $\partial f\left(q^*\right)$, έχουν πραγματικά μέρη, οι απόλυτες τιμές των οποίων είναι μικρότερες από 1. Οι προηγούμενες συνθήκες για τις κλίσεις των συναρτήσεων αντίδρασης είναι σαφέστατα επαρκείς για να εξασφαλίσουν ότι θα ικανοποιείται αυτή η συνθήκη (Fundenberg and Tirole 1991).

Για την περίπτωση N παικτών, αν ισχύουν οι συνθήκες των Dubey et al. (2006) , το παίγνιο Cournot είναι pseudo-potential game. Αν τώρα το σύνολο των εναλλακτικών είναι πεπερασμένο, τότε οι ακολουθίες βέλτιστής απάντησης δεν έχουν κύκλους. Σ΄ αυτήν την περίπτωση και εφόσον έχουμε διαδοχικές απαντήσεις, η ακολουθία των βέλτιστων απαντήσεων θα συγκλίνει, στην ισορροπία κατά Nash σε καθαρές στρατηγικές (Dubey et al. 2006- remark 1). Αν μάλιστα η ισορροπία είναι μοναδική, ισχύει δηλαδή

$$0 > \frac{q' - q}{q'_{-i} - q_{-i}} > -1$$

τότε το παίγνιο είναι παίγνιο δυναμικού βέλτιστων απαντήσεων (best-reply potential game). Έτσι, «αν οι συναρτήσεις βέλτιστης απάντησης είναι μονότιμες και αύξουσες, έχουμε μια μοναδική ισορροπία κατά Nash σε καθαρές στρατηγικές η οποία είναι σταθερή κατά Cournot, ενώ ακόμα κι αν η ισορροπία δεν είναι μοναδική, οι διαδοχικές βέλτιστες απαντήσεις θα συγκλίνουν στην ισορροπία σε καθαρές στρατηγικές» (Jensen 2007). Τέλος οι Dindos και Mezzeti (2003) εξέτασαν την σύγκλιση τυχαίων ακολουθιών βελτίωσης, την περίπτωση δηλαδή που επιλέγεται ένας παίκτης με τυχαίο τρόπο, ο οποίος επιλέγει επίσης τυχαία μια στρατηγική η οποία -ενώ δεν είναι η βέλτιστη απάντηση- βελτιώνει ωστόσο το κέρδος του, και έδειξαν ότι μπορούμε να έχουμε σύγκλιση στην ισορροπία κατά Nash, κάτω από κάποιες προϋποθέσεις (βλ. Dubey et al. 2006).

Αν τώρα έχουμε ταυτόχρονες απαντήσεις, οι Dubey et al δείχνουν στην 2η τους παρατήρηση ότι θα πρέπει να ικανοποιούνται περισσότερες προϋποθέσεις για να έχουμε, εγγυημένα, σύγκλιση στην ισορροπία σε σταθερές στρατηγικές. Πιο συγκεκριμένα, οι πρόσθετες ικανές συνθήκες είναι η κυρτότητα του χώρου των στρατηγικών και η μοναδικότητα των βέλτιστων απαντήσεων. Η μοναδικότητα αυτή, είναι δεδομένη στην «κλασική» περίπτωση, με κοίλες συναρτήσεις κέρδους. Βέβαια, οι συνθήκες αυτές είναι ικανές και όχι αναγκαίες, οπότε δεν αποκλείεται να έχουμε ευστάθεια κατά Cournot, ακόμα και σε περιπτώσεις που δεν ισχύουν οι παραπάνω συνθήκες.

2.2.3. Εξελικτική προσέγγιση στο παίγνιο Cournot

Μια τελείως διαφορετική προσέγγιση είναι αυτή της εξελικτικής θεωρίας παιγνίων, και των εξελικτικών αλγόριθμων βελτιστοποίησης που τη συνοδεύουν. Η προσέγγιση αυτή εισήχθη από τους Smith και Price (1973) για να μοντελοποιήσουν τη συμπεριφορά των ζώων. Θεώρησαν ότι τα ζώα είναι γενετικά προγραμματισμένα να έχουν συγκεκριμένες συμπεριφορές- να παίζουν με άλλα λόγια συγκεκριμένες καθαρές στρατηγικές. Τα ζώα εκείνα με τις πιο επιτυχημένες (προσαρμοσμένες στο περιβάλλον τους) στρατηγικές έχουν μεγαλύτερες πιθανότητες επιβίωσης και διασταύρωσης και συνεπώς τα γονίδια τους έχουν μεγαλύτερη πιθανότητα διάχυσης στις επόμενες γενεές. Έτσι οι στρατηγικές των οποίων η προσαρμοστικότητα είναι μεγαλύτερη από τη μέση προσαρμοστικότητα του πληθυσμού θα τείνουν να υιοθετηθούν από έναν ολοένα αυξανόμενο αριθμό ζώων στις επόμενες γενιές. Κάθε στάσιμη κατάσταση, σημειώνουν οι Smith και Price θα πρέπει να είναι ισορροπία κατά Nash.

Πολλοί οικονομολόγοι, εμπνεύστηκαν από τη συγκεκριμένη μελέτη, και προσπάθησαν να κατασκευάσουν παρόμοια υποδείγματα, για τη μοντελοποίηση της διαδικασίας μάθησης σε οικονομικά παίγνια, συμπεριλαμβανομένου του παιγνίου Cournot. Η πρώτη σχετική αναφορά έγινε από τον Axelrod (1984), ο οποίος ανέλυσε την εξελικτική σταθερότητα (βλ. παρακάτω) του διλήμματος του φυλακισμένου και ακολούθησε ο Sugden(1986) που χρησιμοποίησε την έννοια της εξελικτικής σταθερότητας σαν κριτήριο επιλογής μεταξύ των εναλλακτικών σημείων ισορροπίας κατά Nash, ένα πρόβλημα που θίξαμε και παραπάνω.

Το βασικό χαρακτηριστικό της προσέγγισης αυτής, όσον αφορά το παίγνιο Cournot είναι ένας μεγάλος αριθμός εναλλακτικών ποσοτήτων που επιλέγονται από τους παίκτες (η αντιστοίχιση εξαρτάται από το συγκεκριμένο υπόδειγμα). Κάθε παίκτης επιλέγει μια συγκεκριμένη ποσότητα, χωρίς να λαμβάνει υπόψη άμεσα τις επιλογές των υπολοίπων παικτών. Οι επιλογές αυτές εισέρχονται έμμεσα στην διαδικασία επιλογής των μελλοντικών στρατηγικών του κάθε πάικτη, διαμορφώνοντας το περιβάλλον στο οποίο θα κληθεί να κάνει τις επιλογές του αυτές. Η επίδραση του περιβάλλοντος στην αποδοτικότητα του κάθε πράκτορα, μετριέται με μια συνάρτηση προσαρμοστικότητας, η οποία προσδιορίζει το κατά πόσον είναι επιθυμητή μια συμπεριφορά και συνεπώς επηρρεάζει άμεσα τη διαδικασία επιλογής. Στο παίγνιο Cournot, η συνάρτηση προσαρμοστικότητας ταυτίζεται συνήθως με το κέρδος του παίκτη.

Εμείς στην μελέτη μας αυτή θα εξετάσουμε μια συγκεκριμένη κατηγορία εξελικτικών μοντέλων που έχουν προταθεί στο συγκεκριμένο πεδίο. Θα εξετάσουμε τα μοντέλα γενετικών αλγορίθμων που χρησιμοποιούνται για την διερεύνηση του παιγνίου Cournot. Όπως θα δούμε, η σύγκλιση των στρατηγικών των πληθυσμών σε ισορροπία κατά Nash, κάθε άλλο παρά δεδομένη θα μπορούσε να θεωρηθεί.

2.3. Εξελικτικά σταθερές στρατηγικές (Evolutionary Stable Strategies – ESS)

Στη εξελικτική θεωρία παιγνίων (evolutionary game theory), ο όρος εξελικτικά σταθερή στρατηγική (Evolutionary Stable Strategy – ESS για συντομία), αναφέρεται σε μια στρατηγική, η οποία, αν υιοθετηθεί από έναν πληθυσμό παικτών, είναι μια σταθερή επιλογή, με την έννοια ότι η ισορροπία που προκύπτει με την υιοθέτησή της από τον πληθυσμό, είναι πολύ δύσκολο να ανατραπεί από την εισαγωγή στον πληθυσμό κάποιας μεταλλαγμένης (mutant), εναλλακτικής στρατηγικής. Η ESS είναι το ανάλογο της ισορροπίας κατά Nash (Nash Equilibrium) για την εξελικτική θεωρία παιγνίων, και εκφράζει ακριβώς τη σταθερότητα της στρατηγικής, στον χρόνο. Πρόκειται για ένα Nash Equilibrium, με άλλα λόγια, που έχει την επιπρόσθετη ιδιότητα, ότι η υιοθέτησή του από ένα μεγάλο μέρος των παικτών, οδηγεί στην μη ανατροπή της κατάστασης αυτής, με το πέρασμα του χρόνου, αφού ο μηχανισμός που εκφράζει τη «φυσική επιλογή» στη θεωρία (εξελικτικός ή γενετικός αλγόριθμος κλπ.) δεν επιτρέπει σε διαφορετικές στρατηγικές, να ανατρέψουν την υιοθέτησή της στρατηγικής.

Η έννοια της ESS, αναπτύχθηκε για να αποτελέσει μια ισοδύναμη θεωρία με τη θεωρία της ισορροπίας κατά Nash, σε περιπτώσεις που οι παίκτες δεν διακρίνονται για την ευφυΐα τους. Έτσι σε πληθυσμούς ζώων ή, όπως στην περίπτωσή μας, σε πληθυσμούς πρακτόρων με απλή σχετικά συμπεριφορά λήψης αποφάσεων, δεν μπορούμε να καταφύγουμε στην δυνατότητα πρόβλεψης και στην επίγνωση του ενός πράκτορα ότι ο άλλος είναι ευφυής, για να δικαιολογήσουμε την επικράτηση μιας ισορροπίας κατά Nash. Παραδείγματα τέτοιων διαδικασιών αποτελούν οι διαδικασίες δοκιμής και λάθους, η συμπεριφορά των κοπαδιών των πουλιών και γενικά όλες οι διαδικασίες μάθησης.

Η έννοια αναπτύχθηκε από τους Smith and Price (1973) και αναλύεται εκτεταμένα στο Smith (1982). Εισήχθη στις κοινωνικές επιστήμες και στην μελέτη της ανθρώπινης συμπεριφοράς από τον Axelrod (1984).

Η έννοια συνδέεται στενά με τους εξελικτικούς και γενετικούς αλγόριθμους. Σε αντίθεση με την ισορροπία κατά Nash, που βασίζεται στην υπόθεση ότι οι παίκτες προσπαθούν συνειδητά να μεγιστοποιήσουν την ωφέλειά τους, έχοντας παράλληλα επίγνωση ότι και οι συμπαίκτες τους κάνουν το ίδιο, εδώ δεν γίνεται καμιά τέτοια υπόθεση. Αντίθετα, υποτίθεται ότι οι στρατηγικές είναι κληρονομούμενες από γενιά σε γενιά, κωδικοποιημένες στα χρωμοσώματα των παικτών και οι παίκτες ενεργούν με τον τρόπο που ενεργούν, απλά γιατί είναι γενετικά προγραμματισμένοι κατ' αυτόν τον τρόπο. Η επιβίωση της στρατηγικής και η κληροδότησή της στις επόμενες γενεές, εξαρτάται από την βιολογική καταλληλότητα (fitness) του παίκτη που φέρει την συγκεκριμένη στρατηγική. Υποτίθεται επίσης ότι μπορούν να εισαχθούν νέες στρατηγικές στον πληθυσμό, μέσω ενός μηχανισμού μετάλλαξης (mutation).

Ο φορμαλιστικός ορισμός έχει ως εξής: Όπως είπαμε προηγουμένως, η ισορροπία κατά Nash σε ένα παιγνίδι δύο παικτών αποτελείται από στρατηγικές τέτοιες ώστε κανείς παίκτης να μην μπορεί να ωφεληθεί, αν αλλάξει στρατηγική, με τη δεδομένη επιλογή του αντιπάλου του. Για τους Smith-Price (1973) για να είναι οι στρατηγικές $S_1, S_2, ..., S_n$ εξελικτικά σταθερές θα πρέπει να ισχύει μια από τις επόμενες δύο συνθήκες, για όλους τους παίκτες.

1) $E(S_1,...,S_n) > E(T_1,...,T_n)$, $\forall\ T_i \neq S_i$, να είναι δηλαδή αυστηρή ισορροπία Nash ή

2) $E(S_1,...,S_n) = E(T_1,...,T_n)$ και $E(T_1,...,T_{k-1},S_k,T_{k+1},...,T_n) > E(T_1,...,T_{k-1},T_k,T_{k+1},...,T_n)$ $\forall\ T_i \neq S_i$

Η δεύτερη συνθήκη υποδηλώνει ότι η υιοθέτηση της στρατηγικής S_i από τον παίκτη i, είναι γενικά αδιάφορη σε σχέση με την υιοθέτηση της T_i (προσφέρουν το ίδιο κέρδος κάτω από γενικές συνθήκες), αλλά η στρατηγική S_i έχει πλεονέκτημα απέναντι στην στρατηγική T_i. Δηλαδή, είτε η επιλογή της S_i, μεγιστοποιεί το αναμενόμενο κέρδος, για όλους τους παίκτες έναντι όλων των εναλλακτικών στρατηγικών T_i, είτε η επιλογή των S_i, είναι αφενός αδιάφορη σε σχέση με κάποιες συγκεκριμένες στρατηγικές T_i, αλλά όταν ένας παίκτης υιοθετήσει την S_i και οι υπόλοιποι τις στρατηγικές T, ο παίκτης αυτός θα έχει μεγαλύτερο αναμενόμενο κέρδος.

Ο παρακάτω ορισμός δόθηκε από τον Thomas (1985) : Οι στρατηγικές S είναι εξελικτικά σταθερές, εφόσον είναι

α) Nash Equilibrium και

β) $E(T_1,...,T_{k-1},S_k,T_{k+1},...,T_n) > E(T_1,...,T_{k-1},T_k,T_{k+1},...,T_n) \;\; \forall\, T_i \neq S_i$

Για να διασαφηνίσουμε τις διαφορές μεταξύ της ισορροπίας κατά Nash και τις εξελικτικά σταθερές στρατηγικές θα δώσουμε ένα παράδειγμα σχετικό με το δίλημμα του φυλακισμένου. Η μήτρα κερδών στο δίλημμα του φυλακισμένου έχει ως εξής:

	Μη ομολογία	Κάρφωμα
Μη ομολογία	3,3	1,4
Κάρφωμα	4,1	2,2

Ως γνωστόν στο δίλημμα του φυλακισμένου, οι δύο παίκτες μπορούν είτε να «καρφώσουν» τον άλλο, οπότε θα κερδίσουν 4 μονάδες αντί για 1 ο άλλος παίκτης, ή να «κρατήσουν» κι οι δύο και να κερδίσουν 3 και 3 μονάδες, ή να αλληλοκαρφωθούν και να πάρουν 2 και 2 μονάδες αντίστοιχα. Το τελευταίο σημείο είναι και η ισορροπία κατά Nash. Εφόσον το παιχνίδι παίζεται επαναλαμβανόμενα, τότε ο ένας παίκτης μπορεί να τιμωρήσει τον άλλον για την ενέργειά του στο παρελθόν. Μια ισορροπία κατά Nash είναι για παράδειγμα η στρατηγική «ό,τι κάνεις κάνω», οπότε ο κάθε παίκτης επιλέγει τη συνεργασία στον πρώτο γύρο και στους επόμενους γύρους υιοθετεί τη συμπεριφορά του συμπαίκτη του στον αμέσως προηγούμενο γύρο. Αν έχουμε ν παίκτες και ολόκληρος ο πληθυσμός παίζει τη στρατηγική αυτή και εμφανιστεί ένας παίκτης που παίζει ας πούμε μια στρατηγική διαρκούς καρφώματος, πάλι η στρατηγική μια σου και μία μου θα είναι πιο αποτελεσματική. Οπότε εφόσον υπάρχει κάποιος μηχανισμός φυσικής επιλογής, η στρατηγική διαρκούς καρφώματος, θα πρέπει να πάψει να χρησιμοποιείται. Η «ό,τι κάνεις κάνω» είναι εξελικτικά σταθερή σε σχέση με την «διαρκές κάρφωμα». Αν όμως κάποιος παίκτης παίζει την «διαρκή συνεργασία», τότε η στρατηγική «ό,τι κάνεις κάνω» έχει βασικά το ίδιο αποτέλεσμα με την στρατηγική αυτή. Ολόκληρος ο πληθυσμός θα συνεργάζεται συνέχεια. Άρα η στρατηγική διαρκούς συνεργασία, θα παραμείνει στον πληθυσμό. Γενικά η «ό,τι κάνεις κάνω» δεν είναι εξελικτικά σταθερή.

Θα επανέλθουμε στο θέμα, παρακάτω, συσχετίζοντάς το παράλληλα με τους γενετικούς αλγορίθμους.

2.4. Ιδιότητες των γενετικών αλγόριθμων στην μοντελοποίηση διαδικασιών μάθησης

Οι γενετικοί και γενικότερα οι εξελικτικοί αλγόριθμοι είναι μια από τις διαφορετικές μεθόδους που μπορούμε να χρησιμοποιήσουμε για να μοντελοποιήσουμε τη διαδικασία μάθησης των πρακτόρων στα Agent Computational Economics και πιο συγκεκριμένα στο παίγνιο Cournot. Ένας πράκτορας που χρησιμοποιεί έναν γενετικό αλγόριθμο, διαθέτει ένα χρωμόσωμα (στην περίπτωση αλγόριθμου με έναν πληθυσμό για όλους τους πράκτορες) ή έναν πληθυσμό χρωμοσωμάτων (στην περίπτωση αλγόριθμου με έναν πληθυσμό για κάθε πράκτορα), για να προσδιορίσει την στρατηγική του, η οποία στην περίπτωση του παιγνίου Cournot αντιστοιχεί στην προσφερόμενη ποσότητα (στην περίπτωση που οι πράκτορες χρησιμοποιούν καθαρές στρατηγικές). Με βάση τις επιλογές των πρακτόρων, προσδιορίζεται μια ολική ποσότητα και συνεπώς μια τιμή στην αγορά, η οποία χρησιμεύει για να υπολογιστεί το κέρδος και η συνάρτηση προσαρμοστικότητας (fitness function) του κάθε πράκτορα. Μετά από κάθε τέτοιο γύρο της προσομοίωσης, τα χρωμοσώματα των πρακτόρων μεταβάλλονται, με τρόπο που σχετίζεται με την προσαρμοστικότητά τους (η οποία σημειώνω ότι εξαρτάται άμεσα από την τρέχουσα τιμή στην εικονική αγορά), οπότε έχουμε μια εξελικτική κατάσταση, στην οποία οι πράξεις των πρακτόρων προσδιορίζουν την κατάσταση στην αγορά, η οποία προσδιορίζει με τη σειρά της τις αποφάσεις των πρακτόρων κ.ο.κ. Το σημαντικότερο λοιπόν ζήτημα που ανακύπτει είναι οι ιδιότητες σύγκλισης των γενετικών αλγορίθμων, μέσα στο συγκεκριμένο πλαίσιο. Αμέσως παρακάτω, θα επισημάνουμε κάποια σημαντικά σημεία σχετικά με τις ιδιότητες των αλυσίδων Markov που, πάντα, περιγράφουν την δυναμική συμπεριφορά ενός γενετικού αλγόριθμου και στην επόμενη ενότητα θα επανέλθουμε στο ζήτημα των των εξελικτικά σταθερών στρατηγικών (evolutionary stable strategies).

2.4.1. Η έννοια της ευστάθειας (stability)

«Η έννοια της σταθερότητας στην κλασική οικονομική επιστήμη, αναφέρεται σε μία κατάσταση που καταλήγουμε μετά από μια αρχική φάση προσαρμογής και στην οποία οι αποφάσεις των οικονομικών μονάδων είτε παραμένουν σταθερές (ασυμπτωτική ευστάθεια – asymptotic stability), είτε μεταβάλλονται μέσα σε ένα περιορισμένο σύνολο εναλλακτικών (ευστάθεια κατά Ljapunov-Ljapunov stability), είτε μεταβάλλονται με κάποιο καθόρισμένο τρόπο (κυκλική ευστάθεια – cyclical stability)[7]» (Riechmann 1997). Παρατηρούμε ότι ο ορισμός αυτός αναφέρεται σε μακροοικονομικά δεδομένα. Το κυρίαρχο ρεύμα της οικονομικής, ασχολείται με την ευστάθεια και

[7] Για μια αναλυτική παρουσίαση βλέπε Azariades(1993) ή Hofbauer and Sigmund (1988).

τις ιδιότητες του μακρο-συστήματος, ενώ η μετάβαση στο μικροοικονομικό επίπεδο γίνεται με την βοήθεια της υπόθεσης του «αντιπροσωπευτικού ατόμου», μιας θεωρητικής οντότητας δηλαδή που υποτίθεται ότι περιγράφει την κοινή συμπεριφορά της μεγάλης μάζας τουλάχιστον των οικονομικών ενοτήτων που μας απασχολούν. Συνεπώς οι έννοιες της ευστάθειας σε μακροσκοπικό και ατομικό επίπεδο, ταυτίζονται.

Η Υπολογιστική Οικονομική Πολλαπλών Πρακτόρων (Agent Computational Economics) δεν ακολουθούν αυτήν την σύμβαση και συνεπώς μπορούμε να παρατηρήσουμε διαφορές στην ευστάθεια στο μάκροοικονομικό επίπεδο (συνολικές ποσότητες και τιμές) και στο ατομικό επίπεδο (διαφορετικές επιλογές απ΄ τον κάθε πράκτορα). Πιο συγκεκριμένα μπορούμε να παρατηρήσουμε :

- Ασυμπτωτική ευστάθεια στις ποσότητες και τις τιμές που παραμένουν έτσι αμετάβλητες και παρόμοια σταθερότητα στη συμπεριφορά των πρακτόρων. Ένα τέτοιο αποτέλεσμα δίνει για παράδειγμα ο «επαυξημένος» αλγόριθμος της Arifovic(1994).
- Ασυμπτωτική ευστάθεια στα μακροοικονομικά δεδομένα, με τις επιλογές των πρακτόρων να μεταβάλλονται κατά καιρούς, παρουσιάζοντας γενικά ετερογενή συμπεριφορά. Ο Dawid(1996) παρουσιάζει μια τέτοια κατάσταση.
- Ευστάθεια κατά Ljapunov για τα μακροοικονομικά δεδομένα, με τις επιλογές των πρακτόρων να ανήκουν επίσης σε ένα περιορισμένο σύνολο. Αυτή την εικόνα παρουσιάζουν οι περισσότεροι γενετικοί αλγόριθμοι, όπως π.χ. οι «βασικοί» αλγόριθμοι της Arifovic(1994).
- Κυκλική ευστάθεια για τα μακροοικονομικά δεδομένα, με ετερογενή συμπεριφορά των πρακτόρων (Riechmann 1997).

Έτσι σε εξελικτικά μοντέλα Υπολογιστικής Οικονομικής Πολλαπλών Πρακτόρων (ACE), θα πρέπει να εξετάζεται τόσο η μακροοικονομική, όσο η μικροοικονομική στασιμότητα.

2.4.2. Βασικές ιδιότητες γενετικών αλγορίθμων

Ένας γενετικός αλγόριθμος βασίζεται σε έναν πληθυσμό χρωμοσωμάτων. Τα χρωμοσώματα είναι ακολουθίες δυαδικών ψηφίων (0 και 1), οι οποίες έχουν κάποιο συμβολισμό, σχετικό με το πρόβλημα που μελετάται κάθε φορά. Έτσι στο παίγνιο Cournot, με καθαρές στρατηγικές, το χρωμόσωμα θεωρείται ότι απεικονίζει κάποια ποσότητα προϊόντος. Υπάρχει δηλαδή μια απεικόνιση του χώρου των χρωμοσωμάτων, στον χώρο των εναλλακτικών ποσοτήτων που μπορεί να προσφέρει μια επιχείρηση στην αγορά. Βέβαια ένα χρωμόσωμα μπορεί να περιέχει πεπερασμένη ποσότητα πληροφορίας (μπορεί να πάρει 2^n διαφορετικές τιμές, με n τον αριθμό των δυαδικών ψηφίων), ενώ ο χώρος των ποσοτήτων είναι συνεχής. Έτσι διακρίνουμε ήδη έναν βασικό περιορισμό των γενετικών αλγορίθμων, όσον αφορά την μοντελοποίηση προβλημάτων με συνεχείς μεταβλητές. Αυτό βέβαια δεν σημαίνει ότι οι γενετικοί αλγόριθμοι δεν είναι αποτελεσματικοί σε αυτά τα προβλήματα. Κάθε άλλο. Απλά πρέπει να θεωρούμε δεδομένη την περιορισμένη ακρίβεια, στην προσέγγισή μας.

Ο κλασικός γενετικός αλγόριθμος, όπως ορίστηκε από τον Holland (1975), συνίσταται στην μεταβολή του συνόλου αυτού των χρωμοσωμάτων, με έναν συγκεκριμένο τρόπο (με τους τελεστές διασταύρωσης, μετάλλαξης και αντιστροφής πιο συγκεκριμένα), ώστε να εξελιχθεί, μέσα σε έναν πεπερασμένο αριθμό βημάτων, σε έναν πληθυσμό με κάποιες επιθυμητές ιδιότητες (που θα περιέχει ένα χρωμόσωμα το οποίο θα απεικονίζει την ελάχιστη τιμή μιας συνάρτησης για παράδειγμα).

Η εξελικτική διαδικασία του γενετικού αλγορίθμου παρομοιάζει σε μεγάλο βαθμό με την εξελικτική διαδικασία στα βιολογικά συστήματα και για αυτό άλλωστε ονομάστηκε έτσι. Κάθε χρωμόσωμα χαρακτηρίζεται από μια θετική τιμή προσαρμοστικότητας (fitness value ή απλά fitness) η οποία δείχνει την «καταλληλότητά» του, για επιβίωση και αναπαραγωγή. Στην περίπτωση για παράδειγμα που το πρόβλημά μας αφορά την ελαχιστοποίηση μιας συνάρτησης, το κάθε χρωμόσωμα θα πρέπει λογικά να απεικονίζει ένα στοιχείο $\mathbf{x}$ του πεδίου ορισμού της συνάρτησης και η προσαρμοστικότητά του θα μπορούσε να είναι η $c - f(\mathbf{x})$ (c ένα άνω φράγμα της f). Η προσαρμοστικότητα του χρωμοσώματος είναι εξ΄ ορισμού, τόσο μεγαλύτερη, όσο καταλληλότερο είναι το χρωμόσωμα. Τα χρωμοσώματα μεταβάλλονται με διασταύρωση (crossover) και μετάλλαξη (mutation).

Ο τελεστής διασταύρωσης δημιουργεί δύο χρωμοσώματα-παιδιά από δύο χρωμοσώματα-γονείς, παίρνοντας τμήματα από το γενετικό υλικό των τελευταίων και ανασυνδιάζοντάς τα, ώστε να δημιουργήσει το γενετικό υλικό των παιδιών. Ο Holland όρισε crossover ενός σημείου, που σημαίνει ότι τα χρωμοσώματα των γονέων «κόβονται» σε ένα (δεδομένο ή τυχαίο) σημείο τους. Το κάθε χρωμόσωμα χωρίζεται έτσι σε δύο τμήματα, το Α1, Α2 για τον 1^{ο} γονέα και το Β1, Β2 για τον δεύτερο. Το χρωμόσωμα για το πρώτο παιδί αποτελείται από τα Α1,Β2 και για το δεύτερο από τα Β1,Α2. Αν για παράδειγμα το χρωμόσωμα του πρώτου γονέα είναι (1,0,0,1) και του 2^{ου} (0,0,1,1) και έχουμε διαχωρισμό μεταξύ δεύτερου και τρίτου ψηφίου, το πρώτο παιδί θα είναι (1,0,1,1) και το δεύτερο (0,0,0,1).

Αν και ο τελεστής διασταύρωσης είναι ένα πραγματικά πανίσχυρο εργαλείο και επιτρέπει την κατασκευή πολλών νέων χρωμοσωμάτων, δεν μπορεί πάντα να καλύψει ολόκληρο τον χώρο των πιθανών χρωμοσωμάτων. Αν για παράδειγμα δεν υπάρχει κανένα χρωμόσωμα με 1^{ο} ψηφίο 0 στον πληθυσμό, τότε είναι αδύνατο για τον τελεστή διασταύρωσης να δημιουργήσει ένα χρωμόσωμα με το ψηφίο αυτό. Έτσι ο Holland εισήγαγε τον τελεστή μετάλλαξης (mutation) σύμφωνα με τον οποίο, μετά την διασταύρωση, είναι δυνατή η αυθόρμητη μεταβολή του καθενός από τα ψηφία του κάθε παιδιού, με κάποια δεδομένη πιθανότητα. Στη συνέχεια ήταν σε θέση να αποδείξει ότι ο πληθυσμός μπορεί να τείνει, κατά πιθανότητα, στην κατάσταση με τη μέγιστη προσαρμοστικότητα, εφόσον η επιλογή των χρωμοσωμάτων που θα χρησιμοποιηθούν σαν γονείς είναι ανάλογη με την προσαρμοστικότητά τους (αυτό λέγεται και κανόνας της ρουλέτας για την επιλογή των χρωμοσωμάτων).

Ο κλασικός αυτός γενετικός αλγόριθμος με επιλογή των γονέων ανάλογη με την προσαρμοστικότητά τους, με εφαρμογή διασταύρωσης με διαχωρισμό σε μεμονωμένο -τυχαίο συνήθως- σημείο και μετάλλαξη, έχει καθιερωθεί να ονομάζεται «κανονικός γενετικός αλγόριθμος» (Goldberg 1987). Ο Holland απέδειξε το λεγόμενο θεώρημα του σχήματος (schema theorem), το θεμελιώδες θεώρημα των γενετικών αλγορίθμων, κατά κάποιο τρόπο. Το θεώρημα βασίζεται στην έννοια του σχήματος. Ένα σχήμα είναι το σύνολο των χρωμοσωμάτων με δεδομένη τιμή σε κάποια από τα δυαδικά ψηφία του χρωμοσώματος. Έτσι το σχήμα $(1,*,0,0,*)$ αποτελείται από τα χρωμοσώματα $(1,1,0,0,1)$, $(1,1,0,0,0)$, $(1,0,0,0,0)$ και $(1,0,0,0,1)$. Η πιθανότητα της μετάλλαξης ενός σχήματος σε κάποιο άλλο σχήμα, είναι ανάλογη με τον αριθμό των σταθερών ψηφίων του σχήματος[8] και την πιθανότητα μετάλλαξης ενός bit. Η πιθανότητα πάλι μετατροπής ενός σχήματος σε κάποιο άλλο, μετά την εφαρμογή του τελεστή διασταύρωσης (με διαχωρισμό σε ένα σημείο), εξαρτάται από την απόσταση, μέσα στο χρωμόσωμα, του πρώτου και του τελευταίου σταθερού ψηφίου του σχήματος[9]. Όσο μεγαλύτερο είναι αυτή η απόσταση, τόσο πιθανότερο είναι ότι ο τελεστής διασταύρωσης θα προκαλέσει το «σπάσιμο» του σχήματος.

Η Mitchell (1996) παρατηρεί τα εξής σημεία, αναφορικά με την έννοια του σχήματος:

- Όλα τα σύνολα χρωμοσωμάτων ανήκουν σε περισσότερα από ένα σχήματα
- Κάθε χρωμόσωμα μήκους l ανήκει, σε 2^l σχήματα
- Κάθε πληθυσμών n χρωμοσωμάτων με μήκος l, ανήκει σε k σχήματα, με $2^l \leq k \leq n2^l$
- Σε κάθε γενιά, ο γενετικός αλγόριθμος υπολογίζει την προσαρμοστικότητα όλων των n χρωμοσωμάτων που βρίσκονται στον τρέχοντα πληθυσμό. Η προσαρμοστικότητες αυτές αποτελούν εκτιμήτριες για την μέση προσαρμοστικότητα όλων των χρωμοσωμάτων που ανήκουν στο κάθε σχήμα που ανήκει το κάθε χρωμόσωμα. Βέβαια απαιτούνται πολλές τέτοιες δοκιμές για να έχουμε ένα αρκετά μεγάλο δείγμα για κάθε σχήμα, ώστε να

[8] τάξη του σχήματος (order of the schema)
[9] μήκος ορισμού του σχήματος (defining length of the schema)

μπορούμε να πούμε ο μέσος των τιμών προσαρμοστικότητας των διαφορετικών χρωμοσωμάτων του σχήματος αποτελεί μια αποτελεσματική εκτιμήτρια.

Μπορούμε να διατυπώσουμε το schema theorem, αφού δώσουμε κάποιους απαραίτητους ορισμούς.

- H είναι ένα σχήμα με τουλάχιστον ένα χρωμόσωμα στον πληθυσμό, τη στιγμή t.
- $m(H,t)$ είναι ο αριθμός των χρωμοσωμάτων του H τη στιγμή t.
- $f(H,t)$ είναι η μέση προσαρμοστικότητα των χρωμοσωμάτων του σχήματος H (σύμφωνα με το δείγμα των χρωμοσωμάτων του H που βρίσκονται στον πληθυσμό πάντα) τη στιγμή t.
- $f(x)$ είναι η προσαρμοστικότητα του χρωμοσώματος x και η $\bar{f}(t)$ η μέση προσαρμοστικότητα του πληθυσμού τη στιγμή t.
- p_{xo} είναι η πιθανότητα εφαρμογής του τελεστή διασταύρωσης (crossover). Σε πολλές υλοποιήσεις $p_{xo} = 1$
- $S_c(H)$ είναι η πιθανότητα επιβίωσης του σχήματος H μετά από την εφαρμογή του τελεστή one-point crossover. Με τον όρο «επιβίωση» εννοούμε ότι ένας τουλάχιστον από τους απογόνους του χρωμοσώματος, θα ανήκει στο σχήμα. Όπως είπαμε η $S_c(H)$ εξαρτάται από το «μήκος ορισμού» του H.
- $L(H)$ είναι το «μήκος ορισμού» του H, ο αριθμός των δυαδικών ψηφίων δηλαδή, μεταξύ του πρώτου και του τελευταίου σταθερού ψηφίου του σχήματος.
- l είναι το μήκος των χρωμοσωμάτων.
- p_m είναι η πιθανότητα μετάλλαξης ενός δυαδικού ψηφίου.
- $o(H)$ είναι η «τάξη» του H, δηλαδή ο αριθμός των σταθερών ψηφίων του χρωμοσώματος.
- $S_m(H)$ είναι η πιθανότητα ότι το σχήμα θα επιβιώσει, μετά την εφαρμογή του τελεστή μετάλλαξης.

Με βάση τους παραπάνω ορισμούς η πιθανότητα καταστροφής ενός σχήματος, δεν μπορεί να είναι χειρότερη από $p_{xo}\left(\frac{L(H)}{l-1}\right)$, που συνεπάγεται ότι $S_c(H) \geq 1 - p_{xo}\left(\frac{L(H)}{l-1}\right)$. Τέλος ισχύει $S_m(H) \geq (1-p_m)^{o(H)}$ (Holland 1975).

Θεώρημα Σχήματος (Schema Theorem)

Το θεμελιώδες αυτό θεώρημα του Holland μας δίνει τον αναμενόμενο αριθμό χρωμοσωμάτων που ανήκουν σε δεδομένο σχήμα H, σε μορφή ανισότητας, σε σχέση με την πιθανότητα επιβίωσης του σχήματος, μετά την εφαρμογή των τελεστών διασταύρωσης μονού σημείου και μετάλλαξης.

$$E(m(H,t+1)) \geq \frac{f(H,t)}{\bar{f}(t)} \cdot m(H,t) \cdot \left(1 - p_{xo}\frac{L(H)}{l-1}\right)(1-p_m)^{o(H)}$$

Βλέπουμε λοιπόν ότι το κάτω φράγμα για τον αναμενόμενο αριθμό των χρωμοσωμάτων ενός σχήματος H, αυξάνεται όταν η μέση προσαρμοστικότητα των χρωμοσωμάτων του H είναι μεγαλύτερη από τη μέση προσαρμοστικότητα του πληθυσμού. Ο συντελεστής αύξησης εξαρτάται εκθετικά από την τάξη του σχήματος, και μικραίνει όσο αυξάνεται το μήκος ορισμού του σχήματος.

Τέλος, αν λάβουμε υπόψη τη πιθανότητα ο τελεστής διασταύρωσης να «σπάσει» το σχήμα, μπορούμε να διατυπώσουμε μια πιο αυστηρή έκδοση του θεωρήματος σχήματος (schema theorem)

$$E\left(m(H,t+1)\right) \geq \frac{f(H,t)}{\bar{f}(t)} \cdot m(H,t) \cdot \left(1 - p_{xo}\frac{L(H)}{l-1}\right)(1-p_m)^{o(H)} \cdot \left[1 - p_{xo}\frac{L(H)}{l-1}\left(1 - \frac{m(H,t)f(H,t)}{M \cdot \bar{f}(t)}\right)\right]$$

ή και να το γενικεύσουμε εύκολα, για οποιονδήποτε γενετικό αλγόριθμο, στον οποίο η επιλογή των γονέων γίνεται με επανατοποθέτηση και ανάλογα με τη προσαρμοστικότητα των γονέων (Goldberg 1987).

Το θεώρημα σχήματος (schema theorem) οδήγησε στη λεγόμενη υπόθεση τμηματικής κατασκευής (building block hypothesis), σύμφωνα με την οποιά τα σχήματα με μεγάλη σχετική προσαρμοστικότητα (προσαρμοστικότητα σε σχέση με τη μέση προσαρμοστικότητα του πληθυσμού), μικρό μήκος ορισμού και τάξη, θα αυξάνουν τα χρωμοσώματά τους στον πληθυσμό εκθετικά, γεμίζοντας τελικά τον πληθυσμό με τα χρωμοσωματά τους.

Ο Holland θεώρησε τέλος σκόπιμο να ορίσει έναν τελεστή αναδιάταξης (inversion) ο οποίος επέλεγε, με κάποια πιθανότητα, ένα τμήμα του χρωμοσώματος και ανέστρεφε τα ψηφία (το 1^{ο} γινόταν τελευταίο, το 2^{ο} προτελευταίο κ.ο.κ.), επιτυγχάνοντας έτσι γρηγορότερη σύγκλιση. Σήμερα υπάρχουν αρκετοί μηχανισμοί για να επιτευχθεί η γρηγορότερη σύγκλιση του αλγόριθμου, όπως η ομοιόμορφη διασταύρωση, ελιτίστικες τακτικές, επιλογή με τουρνουά κλπ. και ο τελεστής αυτός δεν χρησιμοποιείται ιδιαίτερα.

Στην ομοιόμορφη διασταύρωση (uniform crossover) επιλέγεται, για κάθε ψηφίο ξεχωριστά, τυχαία (με πιθανότητα 0.5) ο γονέας απ΄ τον οποίο θα ληφθεί το συγκεκριμένο ψηφίο κατά την κατασκευή λ.χ. του 1^{ου} παιδιού. Το ψηφίο του άλλου γονέα χρησιμοποιείται στο 2^{ο} παιδί. Δημιουργείται, με άλλα λόγια, μια «μάσκα» με 0 και 1, με τυχαίο τρόπο. Αν το ν^{ο} ψηφίο της μάσκας είναι 1 τότε το ν^{ο} ψηφίο του πρώτου παιδιού είναι το ν^{ο} ψηφίο του πρώτου γονέα. Διαφορετικά είναι το ν^{ο} ψηφίου του δεύτερου γονέα.

Στην επιλογή με τουρνουά (tournament selection), επιλέγουμε Τ χρωμοσώματα από τον πληθυσμο, από τα οποία μονάχα εκείνο με την μεγαλύτερη προσαρμοστικότητα αντιγράφεται στο σύνολο των χρωμοσωμάτων που θα αναπαραχθούν. Επαναλαμβάνοντας αυτή τη διαδικασία Ν φορές έχουμε των πληθυσμό των γονέων για την επόμενη γενιά. Στην επιβίωση ή εκλογή με τουρνουά τα παιδιά συγκρίνονται με τους γονείς και επιβιώνουν εκείνοι με την υψηλότερη προσαρμοστικότητα. Υπάρχουν διάφορες παραλλαγές για την εκτέλεση αυτής της διαδικασίας. Ένας τρόπος είναι π.χ. το κάθε παιδί να «συναγωνιστεί» με τον αντίστοιχο γονέα. Ένας άλλος τρόπος είναι να επιβιώσουν τα δύο χρωμοσώματα με την καλύτερη προσαρμοστικότητα, από την τετράδα γονείς- παιδιά. Αυτόν τον τρόπο χρησιμοποιεί η Arifovic(1994) στους «επαυξημένους» αλγορίθμους της. Ένας τρίτος τρόπος είναι τα ζευγάρια γονέας-παιδί να προκύψουν τυχαία από το σύνολο των γονέων και των παιδιών, αντίστοιχα.

Οι περισσότεροι ερευνητές χρησιμοποιούν τον προαναφερθέντα «κανόνα της ρουλέτας» για την επιλογή των χρωμοσωμάτων που θα χρησιμεύσουν σαν γονείς. Αν η προσαρμοστικότητα του χρωμοσώματος i είναι f_i και το άθροισμα των προσαρμοστικοτήτων όλων των χρωμοσωμάτων είναι $\sum f_i$, τότε η πιθανότητα επιλογής του i είναι $\frac{f_i}{\sum f_i}$. Η δειγματοληψία γίνεται με επανατοποθέτηση, ώστε να είναι εφικτή η επαναεπιλογή ενός χρωμοσώματος για γονέα.

2.4.3. Οι γενετικοί αλγόριθμοι σαν διαδικασίες Markov

Ο πληθυσμός ενός γενετικού αλγόριθμου αποτελείται, όπως είπαμε, από Μ χρωμοσώματα, μεγέθους $N = 2^n$ δυαδικών ψηφίων. Θα μπορούσε συνεπώς να περιγραφεί με τη μεταβλητή κατάστασης

$$\overline{m} = \left(m(1), m(2), ..., m(N)\right)$$

με $m(1)$ την απόλυτη συχνότητα του καθενός από τα Ν διαφορετικά χρωμοσώματα. Συνεπώς το άθροισμα των συχνοτήτων θα πρέπει να ισούται με τον αριθμό των χρωμοσωμάτων στον πληθυσμό, Μ.

Υπάρχουν N' διαφορετικοί πληθυσμοί, με $N' = \begin{pmatrix} M + N - 1 \\ M \end{pmatrix} = \begin{pmatrix} M + N - 1 \\ N - 1 \end{pmatrix}$ [Nix and Vose (1992), p.81].

Ο γενετικός αλγόριθμος καθεαυτός, μπορεί να θεωρηθεί μια στοχαστική διαδικασία, που μετασχηματίζει τον τρέχοντα πληθυσμό σε κάποιο άλλο, χρησιμοποιόντας τους στοχαστικούς τελεστές της διασταύρωσης, της μετάλλαξης, κ.ο.κ.. Εύκολα διαπιστώνει κανείς ότι οι πιθανότητες μετάβασης ενός γενετικού αλγορίθμου εξαρτώνται αποκλειστικά από τον τρέχοντα πληθυσμό (την τρέχουσα κατάσταση), χωρίς καμμιά αναφορά στις παρελθοντικές καταστάσεις. Έτσι οι γενετικοί αλγόριθμοι αποτελούν διαδικασίες Markov, και πιο συγκεκριμένα πεπερασμένες και ομογενείς αλυσίδες Markov (finite and time-homogeneous Markov chains), αφού ο χώρος καταστάσεων είναι διακριτός και οι πιθανότητες μετάβασης εξαρτώνται κατ΄ αποκλειστικότητα από τον πληθυσμό και δεν αλλάζουν με τον χρόνο (βλ. και Goodman(1998) ή Isaacson and Madsen(1976)). Αν πάλι χρησιμοποιούσαμε λ.χ. έναν τελεστή μετάλλαξης με μεταβαλλόμενη πιθανότητα μετάλλαξης με τον χρόνο (αρχική πιθανότητα 5%, για παράδειγμα, η οποία θα μειώνεται καθώς προχωράμε στις επόμενες γενιές), η ιδιότητα της ομογένειας δεν ισχύει. Ωστόσο δεν θα ασχοληθούμε εδώ με τέτοιες περιπτώσεις.

Εφόσον η πιθανότητα επιλογής ενός χρωμοσώματος για γονέα είναι ανάλογη της προσαρμοστικότητάς του (roulette wheel selection), τότε τελικά, η πιθανότητα να επιλεγεί στην «δεξαμενή διασταύρωσης» είναι ανάλογη και από τη συνολική προσαρμοστικότητα του πληθυσμού. Τελικά δηλαδή η πιθανότητα να εκλεγεί κάποιος σαν γονέας είναι ανάλογη, αφενός της δικής του προσαρμοστικότητας και αφετέρου της μέσης προσαρμοστικότητας του πληθυσμού και είναι

$$P_1(i|\overline{n}) = \frac{n(i)R(i)}{\sum_{j \in N} n(j)R(j)}$$

με $P(i|n)$, την πιθανότητα επιλογής του εν λόγω χρωμοσώματος i, με δεδομένη την μέση προσαρμοστικότητα του πληθυσμού, $n(i)$ την προσαρμοστικότητά του και $R(j)$ τις συχνότητες των διαφόρων χρωμοσωμάτων στον πληθυσμό. Συνεπώς η πιθανότητα να οδηγηθούμε σε κάποια δεδομένη «δεξαμενή διασταύρωσης» $\overline{m}$ από κάποιο πληθυσμό m (αποφεύγουμε να χρησιμοποιήσουμε τον όρο «πληθυσμός» για το σύνολο των χρωμοσωμάτων-γονέων, γιατί θα πρέπει, σύμφωνα με τον κλασσικό ορισμό, να χρησιμοποιήσουμε τους τελεστές διασταύρωσης και μετάλλαξης, για να μιλάμε για νέο πληθυσμό του γενετικού αλγορίθμου), είναι

$$P_1(\overline{m}|\overline{n}) = \binom{M}{\overline{m}} \prod_{i \in S} P_1(i|\overline{n})^{m(i)}$$

με $\binom{M}{\overline{m}} = \frac{M!}{\prod_{i \in S} m(i)!}$ [Riechmann 1999].

Η παραπάνω εξίσωση μας δίνει τις πιθανότητες μετάβασης στην αλυσίδα Markov που θα προέκυπτε, αν αντί για γενετικό αλγόριθμο είχαμε απλά εκλογή των χρωμοσωμάτων της κάθε γενιάς με κανόνα roulette wheel και έχει καταστάσεις απορρόφησης (absorbing states)[10]. Οι καταστάσεις απορρόφησης (absorbing states) είναι φυσικά, οι πληθυσμοί που αποτελούνται από ένα χρωμόσωμα (και τα m δηλαδή χρωμοσώματα, έχουν την ίδια τιμή), οι οποίες δεν μπορούν να αλλάξουν, αφού πάντα θα επιλέγεται το ίδιο χρωμόσωμα σαν γονέας.

Ο τελεστής διασταύρωσης μπορεί να παρασταθεί σαν μια συνάρτηση

[10] Αν η αλυσίδα Markov βρεθεί στην κατάσταση αυτή, τότε είναι αδύνατο να διαφύγει, ή με άλλα λόγια όλες οι πιθανότητες μετάβασης από μια κατάσταση απορρόφησης (absorbing state) σε όλες τις άλλες καταστάσεις είναι 0.

$$I(i,j,k,s) \in \{0,1\} \qquad \text{με } i,j,k \in S,\ s \in \{1,...,L-1\}$$

Τα i, j είναι τα χρωμοσώματα που έχουν επιλεγεί σαν γονείς, το k είναι το χρωμόσωμα- απόγονος, και το s είναι το σημείο που γίνεται ο διαχωρισμός των χρωμοσωμάτων. Δεδομένου ότι δε θα μας χρειαστούν οι τύποι που εκφράζουν τις πιθανότητες μετάβασης, θα περιοριστούμε στην παρουσίαση της διασταύρωσης με διαχωρισμό σε ένα και τυχαίο σημείο. Η ανάλυση στις υπόλοιπες παραλλαγές του τελεστή διασταύρωσης, είναι ανάλογη. Έτσι λοιπόν, το s παίρνει μια από τις τιμές $\{1,...L-1\}$,με ίση πιθανότητα. Η $I(\cdot)$ επιστρέφει 1 αν πρόκειται για το 1ο παιδί (το αριστερό τμήμα του παιδιού λαμβάνεται από τον i) και 0 αν πρόκειται για το 2ο παιδί (το δεξιό τμήμα του παιδιού λαμβάνεται από τον i).

Έτσι η πιθανότητα να πάρουμε ένα συγκεκριμένο χρωμόσωμα, με χρήση επιλογής με roulette wheel και τελεστή διασταύρωσης με διαχωρισμό σε τυχαίο σημείο είναι

$$P_2(k|\overline{n}) = \sum_{i\in S}\sum_{j\in S} P_1(i|\overline{n})P_1(j|\overline{n})\frac{1}{L-1}\sum_{s=1}^{L-1} I(i,j,k,s) \qquad \text{[Riechmann 1999]}$$

Σε κάποιες παραλλαγές γενετικών αλγορίθμων, ο τελεστής διασταύρωσης εφαρμόζεται με πιθανότητα x. Διαφορετικά οι γονείς περνάνε στην επόμενη γενιά, χωρίς μεταβολές. Σ΄ αυτήν την περίπτωση η πιθανότητα θα είναι

$$P_2(k|\overline{n}) = (1-x)P_1(i|\overline{n}) + x\sum_{i\in S}\sum_{j\in S} P_1(i|\overline{n})P_1(j|\overline{n})\frac{1}{L-1}\sum_{s=1}^{L-1} I(i,j,k,s)$$

Η πιθανότητα να πάρουμε έναν πληθυσμό $\overline{m}$ από έναν πληθυσμό $\overline{n}$ είναι

$$P_2(\overline{m}|\overline{n}) = \binom{M}{\overline{m}}\prod_{i\in S} P_2(i|\overline{n})^{m(i)}$$

Η διαδικασία είναι παρόμοια με προηγουμένως. Κι εδώ επίσης είναι αδύνατο να διαφύγουμε όταν βρεθούμε σε μια κατάσταση με ένα χρωμόσωμα στον πληθυσμό. Πάλι οι καταστάσεις με ομογενή πληθυσμό είναι καταστάσεις απορρόφησης (absorbing states) της διαδικασίας Markov και μοιραία η διαδικασία θα κατασταλλάξει σε μια από αυτές τις καταστάσεις, μετά την παρέλευση αρκετών χρονικών βημάτων.

Η οικονομική ερμηνεία του τελεστή διασταύρωσης εξαρτάται από τον τρόπο που έχει κατασκευαστεί το όλο υπόδειγμα. Στην περίπτωση που έχουμε ένα πληθυσμό για όλους τους πράκτορες -όπως στον αλγόριθμο του Vriend (2000), το κάθε χρωμόσωμα (ή η κάθε ομάδα χρωμοσωμάτων, μέσα στον πληθυσμό) απεικονίζει την επιλογή της συγκεκριμένης επιχείρησης. Σ΄ αυτήν την περίπτωση, ο τελεστής διασταύρωσης συνδιάζει τις επιλογές 2 επιχειρήσεων (εκ των οποίων η μια θα μπορούσε να είναι η ίδια) για να δημιουργήσει την νέα στρατηγική της επιχείρησης. Συνεπώς ο τελεστής διασταύρωσης παίζει το ρόλο της μίμησης και για αυτόν τον λόγο τα μοντέλα με ένα πληθυσμό, λέγονται και μοντέλα κοινωνικής μάθησης (social learning).

Στην περίπτωση που κάθε πράκτορας διαθέτει το δικό του πληθυσμό χρωμοσωμάτων και εκτελεί τον δικό του γενετικό αλγόριθμο (όπως στους αντίστοιχους αλγόριθμους της Arifovic(1994)), ο τελεστής διασταύρωσης ανασυνδιάζει τα διαφορετικά χρωμοσώματα του πληθυσμού του ίδιου πράκτορα, τα οποία θα μπορούσαν να θεωρηθούν σαν εναλλακτικές στρατηγικές και συνεπώς αποτελεί μια μέθοδο λογισμού και λήψης απόφασης για την ίδια την επιχείρηση. Τα μοντέλα αυτού του τύπου λέγονται και μοντέλα ατομικής μάθησης (individual learning).

'Οσον αφορά τον τρίτο τελεστή, αυτόν της μετάλλαξης, η αλλαγή ενός bit στο χρωμόσωμα της στρατηγικής, υποδηλώνει ουσιαστικά, τον «πειραματισμό», την δοκιμαστική δηλαδή υιοθέτηση εναλλακτικών στρατηγικών, η οποία είναι απαραίτητη στη διαδικασία μάθησης, όχι μόνο κατ΄ αρχήν, αλλά και για τους λόγους που αναλύσαμε παραπάνω, όσον αφορά τη σύγκλιση του γενετικού αλγόριθμου. Επίσης, μπορεί να θεωρηθεί ότι η μετάλαξη απεικονίζει την καινοτομία, δεδομένου ότι μπορεί να οδηγήσει στην ανακάλυψη εντελώς νέων στρατηγικών, που δεν έχουν χρησιμοποιηθεί ποτέ στο παρελθόν.

Δεδομένου ότι το ενδεχόμενο μετάλλαξης ενός δυαδικού ψηφίου είναι ανεξάρτητο με το ενδεχόμενο μετάλλαξης κάποιου άλλου, η πιθανότητα μετάβασης, μέσω της μετάλλαξης, από ένα χρωμόσωμα i σε ένα άλλο j, εξαρτάται άμεσα από τον αριθμό των διαφορετικών bits στα δύο χρωμοσώματα (την απόσταση Hamming, όπως αλλιώς ονομάζεται). Έτσι για πιθανότητα μετάλλαξης μ και απόστασης Hamming $H(i,j)$, η πιθανότητα μετάβασης από το i στον j μέσω της μετάλλαξης είναι

$$\mu^{H(i,j)}(1-\mu)^{(L-H(i,j))}$$

Έτσι, στην περίπτωση διαδοχικής εφαρμογής ανασυνδιασμού (crossover) και μετάλλαξης, η πιθανότητα μετάβασης απ΄ το i στο j είναι

$$P_3(i|\overline{n}) = \sum_{j\in S} \mu^{H(i,j)}(1-\mu)^{(L-H(i,j))} P_2(j|\overline{n})$$

Τέλος η πιθανότητα μετάβασης απ΄ τον πληθυσμό $\overline{n}$ στον πληθυσμό $\overline{m}$ είναι

$$P_3(\overline{m}|\overline{n}) = \binom{M}{\overline{m}} \prod_{i\in S} P_3(i|\overline{n})^{m(i)}$$

Σε αντίθεση με προηγουμένως, η εισαγωγή του τελεστή μετάλλαξης δεν έχει καταστάσεις απορρόφησης (absorbing states). Ο τελεστής μετάλλαξης, μπορεί να οδηγήσει στον οποιονδήποτε πληθυσμό, με πιθανότητα τόσο μεγαλύτερη, όσο μικρότερη είναι η απόσταση Hamming, μεταξύ του ενδιάμεσου πληθυσμού που παράγει ο τέλεστης μετάλλαξης και του τελικού αυτού πληθυσμού. Για οποιονδήποτε πάντως πληθυσμό, η πιθανότητα είναι θετική, δεδομένου ότι πάντα υπάρχει η πιθανότητα να αντιστραφούν οι τιμές ακόμα και όλων των ψηφίων ενός πληθυσμού. Έτσι ο τελεστής μετάλλαξης μπορεί να μας οδηγήσει από τον οποιονδήποτε αρχικό πληθυσμό, στον οποιονδήποτε άλλο, με την πιθανότητα μετάβασης να εξαρτάται αποκλειστικά από τους δύο αυτούς πληθυσμούς. Έχουμε δηλαδή μια κανονική εργόδικη[11] αλυσίδα Markov (canocical ergodic Markov chain). Ως γνωστόν, για όλες τις εργόδικες αλυσίδες Markov, υπάρχει ένα διάνυσμα, που περιγράφει τη συχνότητα εμφάνισης μιας κατάστασης, η οποία παραμένει (σχεδόν) σταθερή, μετά από έναν κάποιο αριθμό χρονικών βημάτων. Η πιθανότητες δηλαδή μετάβασης από μια δεδομένη κατάσταση, σε μια άλλη, μετά από m βήματα, τείνει σε μια σταθερή τιμή, ανεξάρτητη από την αρχική κατάσταση. Πιο φορμαλιστικά, αν $P = \{p_{ij}\}$ είναι η μήτρα των πιθανοτήτων μετάβασης p_{ij} από την κατάσταση i στην κατάσταση j, σε ένα βήμα, τότε η P^n δίνει τις πιθανότητες μετάβασης σε $n-$ βήματα. Για μια εργόδικη αλυσίδα Markov,

$$\lim_{n \to \infty} P^n = W$$

με την W μια μήτρα, της οποίας όλες οι γραμμές είναι ίδιες $\mathbf{w}$. Το διάνυσμα $\mathbf{w}$ (ή $\boldsymbol{\pi}$) δείχνει τις πιθανότητες μετάβασης από την οποιαδήποτε αρχική κατάσταση στην οποιαδήποτε άλλη, μετά από n βήματα. Το γεγονός ότι τείνει σε μια σταθερή τιμή, υποδηλώνει ότι η πιθανότητα εμφάνισης της όποιας κατάστασης (και άρα και η συχνότητα) είναι σταθερή και ανεξάρτητη της αρχικής κατάστασης. Το διάνυσμα αυτό μπορεί να εκτιμηθεί με βάση τις παρατηρήσεις ενός δείγματος της αλυσίδας Markov.

Διαπιστώνουμε λοιπόν, ότι η εισαγωγή του τελεστή μετάλλαξης οδηγεί σε μια εξελικτική ευστάθεια που χαρακτηρίζεται από τον ορισμό του Ljapunov. Η αλυσίδα Markov δηλαδή, δεν συγκλίνει σε μια συγκεκριμένη τιμή, αλλά παρουσιάζει μια ευστάθεια με την γενικότερη έννοια, αφού όλες οι καταστάσεις με $\mathbf{w}_i > 0$, παρουσιάζονται, με δεδομένη συχνότητα , ανάλογη με τη $\mathbf{w}_i$

.

[11] Υπενθυμίζουμε ότι μια αλυσίδα Markov λέγεται εργόδικη αν μπορούμε να μεταβούμε από όλες τις καταστάσεις τις σε οποιανδήποτε άλλη. Αν μάλιστα για όλες τις αρχικές καταστάσεις, υπάρχει N ώστε $\forall\, n > N$ η πιθανότητα να βρεθούμε σε οποιανδήποτε άλλη κατάσταση της αλυσίδας μετά από n βήματα (η P_{ij}^n δηλαδή) είναι μεγαλύτερη του μηδενός (ή με άλλα λόγια, η αλυσίδα δεν έχει κύκλους- δεν υπάρχει δηλαδή μια περιοδική εναλλαγή μεταξύ διαφορετικών συνόλων εφικτών κατάστασεων), η εργόδικη αλυσίδα ονομάζεται κανονική.

2.4.4. Μάθηση μέσω γενετικών αλγορίθμων και εξελικτικά οικονομικά παίγνια

Στην ενότητα αυτή θα διερευνήσουμε τη συσχέτιση μεταξύ της μάθησης μέσω γενετικών αλγορίθμων και της εξελικτικής θεωρίας παιγνίων. Θα χρησιμοποιήσουμε τον «κανονικό» γενετικό αλγόριθμο (canonical genetic algorithm) [Goldberg 1988], που αναλύσαμε και παραπάνω σαν αλγόριθμο αναφοράς, για να επισημάνουμε ότι η εφαρμογή των γενετικών αλγόριθμων σε οικονομικά παίγνια είναι στην πραγματικότητα ένα εξελικτικό δυναμικό παίγνιο, στο οποίο ο πληθυσμός τείνει να συγκλίνει σε μια κατάσταση που μπορεί να θεωρηθεί σαν ισορροπία. Η διαφορά ενός «οικονομικού» γενετικού αλγόριθμου συνίσταται, στο γεγονός ότι η συνάρτηση βιολογικής προσαρμοστικότητας (fitness function), εξαρτάται και από τις στρατηγικές των υπολοίπων επιχειρήσεων, εξαρτάται από το σύνολο του πληθυσμού που έχει δηλαδή, στην κάθε χρονική στιγμή ο αλγόριθμος, είναι όπως λέμε εξαρτώμενος από την κατάσταση (state-dependent).

Οι καταστάσεις στις οποίες βρίσκει εφαρμογή η θεωρία παιγνίων, είναι καταστάσεις στις οποίες το κέρδος του κάθε πράκτορα, εξαρτάται από την στρατηγική που υιοθετεί ο κάθε άλλος συμπαίκτης. Συνεπώς ένας γενετικός αλγόριθμος μπορεί να θεωρηθεί (και να μοντελοποιηθεί) σαν παίγνιο. Το γεγονός μάλιστα ότι έχουμε παραπάνω από μια γενεές, οδηγεί στο συμπέρασμα ότι πρόκειται για δυναμικό παίγνιο. Αν λοιπόν έχουμε m πράκτορες στον πληθυσμό και n διαφορετικές επιλογές διακριτών ποσοτήτων για κάθε πράκτορα, πρόκειται για ένα επαναλαμβανόμενο παίγνιο m παικτών με n καθαρές στρατηγικές. Πιο συγκεκριμένα, αν τα χρωμοσώματα έχουν L δυαδικά ψηφία, τότε το πλήθος των στρατηγικών είναι $n = 2^L$.

Βλέπουμε λοιπόν ότι η εξελικτική διαδικασία ενός γενετικού αλγόριθμου, μπορεί να θεωρηθεί παρόμοια με αυτή ενός δυναμικού παιγνίου. Ωστόσο, η συμπεριφορά των πρακτόρων σε ένα επαναλαμβανόμενο παίγνιο είναι εντελώς διαφορετική. Οι παίκτες επιλέγουν τη στρατηγική εκείνη, που μεγιστοποιεί το αναμενόμενο κέρδος τους, ενώ σε έναν γενετικό αλγόριθμο αυτό που έχει σημασία είναι το σχετικό κέρδος της κάθε στρατηγικής σε σχέση με το μέσο κέρδος ολόκληρου του πληθυσμού (τη σχετική προσαρμοστικότητα).

Οι Marimon et al (1993) έδωσαν μια σαφή περιγραφή των ομοιοτήτων μεταξύ των διαδικασιών μάθησης από τη μία και των εξελικτικών διαδικασιών από την άλλη. Δεδομένου ότι οι γενετικοί αλγόριθμοι αποτελούν διαδικασίες μάθησης (βλ. και Dawid (1999)), θα πρέπει να αποτελούν και εξελικτική διαδικασία. Η εξελικτική οικονομική θεωρία και οι γενετικοί αλγόριθμοι έχουν την ίδια φιλοσοφία. Και οι δύο ασχολούνται με έναν πληθυσμό οικονομικών πρακτόρων που συναγωνίζονται, αποσκοπώντας στη μεγιστοποίηση του κέρδους τους, μέσα σε ένα δεδομένο περιβάλλον. Χρησιμοποιώντας τον ορισμό του Friedman (1998) για τα εξελικτικά παίγνια

«Ένα εξελικτικό παίγνιο είναι μια μορφή στρατηγικής αλληλεπίδρασης μεταξύ n παικτών, με τις εξής τρεις ιδιότητες : α) στρατηγικές με υψηλότερα κέρδη έχουν μεγαλύτερη πιθανότητα να αντικαταστήσουν στρατηγικές με χαμηλότερα κέρδη, καθώς το παίγνιο εξελίσσεται, β) υπάρχει αδράνεια και γ) οι παίκτες δεν επηρρεάζουν σκόπιμα τις μελλοντικές επιλογές των υπολοίπων παικτών» (Friedman 1998),

ο Riechmann (2001) ήταν σε θέση να δείξει ότι ένας γενετικός αλγόριθμος είναι ένα εξελικτικό παίγνιο. Είναι προφανές ότι ο μηχανισμός της φυσικής επιλογής που χαρακτηρίζει τους γενετικούς αλγόριθμους, ικανοποιεί την 1η συνθήκη του Friedman. Αυτό είναι άλλωστε το περιεχόμενο του θεωρήματος σχήματος (schema theorem), του θεμελιώδους θεωρήματος της θεωρίας των γενετικών αλγορίθμων, που παρουσιάσαμε παραπάνω. Όσον αφορά τη δεύτερη συνθήκη, ο Friedman την συγκεκριμενοποιεί λέγοντας «...οι αλλαγές στη δομή του πληθυσμού, δεν συμβαίνουν απότομα» (Friedman1998, p. 16). Οι γενετικοί αλγόριθμοι έχουν και αυτήν την ιδιότητα, αφού ο πληθυσμός ενός γενετικού αλγόριθμου, εξελίσσεται βαθμιαία, μέσω των τελεστών διασταύρωσης και μετάλλαξης, με τα χρωμοσώματα της κάθε γενιάς να σχετίζονται άμεσα με τον πληθυσμό της αμέσως προηγούμενης γενεάς. Για τις συνήθεις, μικρές τιμές της πιθανότητας μετάλλαξης, η ομοιότητα των δύο γενεών είναι αρκετά μεγάλη, ιδιαίτερα όταν έχουμε φτάσει την περιοχή της εργοδικής διαδικασίας Markov που χαρακτηρίζει τον γενετικό αλγόριθμο, κατά την οποία οι πιθανότητες μετάβασης είναι -σχεδόν- σταθερές (Riechmann 2001). Αλλά και η 3η συνθήκη ικανοποιείται από έναν γενετικό αλγόριθμο, αφού οι πράκτορες αποσκοπούν στη μεγιστοποίηση της δικής τους προσαρμοστικότητας, ενώ τα χρωμοσώματα αναβαθμίζονται με έμμεσο τρόπο, μέσω της σύγκρισης της προσαρμοστικότητάς τους με τη μέση προσαρμοστικότητα του πληθυσμού.

Όπως αναφέρθηκε προηγουμένως, κάθε πράκτορας σε έναν γενετικό αλγόριθμο (υποθέτουμε ότι το κάθε χρωμόσωμα αντιστοιχεί στην στρατηγική ενός πράκτορα) προσπαθεί να μεγιστοποιήσει τη σχετική προσαρμοστικότητά του, με δεδομένη την προσαρμοστικότητα του πληθυσμού. Αποσκοπεί δηλαδή στην μεγιστοποίηση $\max_{i \in S} R(i|\overline{n})$. Αυτό δείχνει ότι οι πράκτορες ακολουθούν τη λογική Nash, έστω και ασυνείδητα. Προσπαθούν δηλαδή να μεγιστοποιήσουν το ιδίον όφελος, με δεδομένες τις στρατηγικές επιλογές των συμπαικτών τους. Ο Riechmann(2001) επιχειρηματολογεί -χωρίς να δίνει απόδειξη- υπέρ του ότι «κάθε κανονικός οικονομικός γενετικός αλγόριθμος [στον οποίο κάθε χρωμόσωμα του πληθυσμού αντιστοιχεί σε μια στρατηγική επιλογή ενός πράκτορα], τείνει στην ισορροπία κατά Nash, χωρίς ποτέ να καταλήγει σε αυτήν». Ο λόγος που δεν καταλήγουμε στην ισορροπία κατά Nash, είναι βέβαια ότι ο τελεστής μετάλλαξης εισάγει συνεχώς νέες στρατηγικές στον πληθυσμό και συνεπώς -όπως έχουμε ήδη πει- έχουμε μια συνεχή διαφοροποίηση της κατάστασης του αλγορίθμου. Βεβαίως όπως θα δούμε παρακάτω στις περισσότερες περιπτώσεις που μελετάμε εδώ, οι αλγόριθμοι δεν συγκλίνουν στην ισορροπία κατά Nash . Από την άλλη, σύμφωνα και με το θεώρημα σχήματος (schema theorem), οι τελεστές της διασταύρωσης και της μετάλλαξης λειτουργούν έτσι ώστε τα χρωμοσώματα με μεγαλύτερη σχετική προσαρμοστικότητα, έχουν σημαντικά μεγαλύτερη πιθανότητα να αναπαραχθούν και να μεταδώσουν το γενετικό τους υλικό στις επόμενες γενιές. Αυτό σημαίνει ότι λειτουργούν προς όφελος των εξελικτικά ευσταθών στρατηγικών, οι οποίες είναι εξ΄ ορισμού οι βέλτιστες επιλογές για δεδομένες επιλογές των συμπαικτών, στην προηγούμενη γενιά.

2.4.5. Εξελικτική ευστάθεια των γενετικών αλγορίθμων

Η έννοια της εξελικτικής ευστάθειας (ESS) που αναλύσαμε σε προηγούμενη ενότητα, έχει κάποιες αδυναμίες, όταν πρέπει να εφαρμοστεί σε γενετικούς αλγόριθμους. Ο λόγος είναι ότι δεν ορίζεται επακριβώς, ο τρόπος που αποβάλλονται οι ανεπιτυχείς στρατηγικές από τον πληθυσμό. Πιο συγκεκριμένα, ο ορισμός της εξελικτικής ευστάθειας, προϋποθέτει ότι οι μεταλλαγμένες στρατηγικές αποβάλλονται με κάποιο τρόπο από τον πληθυσμό, έτσι ώστε ο πληθυσμός να αποτελείται μονάχα από εκείνες τις στρατηγικές που «κερδίζουν» έτσι τον χαρακτηρισμό των εξελικτικά ευσταθών, στη στάσιμη κατάσταση. Αντίθετα, ένας γενετικός αλγόριθμος που χαρακτηρίζεται από εργόδικη αλυσίδα Markov, δεν πρόκειται ποτέ να συγκλίνει σε μια κατάσταση, στην οποία οι στρατηγικές που θα παραμείνουν στον πληθυσμό, είναι στάσιμες. Πάντοτε θα υπάρχει πιθανότητα μια μεταλλαγμένη στρατηγική να εισαχθεί στον πληθυσμό, παραγκωνίζοντας μια άλλη στρατηγική, ακόμα κι αν η προσαρμοστικότητα της δεύτερης είναι πολύ υψηλότερη.

Συνεπώς πρέπει να υπάρξει μια προσαρμογή στην έννοια της εξελικτικής ευστάθειας, ώστε να γίνει κατάλληλη για την ανάλυση γενετικών αλγορίθμων. Ένας τρόπος να γίνει αυτό, πάντα μέσα στο πνεύμα της ESS, είναι να θεωρήσουμε ότι «ένας γενετικός αλγόριθμος είναι εξελικτικά σταθερός, αν η [εξελικτική] διαδικασία του αλγορίθμου, αποβάλλει μια εισβολή μεταλλαγμένων στρατηγικών στον πληθυσμό. Η εισβολή αυτή, αναφέρεται είτε σε μια εντελώς νέα στρατηγική που μπαίνει στον πληθυσμό, είτε απλά σε μια αλλαγή στις συχνότητες των στρατηγικών που απαρτίζουν τον πληθυσμό σε κάθε χρονικό βήμα. Έτσι ένας καταλληλότερος ορισμός θα μπορούσε να είναι : ένας γενετικός αλγόριθμος είναι ESS αν ανθίσταται στις αλλαγές στη σύνθεσή του» [Riechmann 2001].

Πριν παρουσιάσουμε αναλυτικότερα τις προσθήκες που προτείνει ο Riechmann (2001) θεωρούμε σκόπιμο να αναφερθούμε σε μια επαναδιατύπωση του κλασικού ορισμού, όπως τον δίνει ο Weibull (2006), καθώς και σε μια πρόταση του Schaffer (1998), σχετική με την εξελικτική ευστάθεια σε πεπερασμένους πληθυσμούς. Αν λοιπόν συμβολίσουμε την στρατηγική του παίκτη i με q_i και τις επιλογές των υπολοίπων παικτών με Q_{-i}, τότε, σύμφωνα με τον Weibull (2006), ένα προφίλ στρατηγικών $Q^* = \left(q_1^*,...,q_n^*\right)$ είναι εξελικτικά σταθερό, αν και μόνο αν ικανοποιούνται οι δύο επόμενες συνθήκες

$$\pi_i\left(q_i, Q_{-i}^*\right) \le \pi_i\left(q_i^*, Q_{-i}^*\right) \qquad \forall\, q_i, i = 1,...,n$$

$$\pi_i\left(q_i, Q_{-i}^*\right) = \pi_i\left(q_i^*, Q_{-i}^*\right) \Rightarrow \pi_i\left(q_i, Q_{-i}\right) = \pi_i\left(q_i^*, Q_{-i}\right) \qquad \forall\, Q_{-i} \ne Q_i, i = 1,..,n$$

Πρόκειται για τον κλασικό ορισμό, επανδιατυπωμένο για την περίπτωση παιγνίων πολλαπλών παικτών. Σημειώνουμε ότι η πρώτη συνθήκη συνεπάγεται ότι μια εξελικτικά ευσταθής στρατηγική είναι ισορροπία κατά Nash. Το αντίστροφο όμως δεν ισχύει, δεν είναι όλες οι ισορροπίες κατά Nash, εξελικτικά ευσταθείς, όπως παρουσιάσαμε και στο παράδειγμα της αντίστοιχης ενότητας.

Ο Schaffer(1988), διαπίστωσε ότι στις περιπτώσεις πληθυσμών με πεπερασμένο αριθμό στρατηγικών, μπορούν κάποιες στρατηγικές που δεν ικανοποιούν τον παραπάνω ορισμό να αποτελούν σταθερά συστατικά του πληθυσμού καθώς αυτός εξελίσσεται και πρότεινε τον εξής εναλλακτικό ορισμό για μια εξελικτικά ευσταθή στρατηγική : «αν ο παίκτης j παίζει μια μεταλλαγμένη στρατηγική $q_j^m \ne q^*{}_j$, τότε το προφίλ στρατηγικών q^* θα είναι εξελικτικά ευσταθές αν

$$\pi_i\left(q_i^*, q_j^m, Q_{-j-i}^*\right) \ge \pi_i\left(q_i, q_j^m, Q_{-j-i}^*\right) \qquad \forall\, q_i \;\&\; \forall\, i \ne j \;\&\; \forall\, q_j^m \ne q_j^* \quad \text{Schaffer(1988)»}$$

Σύμφωνα με τον παραπάνω ορισμό, μια εξελικτικά ευσταθής στρατηγική δεν είναι κατ΄ ανάγκη ισορροπία κατά Nash.

Επιστρέφουμε τώρα στην πρόταση του Riechmann(2001). Ο Riechmann δίνει πρώτα τον ορισμό του εξελικτικά ανώτερου πληθυσμού (evolutional superior population). Ο πληθυσμός $\overline{n}$ είναι εξελικτικά ανώτερος από τον $\overline{m}$ (συμβολικά $\overline{n} \overset{es}{>} \overline{m}$) αν ικανοποιούνται οι εξής δύο συνθήκες:

a) Κάθε στρατηγική του $\overline{n}$ περιέχεται στον $\overline{m}$ και έχει τουλάχιστον την ίδια προσαρμοστικότητα στον $\overline{n}$, με την προσαρμοστικότητα που έχει στον $\overline{m}$, ενώ τουλάχιστον μια από τις κοινές στρατηγικές έχει μεγαλύτερη προσαρμοστικότητα στον $\overline{n}$ από ότι στον $\overline{m}$.

b) Οι μη κοινές στρατηγικές $k \in \left\{\overline{m} - \overline{n}\right\}$, έχουν χαμηλότερη προσαρμοστικότητα από ότι οι κοινές στρατηγικές, οπότε τείνουν να αποβληθούν.

Ο μαθηματικός φορμαλισμός των παραπάνω έχει ως εξής :

a) $R\left(i|\overline{n}\right) \ge R\left(i|\overline{m}\right) \qquad \forall\, i \in \overline{n}$

b) $\exists j$ με $R(j|\overline{n}) > R(j|\overline{m})$

c) $R(k|\overline{m}) < R(i|\overline{m}) \quad \forall i \in \overline{n}, \forall k \in \{\overline{m} - \overline{n}\}$

Τα παραπάνω χαρακτηρίζουν μια μορφή ασθενούς κυριαρχίας (weak dominance). Ο παραπάνω ορισμός οδηγεί σε μια ασθενή διάταξη (partial ordering) του χώρου των πληθυσμών.

Για να κατανοήσουμε τα παραπάνω, ας αναφερθούμε σε ένα παράδειγμα από το Riechmann(2001). Ας υποθέσουμε ότι έχουμε ένα Cournot game, και ότι το διάνυσμα στρατηγικών $\overline{n} = (3,2,2,3,2)$ οδηγεί σε κέρδη για όλες τις επιχειρήσεις. Σημειώνουμε ότι ο κάθε αριθμός υποδηλώνει την ποσότητα που παράγει ο κάθε πράκτορας. Ας υποθέσουμε τώρα ότι η δεύτερη επιχείρηση αρχίζει να πειραματίζεται και μεταβάλλει την παραγωγή της σε $q_2 = 10$. Η αύξηση της συνολικής ποσότητας, οδηγεί σε μείωση των κερδών των επιχειρήσεων, λόγω της μείωσης της τιμής. Ας υποθέσουμε ότι η μείωση των κερδών είναι τέτοια, ώστε να είναι προς το συμφέρον μιας εκ των επιχειρήσεων να αλλάξουν στρατηγική. Εννοείται ότι οι τελεστές διασταύρωσης και μετάλλαξης, θα μας οδηγήσουν κάποια στιγμή να διαφύγουμε από τον πληθυσμό $\overline{m} = (3,10,2,3,2)$, χωρίς βέβαια να υπάρχει εγγύηση ότι θα επιστρέψουμε στον πληθυσμό $\overline{n}$. Κατ΄ αυτήν την έννοια, ο $\overline{n}$ είναι καλύτερος από τον $\overline{m}$, και για αυτό και χαρακτηρίζεται εξελικτικά ανώτερος.

Ακολουθεί ο ορισμός των γενετικά ευσταθών πληθυσμών (evolutionary stable populations). Ένας πληθυσμός $\overline{n}$ θα λέγεται εξελικτικά ευσταθής, αν δεν υπάρχει άλλος πληθυσμός $\overline{m}$ ώστε

να είναι εξελικτικά ανώτερος από τον $\overline{m}$. Δηλαδή

$$\text{δεν υπάρχει } m \in S \text{ ώστε } \overline{m} \overset{es}{>} \overline{n}$$

Ο ορισμός αυτός είναι μια γενίκευση του «παραδοσιακού» ορισμού. Σημειώνουμε ότι, σύμφωνα με ορισμό των εξελικτικά ανώτερων πληθυσμών, θα μπορούσαμε να έχουμε περισσότερους από έναν εξελικτικά ευσταθείς πληθυσμούς. Όταν φτάσουμε σε έναν τέτοιο πληθυσμό, δύο πράγματα μπορούν να συμβούν, μετά από την «μόλυνσή» του από μια μεταλλαγμένη στρατηγική: «είτε η στρατηγική να απορριφθεί και να επιστρέψουμε στον αρχικό πληθυσμό, είτε οι μεταβολές που θα προκληθούν να οδηγήσουν σε κάποιον άλλο, εξελικτικά ευσταθή πληθυσμό» (Riechmann 2001).

Δεδομένου ότι σε έναν γενετικό αλγόριθμο, η διαδικασία επιλογής ενός χρωμοσώματος είναι πιθανοκρατική- η πιθανότητα επιβίωσης είναι ανάλογη της προσαρμοστικότητας- ενδεχομένως να χρειαστούν αρκετές γενιές μέχρις ότου απορριφθεί μια μεταλλαγμένη στρατηγική, από έναν εξελικτικά ευσταθή πληθυσμό. Αυτό που μπορούμε να υποστηρίξουμε είναι ότι μακροπρόθεσμα, θα οδηγηθούμε σε έναν εξελικτικά ευσταθή πληθυσμό. Και, όπως είπαμε και στην προηγούμενη ενότητα, η εξελικτική δυναμική ενός γενετικού αλγορίθμου με εργοδική αλυσίδα οδηγεί σε μια ευσταθή κατά Ljapunov διαδικασία. Διαφορετικές καταστάσεις εμφανίζονται, με σταθερή όμως συχνότητα μακροπρόθεσμα, καθώς ο αλγόριθμος μεταβαίνει από μια εξελικτικά σταθερή κατάσταση σε νοθευμένες καταστάσεις, απεικονίζοντας ακριβώς αυτήν την διαδικασία επανασταθεροποίησης στην εξελικτικά σταθερή κατάσταση.

2.4.6. Οικονομική ερμηνεία των μοντέλων μάθησης με βάση τους γενετικούς αλγορίθμους

Πρώτα από όλα είναι σημαντικό να κατανοήσουμε τη σημασία του πληθυσμού του γενετικού αγλορίθμου, τόσο όσον αφορά την εξελικτική θεωρία παιγνίων, όσο και τη θεωρία μάθησης. Στην θεωρία παιγνίων, ο πληθυσμός περιγράφει τις επιλογές ενός συνόλου παικτών και το κάθε χρωμόσωμα αντιστοιχεί σε μια από αυτές τις στρατηγικές. Συνήθως κάθε χρωμόσωμα αντιστοιχεί σε -μία ή περισσότερες- επιλογές ενός συγκεκριμένου παίκτη. Ο κάθε παίκτης προσπαθεί να μεγιστοποιήσει το ιδίον όφελος με δεδομένες τις επιλογές των υπολοίπον κάτι που ενδεχομένως οδηγεί την εξελικτική διαδικασία, στην ισορροπία κατά Nash. Ωστόσο δεν είναι καθόλου σίγουρο ότι πραγματικά θα προκύψει κάτι τέτοιο. Η ανάλυση αναφορικά με τις αλυσίδες Markov που προκύπτουν, δείχνει ότι ένας «κανονικός» γενετικός αλγόριθμος θα συγκλίνει σε μία συμπεριφορά που χαρακτηρίζεται από την πραγμάτωση περισσότερων από μια καταστάσεων, με μια δεδομένη συχνότητα. Από την άλλη πλευρά, η έννοια των εξελικτικά σταθερών στρατηγικών μας δίνει ένα επιπλέον εργαλείο για να προσεγγίσουμε την εξελικτική διαδικασία. Μια βασική συνέπειά της είναι ότι ο γενετικός αλγόριθμος θα συγκλίνει σε ένα σύνολο εξελικτικά ευσταθών στρατηγικών, με τη συνεχή όμως «εισβολή» μεταλλαγμένων στρατηγικών, που θα αποβάλλονται σταδιακά, με την παράλληλη συνεχή εισβολή νέων. Οι νέες στρατηγικές είτε θα οδηγήσουν στη βελτίωση της σύνθεσης του πληθυσμού και στην πορεία του προς μια άλλη -καλύτερη- εξελικτικά σταθερή κατάσταση, είτε θα αποβληθούν, μέσω της διαδικασίας της εξέλιξης. Αυτή η κατάσταση συνεχίζεται επ΄ αόριστο.

Όπως ήδη προαναφέραμε, υπάρχουν τρεις μηχανισμοί μάθησης στην εξελικτική διαδικασία ενός γενετικού αλγορίθμου, που εκφράζονται από τις τρεις ξεχωριστές διαδικασίες προσδιορισμού των χρωμοσωμάτων της επόμενης γενεάς, αυτούς της επιλογής των γονέων, του τελεστή διασταύρωσης (crossover) και του τελεστή μετάλλαξης (mutation). Η ερμηνεία τους δε, εξαρτάται γενικότερα, από τη μορφή του γενετικού αλγορίθμου. Διαφορετική είναι η σημασία των τριων αυτών μηχανισμών σε έναν αλγόριθμο ατομικής και διαφορετική σε έναν αλγόριθμο κοινωνικής μάθησης. Η μορφή των τελεστών μάθησης, όπως διαμορφώνεται στα πλαίσια του συγκεκριμένου γενετικού αλγορίθμου, προσδιορίζει τόσο την δυναμική συμπεριφορά του αλγορίθμου, όσο και την ποιότητα των λύσεων που προκύπτουν.

Αναλύσαμε σε προηγούμενη ενότητα, τη δυναμική συμπεριφορά της διαδικασίας επιλογής χρωμοσωμάτων, καθεαυτής. Είδαμε ότι, εφόσον η επιλογή γίνεται με τρόπο ώστε η πιθανότητα επιλογής να είναι ανάλογη με την προσαρμοστικότητα του χρωμοσώματος, τότε, εφόσον δεν λάβουμε υπόψη την επίδραση των άλλων τελεστών, ο πληθυσμός που προκύπτει μακροπρόθεσμα, είναι ομογενής. Αυτό σημαίνει ότι η διαδικασία της επιλογής, οδηγεί στο πλέον ισχυρό αποτέλεσμα, αυτό της πλήρους εξίσωσης των στρατηγικών. Η ερμηνεία της διαδικασίας της επιλογής εξαρτάται από τη μορφή του αλγορίθμου. Εφόσον έχουμε έναν αλγόριθμο στον οποίον ο κάθε πράκτορας έχει έναν ξεχωριστό πληθυσμό χρωμοσωμάτων, τότε, όπως είπαμε παραπάνω, το κάθε χρωμόσωμα αντιστοιχεί σε μια ιδέα του εν λόγω πράκτορα. Έτσι η διαδικασία επιλογής, αναλογεί σε μια προσπάθεια βελτιστοποίησης, στη δεδομένη στιγμή, από την πλευρά του πράκτορα-επιχείρησης. Αν από την άλλη έχουμε έναν αλγόριθμο με κοινό πληθυσμό για όλες τις επιχειρήσεις, τότε η διαδικασία επιλογής αντιστοιχεί σε μια διαδικασία ισχυρής μίμησης (αντιγραφής θα λέγαμε) των πιο επιτυχημένων επιλογών, σε παρελθοντικό χρόνο, των άλλων επιχειρήσεων. Βλέπουμε λοιπόν ότι αυτή η διαδικασία οδηγεί πραγματικά σε εξίσωση των επιλογών όλων των πρακτόρων-επιχειρήσεων. Το ίδιο ισχύει βέβαια και για τον τελεστή διασταύρωσης, μόνο που εδώ έχουμε μια τροποποίηση των επιτυχημένων στρατηγικών, η οποία είναι περισσότερο αποτελεσματική, δεδομένου ότι οδηγεί σε μια διεύρυνση του χώρου των δυνατών λύσεων.

Από την άλλη ο τελεστής μετάλλαξης μπορεί να ερμηνευθεί σαν μια διαδικασία αναζήτησης εντελώς νέων λύσεων, σαν καινοτομία. Είδαμε παραπάνω ότι ο τελεστής οδηγεί σε συνεχείς τυχαίες διακυμάνσεις που αντιστοιχούν στις συνεχείς καινοτομίες που εφόσον αποδειχτούν αποτελεσματικές, καθιερώνονται στον πληθυσμό, ενώ αν αποδειχτούν αναποτελεσματικές, απορρίπτονται. Οι καινοτομίες αυτές είναι απαραίτητες για να εισαχθούν εντελώς καινούργιες δυνητικές καταστάσεις και να οδηγηθούμε -ενδεχομένως- σε μια ολικά βέλτιστη λύση.

Βλέποντας συνολικά τον γενετικό αλγόριθμο, διαπιστώνουμε ότι μπορεί να συμπεριλάβει και τις τρεις αυτές διαδικασίες που, από κοινή λογική και μόνο, γνωρίζουμε ότι υιοθετούνται κατά τη διαδικασία λήψης αποφάσεων. Ωστόσο για να είναι εφικτό αυτό στην πράξη, θα πρέπει το όλο υπόδειγμα να σχεδιαστεί κατάλληλα. Αν για παράδειγμα αρκεστούμε σε έναν πληθυσμό για όλους τους πράκτορες, συμπεριλαμβάνουμε τις έννοιες της «κοινωνικής μάθησης», της μίμησης και της καινοτομίας, αλλά το μοντέλο μας δεν εμπεριέχει την έννοια της βελτιστοποίησης, μέσα στο σύνολο των εναλλακτικών στρατηγικών που λαμβάνει υπόψη του ο κάθε πράκτορας. Αν από την άλλη, χρησιμοποιήσουμε αποκλειστικά έναν γενετικό αλγόριθμο -με ξεχωριστό πληθυσμό εννοείται- για κάθε πράκτορα, μας διαφεύγει η έννοια της μίμησης, μεταξύ των πρακτόρων.

Οφείλω να πω εδώ, ότι όλα τα υποδείγματα που έχουν προταθεί μέχρι στιγμής, πάσχουν από το ένα ή από το άλλο φαινόμενο.

3. Παρουσίαση σημαντικότερων δημοσιεύσεων

Η πρώτη απόπειρα εφαρμογής γενετικών αλγορίθμων στην μοντελοποίηση των στρατηγικών των παικτών σε ένα $n-$ παίγνιο Cournot έγινε από την Arifovic(1994). Τόσο στην μελέτη της Arifovic, όσο και σε όλες όσες ακολούθησαν (με εξαίρεση το υπόδειγμα των Alkemade et al (2007)), παρατηρείται η μη σύγκλιση των στρατηγικών των παικτών στην ισορροπία κατά Nash. Πιο συγκεκριμένα, η κατάσταση ισορροπίας είναι συνήθως η ισορροπία κατά Walras (για αναφορές βλ. ολόκληρη τη βιβλιογραφία της ενότητας). Η ισορροπία Walras ή «ανταγωνιστική» ισορροπία, χαρακτηρίζει τις ανταγωνιστικές και όχι τις ολιγοπωλιακές αγορές, δεδομένου ότι οι παίκτες συμπεριφέρονται σαν price-takers, θεωρούν δηλαδή την τιμή ισορροπίας σαν εξωγενή, σταθερή μεταβλητή, και όχι σαν μια συνάρτηση της ποσότητας που αποφασίζουν να παράγουν. Αντιδιαστέλλοντας την ισορροπία κατά Walras και την ισορροπία κατά Nash σε σταθερές στρατηγικές (που όλα τα παρακάτω υποδείγματα έχουν, αφού οι συναρτήσεις κέρδους είναι κοίλες συναρτήσεις) μπορούμε να πούμε ότι στην ισορροπία κατά Walras, οι παίκτες τείνουν να μεγιστοποιούν την

$$\max_{q_i} \Pi = \max_{q_i} \{ P q_i - c_i(q_i) \}$$

με την τιμή ισορροπίας P να θεωρείται σταθερή, αντί της

$$\max_{q_i} \Pi = \max_{q_i} \{ P(q_i) q_i - c_i(q_i) \}$$

που μεγιστοποιούν οι παίκτες στην ισορροπία κατά Nash, και η οποία συμπεριλαμβάνει την επιρροή της ποσότητας που επιλέγουν q_i στην τιμή ισορροπίας.

Οι λόγοι που συμβαίνει αυτό, μπορούν να αποδοθούν στην δυναμική συμπεριφορά των γενετικών αλγορίθμων που χρησιμοποιήθηκαν, όπως θα αναλύσουμε παρακάτω.

Τέλος σημειώνουμε ότι πριν τον Vriend (2000) οι ερευνητές προσέγγιζαν το υπόδειγμα μέσω του πρίσματος του cobweb model (Leontief 1934), το οποίο εστιάζει περισσότερο στη δυναμική του μοντέλου. Εννοείται ότι οι παραδοχές των δύο υποδειγμάτων είναι ισοδύναμες, οπότε τελικά μιλάμε για το ίδιο πράγμα με διαφορετική ορολογία (βλ. και Arifovic και Maschek 2006).

Arifovic, J. (1994): Genetic Algorithm Learning and the Cobweb Model

Η Arifovic ήταν η πρώτη που χρησιμοποίησε γενετικούς αλγόριθμους στην μοντελοποίηση της συμπεριφοράς των επιχειρήσεων στο ολιγοπώλιο Cournot. Στην εν λόγω δημοσίευση, που προκύπτει από τη διδακτορική της διατριβή, μελετάει ένα παίγνιο Cournot n παικτών με γραμμική αντίστροφη συνάρτηση ζήτησης

$$P = A - B\sum_{i=1}^{n} q_i$$

με q_i τις ποσότητες που παράγουν οι n επιχειρήσεις σε κάθε γύρο της προσομοίωσης. Οι συναρτήσεις κόστους είναι κοινές για τις n επιχειρήσεις

$$c_i = xq_i + \frac{1}{2}ynq_i^2$$

Σημειώνουμε ότι τα A, B, x, y είναι θετικές παράμετροι. Βλέπουμε ότι οι συνάρτηση κόστους είναι τετραγωνική και συνεπώς, κυρτή για $x, y > 0$. Άρα το παίγνιο έχει συμμετρική ισορροπία κατά Nash σε καθαρές στρατηγικές (McManus 1964).

Η Arifovic χρησιμοποιεί δύο είδη γενετικών αλγορίθμων: ενός και πολλών πληθυσμών (single και multiple population algorithms αντίστοιχα), με δύο διαφορετικές εκδοχές ο καθένας: την βασική και την εκτεταμένη (augmented) όπως τις ονομάζει.

Ο πρώτος αλγόριθμος είναι ενός πληθυσμού (single-population). Έχουμε έναν πληθυσμό για τις στρατηγικές όλων των παικτών, με το κάθε χρωμόσωμα του πληθυσμού αυτού να αντιστοιχεί σε μία εταιρία. Η τιμή του κάθε χρωμοσώματος, σε κάθε χρονικό βήμα της προσομοίωσης (ή σε κάθε γενιά του γενετικού αλγόριθμου, πράγμα που είναι το ίδιο) απεικονίζει την ποσότητα που παράγει η αντίστοιχη επιχείρηση, στον συγκεκριμένο γύρο. Το χρωμόσωμα βέβαια είναι μια ακολουθία δυαδικών ψηφίων. Συνεπώς πρέπει να γίνει ένας μετασχηματισμός που θα απεικονίζει τον εν λόγω δυαδικό αριθμό, σε έναν πραγματικό αριθμο $q \in [0, q_{\max}]$. Η Arifovic ορίζει για αυτόν τον σκοπό ένα συντελεστή «επανακανονικοποίησης», όπως λέει, ο οποίος είναι ένας αριθμός τέτοιος, ώστε πολλαπλασιασμένος με τον ακέραιο που αντιστοιχεί στον δυαδικό αριθμό που απεικονίζει το χρωμόσωμα και που ισούται με $\sum_{k=1}^{l} a_k 2^{k-1}$ (με a_k το ψηφίο, 0 ή 1, στην θέση k του χρωμοσώματος, και l το συνολικό μήκος του χρωμοσώματος), να δίνει έναν πραγματικό αριθμό $q \in [0, q_{\max}]$. Δηλαδή ο συντελεστής θα είναι

$$\frac{q_{\max}}{\sum_{k=1}^{l} 2^{k-1}}$$

οπότε στην περίπτωση που το χρωμόσωμα είναι 0 (όλα τα bits=0) η ποσότητα θα είναι μηδενική και στην περίπτωση που παίρνει τη μέγιστη τιμή $\sum_{k=1}^{l} 2^{k-1}$ (όλα τα ψηφία = 1), η ποσότητα θα είναι $q_{\max}$. Μια ανάλογη μέθοδο κωδικοποίησης ακολουθούν και οι περισσότεροι από τους ερευνητές που ακολουθούν.

Η συνάρτηση προσαρμοστικότητας (fitness function) του χρωμοσώματος είναι ίση με το κέρδος που αποκομίζει η εταιρία στον συγκεκριμένο γύρο (εδώ θα πρέπει να σημειώσουμε ότι απαραίτητη προϋπόθεση για να ισχύει κάτι τέτοιο είναι το κέρδος να είναι θετικό, κάτι που «ξέφυγε» από την Arifovic, χωρίς ωστόσο να έχει ουσιαστική σημασία, δεδομένου ότι οι ποσότητες στις οποίες συγκλίνει ο αλγόριθμος, ικανοποιούν αυτήν την προϋπόθεση)

$$\pi_i = P_t q_i - c_i$$

Για να υπολογιστεί βέβαια η προσαρμοστικότητα, θα πρέπει πρώτα να προσδιοριστεί η τιμή για τον συγκεκριμένο γύρο. Αποκωδικοποιούνται λοιπόν τα χρωμοσώματα-ποσότητες και στη συνέχεια προσδιορίζεται η τιμή από την αντίστροφη συνάρτηση ζήτησης. Στη συνέχεια υπολογίζεται η προσαρμοστικότητα κάθε στρατηγικής, σύμφωνα με την δεδομένη τιμή και το κόστος που αντιστοιχεί στη συγκεκριμένη ποσότητα, που παράγει η επιχείρηση, όπως δίνεται από το αντίστοιχό της χρωμόσωμα.

Ο αρχικός πληθυσμός είναι -ως συνήθως- τυχαίος. Σε κάθε βήμα της προσομοίωσης τα χρωμοσώματα μεταβάλλονται με τους γνωστούς τελεστές ανασυνδιασμού (crossover) και μετάλλαξης (mutation). Η Arifovic χρησιμοποιεί τον κανόνα της ρουλέτας που προαναφέραμε, για να επιλέξει τους γονείς. Η πιθανότητα δηλαδή του κάθε χρωμοσώματος να επιλεγεί σαν γονέας είναι ανάλογη με την προσαρμοστικότητά του. Αφού επιλεγούν οι δύο γονείς, εφαρμόζεται ο τελεστής διασταύρωσης με διαχωρισμό σε ένα τυχαίο σημείο (random point crossover). Επιλέγεται δηλαδή ένας τυχαίος αριθμός μεταξύ του 1 και του μήκους του χρωμοσώματος, ο οποίος δείχνει το σημείο που θα χωριστούν τα δύο χρωμοσώματα-γονείς. Στη συνέχεια παίρνουμε το «αριστερό» τμήμα του 1^ου^ γονέα και το «δεξιό» τμήμα του 2^ου^ γονέα για να κατασκευάσουμε το 1^ο^ παιδί. Για την κατασκευή του 2^ου^ παιδιού παίρνουμε τα αντίστροφα τμήματα από τους γονείς. Τέλος κάθε δυαδικό ψηφίο του χρωμοσώματος-παιδιού μπορεί να αντιστραφεί (από 0 σε 1 και αντίστροφα) με κάποια πιθανότητα p_{mut} (μετάλλαξη).

Στη βασική εκδοχή του αλγόριθμου μετά την κατασκευή τόσων παιδιών, όσος ο αριθμός των χρωμοσωμάτων του αρχικού πληθυσμού, έχουμε τη νέα γενιά. Στην επαυξημένη εκδοχή, εφαρμόζεται και ένας τελεστής εκλογής (election) όπως τον ονομάζει η Arifovic, ο οποίος αποτελεί μια εκδοχή της εκλογής με τουρνουά, που αναφέραμε στο κεφάλαιο 2. Μετά από την εφαρμογή των τελεστών διασταύρωσης και μετάλλαξης, υπολογίζεται η προσαρμοστικότητα του κάθε παιδιού, η οποία στη συνέχεια συγκρίνεται με τις προσαρμοστικότητες των γονέων και επιλέγονται τα 2 από τα 4 χρωμοσώματα (2 γονείς και 2 παιδιά) με την μεγαλύτερη προσαρμοστικότητα για την επόμενη γενιά.

Στον αλγόριθμο πολλών πληθυσμών (multi-population algorithm), η κάθε εταιρία έχει τον δικό της πληθυσμό χρωμοσωμάτων. Σε κάθε γύρο της προσομοίωσης υπολογίζεται η προσαρμοστικότητα του κάθε χρωμοσώματος, με βάση την τιμή της προηγούμενης περιόδου και επιλέγεται ένα από τα χρωμοσώματα, με πιθανότητα ανάλογη με την προσαρμοστικότητά τους αυτή, σαν η απόφαση της εταιρίας για τη συγκεκριμένη περίοδο. Έτσι η πιθανότητα αυτή δίνεται από την

$$\pi_{i,t}^{j} = \frac{\mu_{i,t}^{j} P_{t-1}}{\sum_{i=1}^{n} \mu_{i,t}^{j} P_{t-1}},$$

με $\mu_{i,t}^{j}$ την «προσωρινή» αυτή προσαρμοστικότητα του χρωμοσώματος i της εταιρίας j, που υπολογίζεται με βάση την τιμή της προηγούμενη περιόδου P_{t-1}. Στη συνέχεια, αφού προσδιοριστούν οι ποσότητες που θα παράγει η κάθε επιχείρηση, μπορεί να προσδιοριστεί η τρέχουσα τιμή, το κόστος και το κέρδος της κάθε επιχείρησης. Η αναπαραγωγή των πληθυσμών γίνεται πάλι με τη χρήση των ίδιων τελεστών διασταύρωσης, μετάλλαξης και της επιλογής τουρνουά, στην εκτεταμένη εκδοχή. Η προσαρμοστικότητα του κάθε χρωμοσώματος ισούται με το δυνητικό κέρδος που θα είχε η εταιρία αν χρησιμοποιούσε την ποσότητα που αντιστοιχεί στο χρωμόσωμα, με τιμή ίση με την τιμή της τρέχουσας περιόδου, όπως έχει ήδη προσδιοριστεί παραπάνω. Η προσαρμοστικότητα αυτή χρησιμοποιείται για την εφαρμογή των τελεστών και την δημιουργία της επόμενης γενιάς.

Οι διαφορές, όπως διαφαίνεται και από τα αποτελέσματα των προσομοιώσεων της Arifovic, είναι περισσότερο ερμηνευτικής φύσης. Οι αλγόριθμοι ενός και πολλών πληθυσμώνδεν διαφοροποιούνται τόσο ως προς τα αποτελεσματά τους. Οι διαφορές βρίσκονται μεταξύ των εναλλακτικών εκδοχών των αλγορίθμων (βασική και εκτεταμένη). Πιο συγκεκριμένα, όπως αναφέρει η Arifovic, τόσο ο μονοπληθυσμιακός όσο και ο πολυπληθυσμιακός αλγόριθμος συγκλίνουν γύρω από τη λύση των ορθολογικών προσδοκιών (Muth 1961). Η ποσότητα αυτή προσδιορίζεται κάτω από την παραδοχή ότι η κάθε εταιρία προσδοκά ότι η τιμή της περιόδου θα πρέπει να ισούται με την τιμή που ισορροπεί η αγορά. Δηλαδή

$$P_t^e = P_t \text{ ή } x + ynq_t = A - Bnq_t \qquad (1)$$

που οδηγεί στην ποσότητα ισορροπίας για την κάθε επιχείρηση

$$q_t^* = \frac{A - x}{n(B + y)} \qquad (2)$$

Η παραπάνω ποσότητα ισορροπίας λοιπόν, δεν είναι άλλη από την ισορροπία Walras, αφού η εξίσωση (1) είναι ταυτόσημη με τη συνθήκη 1ου βαθμού στην μεγιστοποίηση

$$\max_q \Pi = \max_q \left\{ Pq - xq - \frac{1}{2} ynq^2 \right\}$$

με την τιμή P να θεωρείται σταθερή. Οι εταιρείες, όπως είπαμε, θεωρούνται price-takers στην λύση του Walras. Αντίθετα η ισορροπία κατά Nash, σε καθαρές στρατηγικές, λαμβάνεται, οπως είπαμε. με τη μεγιστοποίηση της

$$\max_q \Pi = \max_q \left\{ P(q)q - xq - \frac{1}{2} ynq^2 \right\}$$

Οι παίκτες δηλαδή λαμβάνουν υπόψη τους το γεγονός ότι η τιμή επηρεάζεται από την ποσότητα που προσφέρουν στην αγορά.

Στην «απλή έκδοση» των αλγορίθμων, η σύγκλιση δεν είναι ακριβής (υπάρχουν διακυμάνσεις), ενώ στην επαυξημένη η σύγκλιση είναι ακριβής (ο αλγόριθμος συγκλίνει σε μια σταθερή τιμή και σταματάει να μεταβάλλεται). Αυτό οφείλεται στο γεγονός ότι στην μεν «απλή» έκδοση ο τελευταίος τελεστής που εφαρμόζεται στη δημιουργία του νέου χρωμοσώματος είναι ο τελεστής μετάλλαξης, ενώ στην περίπτωση της εκλογής με τουρνουά που χαρακτηρίζει την «επαυξημένη» έκδοση, τελευταία εφαρμόζεται η επιλογή των απογόνων μέσω του τουρνουά. Όπως αναφέραμε σε προηγούμενη ενότητα, στην πρώτη περίπτωση έχουμε εργόδικη διαδικασία Markov και οι παρατηρούμενες διακυμάνσεις είναι αναμενόμενες. Στην περίπτωση της επιλογής με τουρνουά, έχουμε μια αλυσίδα ανάλογη με εκείνη της εφαρμογής του τελεστή διασταύρωσης και μόνο. Εύκολα διαπιστώνει κανείς ότι, εφόσον ο πληθυσμός (ή οι πληθυσμοί στην περίπτωση του πολυ-πληθυσμιακού μοντέλου) αποτελείται από τα ίδια χρωμοσώματα, ο μόνος τρόπος να «ξεφύγουμε» από αυτήν την κατάσταση, είναι, τα χρωμοσώματα-απόγονοι που θα προκύψουν από τους τελεστές της διασταύρωσης και της μετάλλαξης, να έχουν προσαρμοστικότητα μεγαλύτερη από αυτή του χρωμοσώματος που απαρτίζει ολόκληρο τον πληθυσμό. Στην περίπτωση όμως, που ο αλγόριθμος έχει συγκλίνει στην ισορροπία Walras, η τιμή ισορροπίας -που προσδιορίζει και την προσαρμοστικότητα της οποιασδήποτε στρατηγικής τη δεδομένη στιγμή- είναι η

$$p = A - B\frac{A - x}{B + y} \qquad (3)$$

Που αντιστοιχεί στην ποσότητα ισορροπίας κατά Walras (2). Στην τιμή (3) όμως, η ποσότητα (2) μεγιστοποιεί το κέρδος της επιχείρησης i, οπότε όλα τα χρωμοσώματα που θα προκύπτουν θα έχουν χαμηλότερη προσαρμοστικότητα από το χρωμόσωμα που αντιστοιχεί στην ποσότητα (2). Έτσι η κατάσταση κατά την οποία ολόκληρος ο πληθυσμός αποτελείται από αυτό το χρωμόσωμα, είναι μια στάσιμη κατάσταση. Έτσι εξηγείται η συμπεριφορά των «επαυξημένων» αλγορίθμων της Arifovic.

Οι προσομοιώσεις που έκανε η Arifovic έγιναν με ένα εύρος παραμέτρων (πιθανότητα μετάλλαξης κλπ.). Διαπίστωσε ότι η τιμές των παραμέτρων παίζουν πολύ σημαντικό ρόλο όσον αφορά τις ιδιότητες σύγκλισης των αλγορίθμων. Αυτό είναι σύνηθες φαινόμενο στους γενετικούς αφού μια χαμηλή πιθανότητα μετάλλαξης, π.χ., μπορεί να οδηγήσει σε γρήγορη σύγκλιση του αλγόριθμου σε ένα τοπικό ελάχιστο, ενώ μια υψηλή τιμή για την πιθανότητα μετάλλαξης μπορεί να οδηγήσει στην αστοχία του αλγόριθμου.

Τελικά, οι διαφορές μεταξύ του μονο-πληθυσμιακού και του πολυ-πληθυσμιακού αλγόριθμου δεν είναι σημαντικές, όσον αφορά τις ιδιότητες σύγκλισής τους, εφόσον ακολουθούμε την ίδια έκδοση αλγορίθμου («κανονικό» ή «επαυξημένο»). Οι διαφορές αφορούν περισσότερο την ερμηνευτική δυνατότητα των δύο μοντέλων, σημειώνει η Arifovic. Στην περίπτωση του ενός πληθυσμού, τα διαφορετικά χρωμοσώματα απεικονίζουν τις επιλογές διαφορετικών επιχειρήσεων. Έτσι ο τελεστής ανασυνδιασμού αντικατοπτρίζει την μίμηση των επιτυχημένων στρατηγικών των άλλων επιχειρήσεων. Πρόκειται δηλαδή για ένα μοντέλο κοινωνικής μάθησης (social learning). Στην περίπτωση πολλών πληθυσμών, το κάθε χρωμόσωμα απεικονίζει μια εναλλακτική ιδέα της ίδιας της επιχείρησης. Πρόκειται δηλαδή για ένα μοντέλο ατομικής μάθησης (individual learning). Άρα τα δύο μοντέλα απεικονίζουν δύο διαφορετικούς τρόπους μάθησης, τελικά.

Dawid, H. και Kopel, M. (1998): On Economic Applications of the Genetic Algorithm: A model of the Cobweb Type

Οι Dawid και Kopel αφενός διερεύνησαν το παίγνιο Cournot με γενετικούς αλγορίθμους, χρησιμοποιώντας μια διαφορετική συνάρτηση κόστους από την Arifovic, και αφετέρου επέκτειναν το μοντέλο της Arifovic, έτσι ώστε οι πράκτορες να λαμβάνουν αποφάσεις, όχι μόνο όσον αφορά την ποσότητα που θα προσφέρουν, αλλά και το αν θα μείνουν ή όχι στην αγορά.

Το μοντέλο Cournot που χρησιμοποίησαν έχει ως εξής: Υπάρχουν n επιχειρήσεις στην αγορά (οι οποίες όπως αποδεικνύεται από την δυναμική του γενετικού αλγόριθμου είναι price-takers) οι οποίες έχουν συνάρτηση κόστους

$$c(q_i) = \begin{cases} a + bq_i^2 & q > 0 \\ 0 & q_i = 0 \end{cases}$$

Η συνάρτηση κόστους είναι κοινή για όλες τις επιχειρήσεις και τα a,b είναι θετικές παραμέτροι. Ορίζοντας τη συνάρτηση κόστους κατ΄ αυτόν τον τρόπο, οι ερευνητές εισάγουν στο μοντέλο την δυνατότητα εγκατάλειψης της αγοράς. Εφόσον μια επιχείρηση κρίνει σκόπιμο, μπορεί να αναστείλει τις δραστηριότητές της, για κάποια δεδομένη χρονική περίοδο του μοντέλου, οπότε τόσο η ποσότητα όσο και το κόστος της είναι μηδενικό.

Η αντίστροφη συνάρτηση ζήτησης είναι κι εδώ

$$P = A - B\sum_{i=1}^{n} q_i$$

με A,B θετικά.

Οι ερευνητές χρησιμοποιούν δύο γενετικούς αλγορίθμους. Στην πρώτη περίπτωση χρησιμοποιούν έναν αλγόριθμο με ένα χρωμόσωμα για κάθε επιχείρηση, με το κάθε χρωμόσωμα να κωδικοποιεί την ποσότητα που επιλέγει η επιχείρηση και μόνο. Ο αλγόριθμος είναι ένας «κανονικός» γενετικός αλγόριθμος, με επιλογή γονέων ανάλογη με την προσαρμοστικότητα, τελεστή διασταύρωσης ενός τυχαίου σημείο (single point random crossover) και τελεστή μετάλλαξης με σταθερή πιθανότητα. Είναι δηλαδή αρκετά παρόμοιος με τον πρώτο «απλό» αλγόριθμο της Arifovic παραπάνω. Η μοναδική διαφορά είναι ότι οι Dawid και Kopel χρησιμοποίησαν έναν μετασχηματισμό του κέρδους σαν συνάρτηση προσαρμοστικότητας (fitness) για να αποφύγουν αρνητικές τιμές προσαρμοστικότητας. Πρώτα απ΄ όλα, σημειώνουν οι ερευνητές, για να υπάρχουν κάποιοι συνδιασμοί κόστους και ποσότητας που να οδηγούν σε θετικά κέρδη, θα πρέπει

$$a < \frac{A^2}{4b}$$

Διαφορετικά, δε θα υπήρχε κίνητρο για μια επιχείρηση για είσοδο στην αγορά. Αλλά ακόμα και αν ισχύει η παραπάνω συνθήκη, θα μπορούσε να μην υπάρχει ισορροπία Walras, λόγω της ασυνέχειας της συνάρτησης κόστους. Εφόσον

$$a < \frac{A^2 b}{\left(2b + 2Bn\right)^2}$$

τότε η ισορροπία Walras δίνεται από την

$$P^* = \frac{2Ab}{2b + 2Bn}$$

και την αντίστοιχη ποσότητα, όπως προκύπτει από την συνάρτηση ζήτησης. Διαφορετικά, όλες οι συναρτήσεις παρουσιάζουν αρνητικά κέρδη στην ποσότητα αυτή ισορροπίας, οπότε τις συμφέρει να αποχωρήσουν από την αγορά , μηδενίζοντας το κόστος τους.

Η συνάρτηση προσαρμοστικότητας που χρησιμοποίησαν είναι

$$f_k(\phi) = \Pi\left(p(\phi, y(k))\right) + a + b\frac{A}{Bn}$$

με k το συγκεκριμένο χρωμόσωμα και ϕ την κατάσταση του πληθυσμού (το σύνολο των χρωμοσωμάτων).

Οι ερευνητές προχώρησαν σε προσομοιώσεις για διαφορετικά σύνολα παραμέτρων, που κάλυπταν και τις δύο περιπτώσεις παραπάνω. Οι προσομοιώσεις έδειξαν ότι οι ποσότητες σύγκλιναν, με τις αναμενόμενες διακυμάνσεις, στην ίδια ισορροπία, τόσο για την 1η, όσο και για τη 2η περίπτωση παραπάνω. Πιο συγκεκριμένα, οι Dawid, Kopel χρησιμοποίησαν

- Αριθμό χρωμοσωμάτων $N = 1000$
- Πιθανότητα εφαρμογής τελεστή διασταύρωσης $p_{cr} = 1$
- Πιθανότητα εφαρμογής τελεστή μετάλλαξης $p_{mut} = 0{,}001$
- Αριθμός ψηφίων στο χρωμόσωμα $l = 10$
- $A = 5, a = 0{,}25, b = 1, B = \frac{5}{1000}$ στο 1ο σύνολο προσομοιώσεων και
- $A = 5, a = 1, b = 1, B = \frac{5}{1000}$ στο 2ο.

Και στις δύο περιπτώσεις, η ποσότητα -μετά από κάποιο αριθμό γενεών - κυμαίνεται γύρω από την ποσότητα $q^* = 0{,}75$ που αποτελεί τηη ισορροπία Walras, στην 1η περίπτωση. Οι επιχειρήσεις δηλαδή συνεχίζουν να μεγιστοποιούν τα κέρδη τους (θεωρώντας δεδομένη την τιμή πάντα), όπως προσδιορίζονται από το 1ο σκέλος της συνάρτησης κόστους, χωρίς να μπορούν να κάνουν το «νοητό» άλμα στην μηδενική ποσότητα, που θα τους επέτρεπε να έχουν μηδενικά, αντί αρνητικά κέρδη.

Για αυτόν το λόγο, οι ερευνητές τροποποιούν την κωδικοποίηση του χρωμοσώματος. Εισάγουν ένα ψηφίο το οποίο απεικονίζει την απόφαση της επιχείρησης για συμμετοχή ή μη στην αγορά. Αν το bit αυτό είναι 1 η επιχείρηση παράγει την ποσότητα που προσδιορίζει το υπόλοιπο χρωμόσωμα. Αν είναι 0, η επιχείρηση απέχει από την αγορά, παράγοντας μηδενική ποσότητα, με μηδενικό κέρδος. Οι προσομοιώσεις που έκαναν για τον τροποποιημένο αυτόν γενετικό αλγόριθμο, έδωσαν διαφορετικά αποτελέσματα για το κάθε σύνολο παραμέτρων. Στην πρώτη περίπτωση οι ποσότητες συνέκλιναν, σωστά, στην Βαλρασιανή ισορροπία $q^* = 0{,}75$. Στη δεύτερη περίπτωση οι ποσότητες κυμαίνονταν γύρω από την τιμή $q = 0{,}6$, με μεγαλύτερη μάλιστα διακύμανση απ΄ ότι προηγουμένως. Η εξήγηση για αυτό το αποτέλεσμα είναι απλή. «Όταν η μέση ποσότητα είναι $q = 0{,}5$, η τιμή είναι $2{,}5 > \sqrt{ab} = 2$. Συνεπώς η βέλτιστη απόφαση για τις επιχειρήσεις είναι να αυξήσουν την παραγωγή τους σε μια (μέση) ποσότητα $q = 1{,}25$ οπότε η εξελικτική δυναμική οδηγεί τη μέση ποσότητα $\bar{q} > 0{,}6$. Όταν όμως συμβεί αυτό η τιμή πέφτει κάτω από 2, οπότε η βέλτιστη απόφαση είναι η αποχή από την αγορά. Από τη στιγμή που η υλοποίηση αυτής της απόφασης απαιτεί την αντιστροφή μονάχα ενός ψηφίου, πολλές επιχειρήσεις την κάνουν άμεσα. Έτσι η μέση ποσότητα μειώνεται και πάλι, μέχρις ότου οι ποσότητες οι μεγαλύτερες από $0{,}6$ να δίνουν την υψηλότερη προσαρμοστικότητα. Αυτές οι ποσότητες διαχέονται στον πληθυσμό και έτσι η μέση ποσότητα ταλαντώνεται γύρω από το $\bar{q} = 0{,}6$» [Dawid, Kopel (1998)].

Τελικά η μελέτη αυτή, δεν αντίκειται, κάτω από το πρίσμα της σύγκλισης σε συγκεκριμένη «κατηγορία» ισορροπίας, με τα αποτελέσματα της Arifovic (1994). Και στις δύο περιπτώσεις έχουμε σύγκλιση στην ισορροπία Walras. Απλά οι Dawid και Kopel, ανακαλύπτουν ότι η κωδικοποίηση απλά των ποσοτήτων στο χρωμόσωμα, δεν αρκεί στην περίπτωση που μια ασυνέχεια στην κυρτή συνάρτηση κόστους (και κοίλη συνάρτηση κέρδους) οδηγεί στην φυγή από την αγορά, σαν βέλτιστη στρατηγική. Σε αυτήν την περίπτωση θα πρέπει να κωδικοποιείται ξεχωριστά η επιλογή αυτή της επιχείρησης. Ωστόσο, επιμένουμε, σε κάθε περίπτωση οι επιχειρήσεις συμπεριφέρονται σαν “price-takers” και όχι σαν ολιγοπωλητές Nash-Cournot.

Franke, R. (1998): Co-evolution and Stable Adjustments in the Cobweb Model

Εκτός από τη διερεύνηση των θεωρητικών ιδιοτήτων των γενετικών αλγορίθμων της Arifovic, ο Franke εισήγαγε μια στοχαστική συνάρτηση ζήτησης, στο μοντέλο της τελευταίας. Πιο συγκεκριμένα, η αντίστροφη συνάρτηση ζήτησης του Franke είναι

$$P = A - B\sum_{i=1}^{n} q_i + u_t \quad \text{με } u_t = \rho u_{t-1} + \upsilon_t$$

με τον υ_t «λευκό θόρυβο». Η u_t είναι συνεπώς μια autoregressive χρονοσειρά, που ακολουθεί το μοντέλο AR(1). Ο Franke διατήρησε την ίδια συνάρτηση κόστους με την Arifovic και συνεπώς η ισορροπία Walras δίνεται από τις

$$p^* = \frac{(A + u_t)y + Bx}{B + y}$$

και

$$q^* = \frac{A + u_t - x}{n(B + y)}$$

Οι κανόνες του γενετικού αλγορίθμου του Franke, είναι οι ίδιοι με εκείνους του επαυξημένου αλγορίθμου της Arifovic, με δύο διαφορές. Ο Franke χρησιμοποιεί μια «ασθενή» επιλογή όπως λέει, η οποία δεν είναι άλλη από την πλέον συνηθισμένη εκδοχή της επιλογής με τουρνουά, σε αντίθεση με την «ισχυρή» επιλογή της Arifovic. Αντί δηλαδή να συγκρίνει τις προσαρμοστικότητες των 2 γονέων και των 2 απογόνων και να διατηρεί στην επόμενη γενεά τα 2 χρωμοσώματα με την υψηλότερη προσαρμοστικότητα, απλά συγκρίνει την προσαρμοστικότητα του κάθε γονέα με την προσαρμοστικότητα του αντίστοιχου παιδιού και διατηρεί τον «νικητή» στην επόμενη γενιά. Φυσικά, αυτό δεν έχει καμμία ουσιαστική διαφορά, όπως διαπιστώνει και ο Franke, αλλά και όπως προκύπτει από την ανάλυση των διαδικασιών Markov που κάναμε παραπάνω. Αλλά και η πιθανότητα επιλογής ενός χρωμοσώματος σαν γονέα, διαφοροποιείται. Ο Franke χρησιμοποιεί τον εξής μετασχηματισμό της συνάρτησης προσαρμοστικότητας, για να προσδιορίσει την πιθανότητα επιλογής (Miller 1996)

$$\hat{\mu}_{it} = \max\left[0, \sigma\left(\mu_{it} - \overline{\mu}_t\right) + \delta_t\right]$$

με μ_{it} την προσαρμοστικότητα του χρωμοσώματος i τη στιγμή t, $\overline{\mu}_t$ τη μέση προσαρμοστικότητα του πληθυσμού, δ την τυπική απόκλιση των προσαρμοστικοτήτων των χρωμοσωμάτων του πληθυσμού την ίδια χρονική στιγμή, και σ μια θετική παράμετρο. Η πιθανότητα επιλογής του χρωμοσώματος i σαν γονέα είναι

$$p_i = \begin{cases} \dfrac{\hat{\mu}_{it}}{\sum_{j=1}^{n} \hat{\mu}_{jt}} & \delta_t > 0 \\ \dfrac{1}{n} & \delta_t = 0 \end{cases}$$

Κατ΄ αυτόν τον τρόπο, «μια στρατηγική αποκλείεται να επιλεγεί εφόσον η προσαρμοστικότητά της είναι χειρότερη από $\frac{1}{\sigma}$ τυπικές αποκλίσεις από τη μέση» [Franke 1998]. Ο Franke θεώρησε αναγκαίο να τροποποιήσει και τη συνάρτηση προσαρμοστικότητας, ώστε να «ανιχνεύει» καλύτερα το στοχαστικό αυτό περιβάλλον. Η προσαρμοστικότητα του πράκτορα δεν δίνεται από το κέρδος του αποκλειστικά στον τρέχοντα γύρο της προσομοίωσης, αλλά από το άθροισμα των κερδών του, στους 5 προηγούμενους γύρους.

$$\mu_{it} = \sum_{\tau = t-4}^{t} \pi\left(p_\tau, q_{i\tau}\right)$$

Στην περίπτωση που η «στρατηγική» q_i δημιουργήθηκε μετά από $t = \tau - 5$ το κέρδος για τις προηγούμενες χρονικές στιγμές δίνεται από το δυνητικό κέρδος που θα είχε, σύμφωνα με την τιμή που επικράτησε στην αγορά εκείνη τη στιγμή.

Ο Franke εισάγει τέσσερα διαφορετικά είδη στρατηγικών για τις επιχειρήσεις:

- Τις στρατηγικές «σταθερής παραγωγής» (QF). Οι στρατηγικές αυτές είναι ίδιες με εκείνες της Arifovic (1994). Μια στρατηγική QF είναι μια απόφαση για δεδομένη ποσότητα, την οποία παράγει η επιχείρηση, μέχρι ο γενετικός αλγόριθμος να αλλάξει το χρωμόσωμα που αντιστοιχεί στην επιχείρηση.

- Τις στρατηγικές «προσαρμοστικών προσδοκιών» (AE). Αυτό το μοντέλο πρόβλεψης έχει μακρά παράδοση στο cobweb model. Σύμφωνα με αυτή τη στρατηγική, η επιχείρηση χρησιμοποιεί μια προσδοκώμενη τιμή, η οποία αυξάνεται (μειώνεται) εάν η πραγματική τιμή είναι υψηλότερη (χαμηλότερη) από την προσδοκώμενη. Συμβολίζοντας την «ταχύτητα προσαρμογής» με a και την προσδοκώμενη τιμή με $p_t^e \equiv p_t^e(AE,a)$, ο μηχανισμός προσαρμογής δίνεται από την $p_t^e = (AE,a) = p_{t-1}^e(AE,a) + a\left[p_{t-1} - p_{t-1}^e(AE,a)\right]$. Η στρατηγική AE, προσδιορίζει την ποσότητα της επιχείρησης κατά τρόπο ώστε να μεγιστοποιούνται τα αναμενόμενα κέρδη, με βάση την προσδοκώμενη τιμή, $q = q^*\left(p_t^e(AE,a)\right)$. Δεδομένου ότι η ποσότητα προσδιορίζεται λύνοντας αυτό το πρόβλημα βελτιστοποίησης, το χρωμόσωμα προφανώς πρέπει να απεικονίζει κάτι διαφορετικό. Αυτό που κωδικοποιεί το χρωμόσωμα, είναι η ταχύτητα προσαρμογής a.

- Παλινδρομήσεις τάξης 1 (R1). Εδώ η επιχείρηση χρησιμοποιεί παλινδρομήσεις ελαχίστων τετραγώνων για να προσδιορίσει την προσδοκώμενη τιμή. Ο Franke υποθέτει εδώ, ότι η παλαιότερη πληροφορία αγνοείται, και οι επιχειρήσεις χρησιμοποιούν ένα «κυλιόμενο» δείγμα περιόδου Τ για τις παρατηρήσεις των τιμών που οδηγούν, μέσω της παλινδρόμησης $p_\tau = \beta_t^{R1} p_{\tau-1} + \varepsilon_\tau \quad (\tau = t-1,\dots,t-T-1)$ στην προσδοκόμενη τιμή $p_t^e = \beta_t^{R1} p_{t-1}$. Η τιμή του β_t^{R1}, δίνεται από την ελαχιστοποίηση του τετραγωνικού σφάλματος και ισούται με $\beta_t^{R1} =$ $\dfrac{\sum_{k=1}^{T-1} p_{t-k}p_{t-k-1}}{\sum_{k=1}^{T-1} p_{t-k-1}^2}$. Αυτό που κωδικοποιείται στο χρωμόσωμα είναι η περίοδος Τ.

- Παλινδρομήσεις τάξης 2 (R2). Οι στρατηγικές αυτές είναι παρόμοιες με τις R1, αλλά οι τιμές που χρησιμοποιούνται για την παλινδρόμηση είναι εκείνες των περιόδων $t-2, t-4, \ldots t-2T-2$. Εδώ το χρωμόσωμα κωδικοποιεί το $S = 2T$.

Ο Franke χρησιμοποιεί και τις τέσσερις στρατηγικές στο μοντέλο του. Αρχικά, κάθε τύπος στρατηγικής χρησιμοποιείται από ένα μέρος των πρακτόρων. Οι διαφοροποιούμενες πιθανότητες επιλογής ενός χρωμοσώματος σαν γονέα, από την άλλη, μεταβάλλουν τις «αναλογίες» των στρατηγικών στις επόμενες γενιές. Σημειώνουμε εδώ ότι ο αλγόριθμος του Franke, απαγορεύει την «διασταύρωση» μεταξύ στρατηγικών διαφορετικού τύπου. Αν ο πρώτος γονέας είναι του πρώτου τύπου λ.χ., ο δεύτερος γονέας θα πρέπει επίσης να είναι του ίδιου τύπου, διαφορετικά επαναλαμβάνεται η διαδικασία επιλογής. Τέλος, για να αποφύγει το ενδεχόμενο της πλήρους εξάλειψης κάποιου τύπου στρατηγικής από τον πληθυσμό, εισάγει μια διαδικασία «ανάνηψης» του είδους των στρατηγικών. Για κάθε τύπο στρατηγικής, υπάρχει μια «δεξαμενή» χρωμοσωμάτων, από την οποία παίρνει δύο χρωμοσώματα και τα επαναεισάγει στον πληθυσμό, όποτε εξαληφθούν τελείως τα χρωμοσώματα του συγκεκριμένου τύπου στρατηγικής (ή παραμείνει μόνο ένα). Η δεξαμενή αυτή αντιπροσωπεύει κατά κάποιο τρόπο τη «συλλογική μνήμη» των παικτών, όπως λέει ο Franke. Για λόγους απλότητας, ο Franke θεωρεί ότι τα χρωμοσώματα της δεξαμενής αυτής, παραμένουν τα ίδια καθόλη τη διάρκεια της προσομοίωσης. Εφόσον μείνει μονάχα ένα χρωμόσωμα της συγκεκριμένης στρατηγικής στον πληθυσμό, υπολογίζεται η δυνητική προσαρμοστικότητα όλων των χρωμοσωμάτων της αντίστοιχης δεξαμενής (η προσαρμοστικότητα δηλαδή που θα είχαν με τις συγκεκριμένες συνθήκες που επικρατούσαν στην αγορά στις t-τ περιόδους, όπου τ η αντίστοιχη παράμετρος στον τύπο στρατηγικής) και το χρωμόσωμα με την υψηλότερη δυνητική προσαρμοστικότητα, μαζί με ένα δεύτερο τυχαίο χρωμόσωμα, εισάγεται στον πληθυσμό, ώστε να «εμπλουτιστεί» το γενετικό υλικό του συγκεκριμένου τύπου στρατηγικής και να τον προφυλάξει από την πλήρη εξάλειψη. Αυτή η διαδικασία γίνεται εφόσον η δυνητική προσαρμοστικότητα του χρωμοσώματος αυτού είναι μεγαλύτερη από τη μέση προσαρμοστικότητα του πληθυσμού. Διαφορετικά δεν γίνεται τίποτε. Τα δύο αυτά χρωμοσώματα αντικαθιστούν δύο, τυχαία επιλεγμένα, χρωμοσώματα του πληθυσμού.

Οι τιμές των παραμέτρων που χρησιμοποιεί ο Franke στις προσομοιώσεις του, είναι, στις περιπτώσεις που υπάρχει αντιστοιχία, οι ίδιοι με αυτούς της Arifovic (1994). Συγκεκριμένα $n = 40$, $A = 2{,}296$, $B = 0{,}0168$, $x = 0$, $y = 0{,}0160$, $p_{cr} = 0{,}6$, $p_{mut} = 0{,}0033$. Ενώ για τις νέες παραμέτρους του μοντέλου του χρησιμοποιεί τις $\rho = 0{,}75$, $sdv = 0{,}075$, $\sigma = 1$. Η sdv είναι η τυπική απόκλιση των (ασυσχέτιστων) τιμών της υ_t (ο μέσος είναι 0, αφού πρόκειται για λευκό θόρυβο).

Ο Franke εστιάζει την προσοχή του σε δύο πράγματα: στον τύπο της ισορροπίας που ανακύπτει σε μακροοικονομικό επίπεδο και στις εσωτερικές διαφοροποιήσεις των στρατηγικών των παικτών (τύπος στρατηγικής που επιλέγεται και παράμετροι). Αν και οι 4 τύποι στρατηγικών παραμένουν ενεργοί καθόλη τη διάρκεια της προσομοίωσης, οι πλέον επιτυχημένες είναι οι AE και σε μικρότερο βαθμό οι QF. Σε μακροοικονομικό επίπεδο διαπιστώνεται και εδώ η (με την ευρεία έννοια) σύγκλιση στη Βαλρασιανή ισορροπία. Όπως λέει ο Franke, ο κατάλληλος όρος για να περιγραφεί η μακροοικονομική δυναμική συμπεριφορά του συστήματος είναι «κοντά στην ισορροπία κατά Walras». Ένα άλλο ενδιαφέρον γεγονός που σημειώνει είναι ότι, όταν ένας τύπος στρατηγικής τείνει να εκλείψει, η δυνητική προσαρμοστικότητα των χρωμοσωμάτων στην δεξαμενή «συλλογικής μνήμης» τείνει να αυξάνεται σε σχέση με την μέση προσαρμοστικότητα του πληθυσμού και συνεπώς η «εμφύτευση» των χρωμοσωμάτων από τη δεξαμενή είναι ιδιαίτερα πιθανή. Γενικά πάντως παρατηρείται μεγάλη ετερογένεια τόσο στους τύπους στρατηγικών που βρίσκονται στον πληθυσμό κάθε φορά (για τους 250 πρώτους γύρους της προσομοίωσης η AE κυριαρχεί με αναμενόμενη τιμή πάνω από 35/40 χρωμοσώματα, με την QF να αυξάνεται σε συχνότητα πάνω από 20/40 μετά τον 250^{ο} γύρο, ενώ και οι R1,R2 επιλέγονται από κάποιες μεμονωμένες επιχειρήσεις), όσο και όσον αφορά τις παραμέτρους των στρατηγικών. Το χρωμόσωμα της ταχύτητας προσαρμογής των AE για παράδειγμα κυμαίνεται μεταξύ του 0 και του 0,5 για τους πρώτους 250 γύρους της προσομοίωσης, για να σταθεροποιηθεί σχεδόν γύρω από το 0,4 στους τελευταίους 50 γύρους.

Διαπιστώνουμε πάντως ότι και εδώ, παρόλη την ποιοτική ευρύτητα των εναλλακτικών στρατηγικών, το σύστημα συγκλίνει (με την έννοια του Ljapunov πάντα) στην ισορροπία κατά Walras και όχι στην ισορροπία κατά Nash.

Vega-Redondo, F. (1997): The Evolution of Walrasian behaviour

Παρόλο που δε θα μας απασχολήσουν εδώ οι γενικότεροι εξελικτικοί αλγόριθμοι, η συγκεκριμένη δημοσίευση σχετίζεται με το θέμα μας με ένα ιδιαίτερο τρόπο και θα αναφέρουμε ορισμένα πράγματα. Ο Vega-Redondo, προτείνει έναν εξελικτικό αλγόριθμο και εξετάζει τις ιδιότητες σύγκλισής του, σε θεωρητικό επίπεδο. Δείχνει ότι κάτω από αρκετά γενικές συνθήκες, η συμπεριφορά των επιχειρήσεων, που βασίζεται στον εν λόγω εξελικτικό αλγόριθμο, συγκλίνει στην ισορροπία Walras.

Το υπόδειγμα του αποτελείται από n ολιγοπωλητές Cournot, με κοινή συνάρτηση κόστους $c_i(q_i)$ και αντίστροφη συνάρτηση ζήτησης $P\left(\sum_{i=1}^{n} q_i\right)$, φθίνουσα φυσικά ως προς την ποσότητα. Οι συνθήκες δηλαδή για τις συναρτήσεις κόστους και ζήτησης είναι εντελώς γενικές. Οι πράκτορες επιλέγουν την ποσότητα που θα παράγουν από ένα διακριτό σύνολο ποσοτήτων

$$\Gamma = \{0, \delta, 2\delta, ..., \upsilon\delta\}$$

με $\delta > 0$ και $\upsilon \in N$. Απαραίτητη προϋπόθεση, η ποσότητα που αντιστοιχεί στην ισορροπία Walras, να ανήκει στο Γ. Κάτω από αυτές τις γενικές συνθήκες, η ισορροπία Walras, αποδίδεται απλά ως $P(n\hat{q})\hat{q} - c(\hat{q}) \geq P(n\hat{q})q - c(q)$, η ποσότητα δηλαδή που μεγιστοποιεί τη συνάρτηση κέρδους, με δεδομένη την τιμή.

Η εξελικτική διαδικασία γίνεται σε διακριτό χρόνο, όπως πάντα. Για κάθε επιχείρηση υπάρχει πιθανότητα p να αλλάξει την τρέχουσα ποσότητά της. Εφόσον συμβεί αυτό, υιοθετεί μία από τις πιο επιτυχημένες επιλογές των υπολοίπων επιχειρήσεων, την προηγούμενη χρονική στιγμή. Πιο συγκεκριμένα, το σύνολο των πιθανών επιλογών είναι το

$$B(t-1) = \{q \in \Gamma : \exists j \in N \text{ ώστε } q = q_j(t-1) \text{ και } \forall k \in N, \pi_j(t-1) \geq \pi_k(t-1)\}$$

Με άλλα λόγια, επιλέγει την ποσότητα που έδωσε το μεγαλύτερο κέρδος στην επιχείρηση που την επέλεξε, στον προηγούμενο γύρο της προσομοίωσης.

Ο Vega-Redondo, επισημαίνει ότι ο εν λόγω αλγόριθμος, οδηγεί σε μια absorbing Markov Chain, με καταστάσεις απορρόφησης (absorbing states)

$$A = \{q, ..., q : q \in \Gamma\}$$

Εφόσον όλες οι εταιρίες επιλέξουν την ίδια ποσότητα, δεν υπάρχει τρόπος να εισαχθεί μια νέα ποσότητα στο σύνολο των δυνατών επιλογών. Σε ενδεχόμενη αλλαγή της ποσότητας της, μια εταιρία επιλέγει κάποια άλλη από το σύνολο των ποσοτήτων που επέλεξαν στον προηγούμενο γύρο οι υπόλοιπες. Έτσι οι πιθανότητες μετάβασης από την $(q, ..., q)$ σε διαφορετική κατάσταση είναι μηδενικές. Από την άλλη, μια μεταβατική μπορεί πάντα να εξελιχθεί σε κάποια από τις παραπάνω καταστάσεις απορρόφησης (absorbing states), αρκεί η ποσότητα q να ισούται με μία από τις ποσότητες που τουλάχιστον μια από τις εταιρίες επιλέγει. Έχουμε δηλαδή έναν διαμερισμό του χώρου των πιθανών καταστάσεων, με ορισμένες καταστάσεις να οδηγούν σε ορισμένες μόνο καταστάσεις απορρόφησης (absorbing states).

Αν όμως εισάγουμε στο μοντέλο το ενδεχόμενο μετάλλαξης, η αλυσίδα Markov γίνεται πλέον εργοδική, αποδεικνύει ο Vega-Redondo. Έτσι έχουμε ένα διάνυσμα σταθερών συχνοτήτων για κάθε μία από τις ενδεχόμενες καταστάσεις στις οποίες μπορεί να βρεθεί η διαδικασία, μακροπρόθεσμα, ανεξάρτητο από την αρχική κατάσταση και γενικότερα την κατάσταση της αλυσίδας Markov σε δεδομένη χρονική στιγμή. Η κατανομή αυτή των πιθανοτήτων των διαφόρων καταστάσεων που αντιστοιχεί στο σταθερό διάνυσμα συχνοτήτων w, εξαρτάται από την πιθανότητα μετάλλαξης. O Vega-Redondo αποδεικνύει ότι καθώς η πιθανότητα μετάλλαξης τείνει στο μηδέν, η κατανομή των διαφόρων πιθανών καταστάσεων τείνει σε μια κατανομή στην οποία η κατάσταση $(\hat{q},..,\hat{q})$ πραγματοποιείται με πιθανότητα 1. Με άλλα λόγια η κατανομή των καταστάσεων είναι ασυμπτωτικά κατά πιθανότητα ίση με την Βαλρασιανή ισορροπία ή η Βαλρασιανή ισορροπία είναι η μοναδική «στοχαστικά σταθερή κατάσταση» (stochastic stable state) [Foster και Young (1990)] που υπάρχει για όλες τις πιθανότητες μετάβασης, οσοδήποτε μικρές κι αν είναι αυτές. Έτσι η ισορροπία κατά Walras αποκτά έναν ξεχωριστό ρόλο, για τον συγκεκριμένο εξελικτικό αλγόριθμο, αφού η επίτευξή της είναι εφικτή για κάθε πιθανότητα μετάλλαξης, ενώ είναι η μοναδική κατάσταση με αυτήν την ιδιότητα καθώς η πιθανότητα μετάλλαξης τείνει στο μηδέν.

Vriend, N. (2000): An Illustration of the Essential Difference between Individual and Social Learning, and it's Consequences for Computational Analyses

Μία από τις σημαντκότερες μελέτες στο πεδίο είναι αυτή του Vriend (2000). O Vriend επικεντρώνει την προσοχή του στις διαφορές που επιφέρουν στις συμπεριφορές των πρακτόρων, οι γενετικοί αλγόριθμοι ατομικής (ένας πληθυσμός για κάθε παίκτη) και συλλογικής ή κοινωνικής μάθησης (ένας πληθυσμός για όλους τους παίκτες). Τις διαφορές αυτές τις αποδίδει στο λεγόμενο "spite effect" [Hamilton (1970), βλ. και Rhode and Stegeman (1995)], το οποίο χαρακτηρίζει μια συμπεριφορά η οποία, ενώ είναι επιβλαβής για τον παίκτη που την χρησιμοποιεί, είναι ακόμα περισσότερο επιβλαβής για τους αντιπάλους του, με αποτέλεσμα να τους «υποχρεώνει» κατά κάποιο τρόπο να αλλάξουν τη συμπεριφορά τους, κατά τρόπο τέτοιο ώστε το κέρδος του παίκτη να είναι τελικά υψηλότερο.

Για παράδειγμα ας πάρουμε το παρακάτω παίγνιο:

παίκτης 1/ παίκτης 2	A	B
A	(10,10)	(3,7)
B	(7,3)	(0,0)

Η ισορροπία κατά Nash βρίσκεται προφανώς στο (A,A). Το σημείο (B,A) είναι το σημείο "spite effect" για τον παίκτη 1. Εδώ, παρόλο που ο παίκτης 1 αποκλίνει από την ισορροπία κατά Nash, μειώνοντας το κέρδος του, μειώνει το κέρδος του παίκτη 2 ακόμα περισσότερο. Το σημείο (A,B) είναι το σημείο "spite effect" για τον παίκτη 2, για τους ίδιους λόγους. Ας υποθέσουμε τώρα ότι οι παίκτες παίζουν τη στρατηγική (B,B) και ότι ο παίκτης 1 αλλάζει -μέσω μετάλλαξης- στην στρατηγική A. Βλέπουμε ότι ο παίκτης B, ακόμα και χωρίς να αλλάξει τη στρατηγική του, κερδίζει περισσότερο από την προηγούμενη κατάσταση (3 αντί για 0). Δεδομένου ότι, σε έναν εξελικτικό αλγόριθμο μάθησης, οι παίκτες δεν προσδιορίζουν τις στρατηγικές τους «σκεπτόμενοι» πως θα μεγιστοποιήσουν το κέρδος τους, αλλά οι στρατηγικές αλλάζουν ως προϊόν τυχαίων καταστάσεων, τα σημεία “spite” έχουν σημασία στη μελέτη της συμπεριφοράς των παικτών. Σε έναν γενετικό αλγόριθμο, η πιθανότητα «επιβίωσης» και «διάχυσης» στον πληθυσμό, εξαρτάται από την προσαρμοστικότητα του χρωμοσώματος. Στο συγκεκριμένο μοντέλο μάθησης μέσω γενετικών αλγορίθμων στα παίγνια Cournot, η προσαρμοστικότητα ενός χρωμοσώματος εξαρτάται από την κατάσταση του πληθυσμού. Στην περίπτωση δηλαδή που έχουμε ένα χρωμόσωμα για κάθε επιλογή (καθαρή στρατηγική) της επιχείρησης, το κέρδος της στρατηγικής εξαρτάται από το σύνολο των στρατηγικών των αντιπάλων επιχειρήσεων. Έτσι θα μπορούσε μια στρατηγική, ενώ δεν είναι η βέλτιστη για τη δεδομένη κατάσταση, να οδηγήσει, μέσω της τυχαίας λ.χ. υιοθέτησής της, σε μια άλλη κατάσταση κατά την οποία το κέρδος της να είναι υψηλότερο από των στρατηγικών των άλλων παικτών. Σε αυτή την περίπτωση η στρατηγική αυτή θα «πριμοδοτηθεί» από τον γενετικό αλγόριθμο και, ενδεχομένως θα κυριαρχήσει στον πληθυσμό. Από την άλλη πλευρά μια στρατηγική που -εφόσον υιοθετηθεί από όλους τους παίκτες- μπορεί να οδηγήσει σε υψηλότερα κέρδη, μπορεί -με βάση τη δεδομένη κατάσταση- να οδηγήσει σε χαμηλότερα κέρδη τον παίκτη που την υιοθετεί και να απορριφθεί από τον πληθυσμό. Παρακάτω στην παρουσίαση της μελέτης των Alos-Ferrer και Ania (2005) δίνεται μια ακόμα πιο εμπεριστατωμένη εξήγηση και μάλιστα γίνεται και μια σύνδεση με την μελέτη του Vega-Redondo (1997) που αναλύσαμε παραπάνω και που θίγει επίσης το θέμα.

Ο Vriend χρησιμοποιεί ένα συνηθισμένο μοντέλο $n-$ οπωλίου Cournot, με n επιχειρήσεις, πολυωνυμική συνάρτηση ζήτησης

$$P = a + bQ^C$$

με $Q = \sum_{i=1}^{n} q_i$, και γραμμική συνάρτηση κόστους

$$C_i = K + kq_i$$

Με δεδομένο ότι οι παράμετροι a,b,c,K,k παίρνουν τις κατάλληλες τιμές (βλ. ενότητα 2.2.2) ώστε να υπάρχει ισορροπία κατά Nash σε καθαρές στρατηγικές, τότε οι ισορροπίες κατά Walrash και Nash θα δίνονται από τις

$$Q_W = \left(\frac{k - a}{b} \right)^{\frac{1}{c}} \text{ και}$$

$$Q_N = \left[\frac{k - a}{b\left(\frac{c}{n} + 1\right)} \right]^{\frac{1}{c}}$$

αντίστοιχα [βλ. Vriend(2000)-Appendix A].

Ο Vriend χρησιμοποιεί δύο γενετικούς αλγορίθμους, έναν ατομικής μάθησης με έναν πληθυσμό για κάθε παίκτη και έναν συλλογικής μάθησης, με κοινό πληθυσμό για όλους τους παίκτες. Στον πρώτο, υπάρχει ένας πληθυσμός 40 χρωμοσωμάτων (ο Vriend θέτει $n = 40$ σε ολόκληρη την μελέτη του), με το χρωμόσωμα i να αντιστοιχεί στην ποσότητα που επιλέγει η επιχείρηση i. Τα χρωμοσώματα παραμένουν ίδια για 100 «μέρες συναλλαγών» (trading days), οπότε ανανεώνονται με τη χρήση των γενετικών τελεστών της διασταύρωσης και της μετάλλαξης. Πρόκειται για μια σημαντική -ίσως τη σημαντικότερη- διαφορά μεταξύ των αλγορίθμων του Vriend και εκείνων της Arifovic. Όπως θα δούμε, η μη ανανέωση του πληθυσμού σε κάθε βήμα και η μη χρήση δυνητικών κερδών για τον υπολογισμό των ανενεργών χρωμομσωμάτων, οδηγεί σε τελείως διαφορετικά αποτελέσματα. Το χρωμόσωμα που επιλέγει η επιχείρηση κάθε φορά προσδιορίζεται τυχαία, με πιθανότητα ανάλογη με το πόσο επιτυχημένο ήταν όταν επιλέχτηκε τελευταία φορά. Η προσαρμοστικότητα του χρωμοσώματος δηλαδή υπολογίζεται κάθε φορά που επιλέγεται, ενώ τα χρωμοσώματα που δεν έχουν ακόμα επιλεγεί δεχόμαστε ότι έχουν πιθανότητα επιλογής ίση με την μέση πιθανότητα του πληθυσμού. Το σύστημα αυτό (λέγεται και classifier system) εισήχθη από τον Holland(1986). Για να μπορέσει να λειτουργήσει σωστά αυτό το σύστημα, θα πρέπει να υπάρχει κάποια χρονική περίοδος κατά την οποία οι πληθυσμοί θα παραμένουν σταθεροί, ουτωσώστε οι προσαρμοστικότητες που υπολογίζονται να συγκλίνουν όσο το δυνατόν με τις αναμενόμενες προσαρμοστικότητες των χρωμοσωμάτων, με δεδομένη την κατάσταση του αλγορίθμου. Εξού και εισάγεται αυτός ο αριθμός των “training days” (εδώ 100), κατά τις οποίες τα χρωμοσώματα παραμένουν σταθερά.

Κατά τα άλλα οι γενετικοί αλγόριθμοι συμπεριφέρονται σαν «κανονικοί» γενετικοί αλγόριθμοι με τελεστή διασταύρωσης μονού και τυχαίου σημείου (random point crossover) και σταθερής πιθανότητας μετάλλαξης. Ο Vriend εκτέλεσε 25 προσομοιώσεις για καθένα αλγόριθμο. Οι παράμετροι είχαν ως εξής:

$$a = -1 \times 10^{-97} \quad b = 1.5 \times 10^{95} \quad c = -39{,}999997 \qquad K = -4.{,}097 \times 10^{-94} \quad \text{και } k = 0$$

Όπως ήδη είπαμε οι “training days” (ή GA-rate) ήταν 100 και ο αριθμός των επιχειρήσεων 40.

Οι προσομοιώσεις έδειξαν ότι ενώ ο αλγόριθμος «της ατομικής μάθησης» συνέκλινε σε $\overline{Q} = 805{,}1$ με μέση μεταβλητότητα $\overline{\sigma} = 80{,}5$ στις γενεές 5000-10000 , ο αλγόριθμος «κοινωνικής μάθησης» συνέκλινε γρήγορα (σε λιγότερες από 100 γενεές) σε $\overline{Q} = 1991{,}3$ με μέση μεταβλητότητα μόλις $\overline{\sigma} = 24{,}7$, στις γενεές 5000-10000. Οι ισορροπίες κατά Walras και Nash είναι, για τις δεδομένες παραμέτρους, $Q_W = 2006{,}1$ και $Q_N = 998{,}2$. Βλέπουμε δηλαδή ότι, ενώ ο αλγόριθμος ατομικής μάθησης κινείται κοντά στην ισορροπία κατά Nash, ο αλγόριθμος κοινωνικής μάθησης συγκλίνει αρκετά κοντά στην Βαλρασιανή ισορροπία. Ο Vriend αποδίδει το φαινόμενο αυτό στο “spite effect” που αναφέραμε παραπάνω.

Για την καλύτερη κατανόηση του φαινομένου, ο Vriend αναλύει την απλή περίπτωση του δυοπωλίου. Αν έχουμε ένα δυοπώλιο με αντίστροφη συνάρτηση ζήτησης

$$P = a + bQ$$

με μηδενικές συναρτήσεις κόστους $c = 0$, η ισορροπία Walras είναι $Q_W = \frac{-a}{b}$ (η ποσότητα που μηδενίζει το κέρδος, και αφού το κόστος είναι 0, την τιμή). Έστω ότι και η δύο επιχειρήσεις παράγουν το μερίδιο τους $q_i = \frac{-a}{2b}$. Σε αυτήν την περίπτωση τα κέρδη θα είναι μηδενικά, όπως και η τιμή. Τι θα γίνει αν η 1η επιχείρηση παράγει πάνω από $\frac{-a}{2b}$; Η τιμή θα γίνει αρνητική, οπότε και οι δύο επιχειρήσεις θα έχουν ζημιές. Ωστόσο η 2η επιχείρηση έχει μικρότερες ζημιές από την πρώτη, αφού παράγει μόνο $\frac{-a}{2b}$ και δεδομένου ότι οι δύο ποσότητες πωλούνται στην ίδια, αρνητική τιμή, θα έχει και μικρότερες ζημιές. Στην αντίθετη περίπτωση, στην οποία η πρώτη επιχείρηση αποφασίσει να παράγει λιγότερο από $\frac{-a}{2b}$, η τιμή θα γίνει θετική, οπότε οι δύο επιχειρήσεις θα έχουν τώρα κέρδη. Πάλι η δεύτερη επιχείρηση θα έχει μεγαλύτερα κέρδη, αφού θα πουλά μεγαλύτερη ποσότητα στην ίδια -θετική αυτή τη φορά- τιμή. Έτσι η επιχείρηση που παράγει $\frac{-a}{2b}$ έχει πάντα μεγαλύτερα κέρδη. Κατά μια έννοια η 2η επιχείρηση «εκμεταλλεύεται τις αλλαγές της 1ης» (Vriend 2000).

Στην περίπτωση των n επιχειρήσεων, «τα πράγματα γίνονται πιο πολύπλοκα. Ωστόσο ισχύει ότι, όποτε η συνολική παραγωγή είναι χαμηλότερη από το επίπεδο της ισορροπίας Walras, δηλ. Οι επιχειρήσεις παράγουν κάτω από $\frac{-a}{2b}$ η τιμή θα είναι θετική, και οι επιχειρήσεις με μεγαλύτερη q_i θα έχουν υψηλότερα κέρδη. Σημειώστε ότι αυτό μπορεί, σε συγκεκριμένες περιπτώσεις, να σημαίνει ότι οι επιχειρήσεις που παράγουν πάνω από $\frac{-a}{2b}$ θα έχουν υψηλότερα κέρδη από αυτές που παράγουν $\frac{-a}{2b}$. Το ακριβώς αντίθετο συμβαίνει όταν η μέση ποσότητα είναι υψηλότερη από $\frac{-a}{2b}$. Όσο χαμηλότερη είναι η παραγωγή μιας επιχείρησης, τόσο χαμηλότερες θα είναι οι ζημιές της» (Vriend2000).

Στην περίπτωση του γενετικού αλγορίθμου «κοινωνικής μάθησης», κάθε επιχείρηση χαρακτηρίζεται από το χρωμόσωμα-ποσότητα που της αντιστοιχεί. Όσο υψηλότερη είναι η προσαρμοστικότητα της στρατηγικής της αυτής, τόσο μεγαλύτερη είναι η πιθανότητα να επιλεγεί για διασταύρωση. Λόγω του “spite effect”, όποτε η συνολική παραγωγή είναι κάτω από την Βαλρασιανή, τα υψηλότερα κέρδη τα έχουν οι επιχειρήσεις που παράγουν μεγαλύτερες ποσότητες. Συνεπώς η συχνότητά των μεγαλύτερων ποσοτήτων στον πληθυσμό τείνει να αυξάνεται και η συνολική ποσότητα τείνει να αυξάνεται. Το αντίθετο συμβαίνει όταν η συνολική παραγωγή είναι πάνω από την ισορροπία Walras. Οι χαμηλότερες ποσότητες έχουν υψηλότερη προσαρμοστικότητα και η συνολική ποσότητα τείνει να μειώνεται. Έτσι η ποσότητα ισορροπίας τείνει στην ισορροπία Walras.

Στον αλγόριθμο «ατομικής μάθησης» από την άλλη, οι ποσότητες που αλληλεπιδρούν στην διαδικασία μάθησης (τις “training days”) δεν αλληλεπιδρούν στις ίδιες συνθήκες αγοράς, δεδομένου ότι κάθε επιχείρηση επιλέγει ένα χρωμόσωμα κάθε φορά. Έτσι, «το “spite effect”, παρότι είναι παρών στο σύστημα, δεν επηρρεάζει τη διαδικασία μάθησης, αφού το κέρδος που προκύπτει από την τρέχουσα επιλογή, δεν επηρρεάζεται από το κέρδος των παρελθούσων καταστάσεων στην αγορά. Φανερά υπάρχει ένα “spite effect”στα κέρδη που γίνεται αντιληπτό από τα χρωμοσώματα των υπολοίπων επιχειρήσεων, αλλά αυτά δεν συναγωνίζονται τα χρωμοσώματα της εν λόγω επιχείρησης, μέσα στα πλαίσια της τρέχουσας διαδικασίας μάθησης. Σημειώνουμε ότι είναι αυτή η διαδικασία μάθησης που είναι το κρίσιμο στοιχείο εδώ και όχι οι στόχοι του κάθε πράκτορα. Τόσο οι πράκτορες ατομικής μάθησης, όσο οι πράκτορες κοινωνικής μάθησης προσπαθούν να μεγιστοποιήσουν τα απόλυτα κέρδη τους. Η μόνη διαφορά είναι ότι η διαδικασία τους της μάθησης βασίζεται σε ένα διαφορετικό σύνολο παρατηρήσεων» (Vriend 2000).

Ο Vriend λοιπόν αποδίδει τη σύγκλιση στη Βαλρασιανή ισορροπία στην περίπτωση του γενετικού αλγόριθμου «κοινωνικής μάθησης» στο “spite effect”, το οποίο δεν εκφράζεται στην περίπτωση ξεχωριστών πληθυσμών για κάθε πράκτορα, λόγω της ιδιομορφίας της διαδικασίας μάθησης στη δεύτερη αυτή περίπτωση.

Barr J., Saraceno F. (2005): Cournot Competition, Organization and Learning

Παρόλο που το θέμα των Barr και Saraceno δεν άπτεται άμεσα του αντικειμένου της εφαρμογής γενετικών αλγορίθμων για την μοντελοποίηση της διαδικασίας λήψης αποφάσεων στο παίγνιο Cournot, δεδομένου ότι χρησιμοποιεί νευρωνικά δίκτυα, θεωρούμε σκόπιμο να αναφερθούμε σύντομα στη μελέτη τους αυτή, δεδομένου ότι το νευρωνικό δίκτυο που χρησιμοποίησαν, συνέκλινε στην ισορροπία κατά Nash.

Οι Barr, Saraceno μοντελοποίησαν τις αποφάσεις που αφορούν την παραγόμενη ποσότητα, των δυοπωλητών Cournot, μέσα στα πλαίσια ενός επαναλαμβανόμενου παιγνίου Cournot. Εστίασαν σε τρία αλληλοσυσχετιζόμενα ζητήματα: 1) τον ρόλο της μάθησης στην προσαρμοστική διαδικασία προς την ισορροπία, 2) τον ρόλο της οργανωτικής δομής στη λήψη αποφάσεων και 3) τον ρόλο των μεταβολών του περιβάλλοντος στη μάθηση και στις αποφάσεις που λαμβάνονται κάθε φορά. Η επιχείρηση χαρακτηρίζεται στη μελέτη σαν από ένα τεχνητό νευρωνικό δίκτυο, το οποίο πρέπει να εκτιμήσει τη βέλτιστη ποσότητα που πρέπει η επιχείρηση να παράγει, λαμβάνοντας εισόδους από το οικονομικό περιβάλλον, το οποίο επηρρεάζει την συνάρτηση ζήτησης. Μέσω προσομοιώσεων έδειξαν 1) πως οι οργανισμοί μαθαίνουν να εκτιμούν τη βέλτιστη παραγωγή στο χρόνο σαν συνάρτηση της περιβαλλοντικής δυναμικής, 2) ποιό είδος δικτύου είναι βέλτιστο για κάθε δεδομένο επίπεδο πολυπλοκότητας του περιβάλλοντος και 3) τη δομή της κατάστασης ισορροπίας.

Η μελέτη των Barr και Saraceno αποσκοπούσε στην διερεύνηση των αποτελεσμάτων τόσο των περιβαλλοντικών, όσο και των οργανωτικών παραγόντων στο αποτέλεσμα των επαναλαμβανόμενων παιγνίων Cournot. Μοντελοποίησαν την επιχείρηση σαν ένα δίκτυο επεξεργασίας πληροφοριών, ικανό να μαθαίνει ένα σύνολο περιβαλλοντικών μεταβλητών. Τα συνηθισμένα μοντέλα για την επιχείρηση, γενικά τείνουν να επικεντρώνουν στην ποσοτική στρατηγική, αγνοόντας όμως το γεγονός ότι οι αποφάσεις λαμβάνονται μέσα στα πλαίσια ενός συγκεκριμένου οργανογράμματος. Αυτό ισχύει ιδιαίτερα στις ολιγοπωλιακές επιχειρήσεις, που τείνουν να είναι μεγάλες επιχειρήσεις, με χιλιάδες εργαζομένους και έναν μεγάλο αριθμό managers που προσδιορίζουν την στρατηγική της επιχείρησης σε κάθε περίοδο. Αυτά τα γεγονότα δεν λαμβάνονται συνήθως υπόψη στη μοντελοποίηση ολιγοπωλίων, ακόμα και στις περιπτώσεις μοντέλων μάθησης.

Αντίθετα με τις περισσότερες μελέτες του ολιγοπωλίου Cournot, οι ερευνητές δεν ενδιαφέρονται για την μοντελοποίηση της διαδικασίας μάθησης της βέλτιστης στρατηγικής per se, αλλά περισσότερο ενδιαφέρονται για τη μοντελοποίηση της μάθησης του οικονομικού περιβάλλοντος (της συνάρτησης ζήτησης, βασικά). Πιο συγκεκριμένα μοντελοποιούν ενότητες διαφορετικών οργανωτικών δομών που ανταγωνίζονται προσπαθόντας να εκτιμήσουν όσο το δυνατόν καλύτερα τις μεταβαλλόμενες συνθήκες ζήτησης. Χρησιμοποιώντας αυτήν την προσέγγιση, ήταν δυνατόν να διερευνήσουν την σχέση μεταξύ της βέλτιστης δομής της επιχείρησης και της πολυπλοκότητας του περιβάλλοντος μέσα στο οποίο καλείται η επιχείρηση να πάρει τις αποφάσεις της. Βασιζόμενοι στο Barr, Saraceno (2002), μοντελοποιούν την επιχείρηση σαν ένα τεχνητό νευρωνικό δίκτυο (ANN), το οποίο πρέπει να μάθει τη βέλτιστη ποσότητα παραγωγής, λαμβάνοντας εισόδους από το οικονομικό περιβάλλον και πιο συγκεκριμένα, από μια, διαρκώς μεταβαλλόμενη συνάρτηση ζήτησης.

Η δομή της επιχείρησης αντιπροσωπεύεται από το μέγεθος του δικτύου, από τον αριθμό δηλαδή των νευρώνων του. Στο Barr and Saraceno (2002), οι ερευνητές έδειξαν ότι οι επιχειρήσεις αντιμετωπίζουν ένα δίλημμα αναφορικά με την ταχύτητα και την ακρίβεια προσαρμογής τους. Οι μικρότερες επιχειρήσεις είναι πιο ευέλικτες και μαθαίνουν γρηγορότερα, αλλά οι μεγαλύτερες επιχειρήσεις είναι πιο ακριβείς στις αποφάσεις τους μακροπρόθεσμα. Ο σκοπός της μελέτης είναι λοιπόν να διαπιστώσει, το κατά πόσον αυτό το trade-off επηρεάζεται από περιβαλλοντικούς παράγοντες, στην περίπτωση επιχειρήσεων που ανταγωνίζονται σύμφωνα με τους κανόνες του παιγνίου Cournot και πρέπει να μάθουν μια μεταβαλλόμενη συνάρτηση ζήτησης και τον τρόπο με τον οποίο η πολυπλοκότητα του περιβάλλοντος επιδρά στη βέλτιστη δομή (μέγεθος) των επιχειρήσεων.

Το πρώτο, πολύ σημαντικό πόρισμα της μελέτης είναι ότι τα νευρωνικά δίκτυα συγκλίνουν στην ισορροπία κατά Nash του παιγνίου Cournot. Με το πέρασμα του χρόνου, οι επιχειρήσεις μαθαίνουν να επιτυγχάνουν την αντιστοίχιση μεταξύ των χαρακτηριστικών του περιβάλλοντος και των βέλτιστων αποφάσεων παραγωγής. Το δεύτερο πόρισμα είναι όσο μεγαλύτερη είναι η πολυπλοκότητα του περιβάλλοντος, τόσο μικρότερη είναι η κερδοφορία των επιχειρήσεων. Το ίδιο βέβαια συμβαίνει και στη σχέση της κερδοφορίας με το σφάλμα της εκτίμησης του νευρωνικού δικτύου - επιχείρησης, όσον αφορά τις περιβαλλοντικές παραμέτρους.

Βλέπουμε λοιπόν ότι οι ερευνητές έχουν μια τελείως διαφορετική προσέγγιση στο πρόβλημα, από εκείνη όλων των υπολοίπων μελετητών του παιγνίου Cournot. Αντί να μοντελοποιούν την απόφαση της επιχείρησης σαν συνάρτηση των επιλογών των συμπαικτών της, μοντελοποιούν την προσπάθεια της να κατανοήσει το περιβάλλον της. Αντί δηλαδή η εκτιμώμενη ποσότητα να είναι η ποσότητα που καλείται να παράγει η επιχείρηση - πράκτορας, το νευρωνικό δίκτυο καλείται να εκτιμήσει τη συνάρτηση ζήτησης και μέσω αυτής, τη συνάρτηση κέρδους της επιχείρησης. Μέσω αυτής προσδιορίζεται, από το νευρωνικό δίκτυο και αυτή, η βέλτιστη ποσότητα που πρέπει να παραχθεί. Η σύγκλιση στην ισορροπία κατά Nash, καταδεικνύει την αποτελεσματικότητα τόσο της συγκεκριμένης προσέγγισης, όσο και των νευρωνικών δικτύων ως εργαλείου μοντελοποίησης διαδικασιών μάθησης και ταυτόχρονης λήψης αποφάσεων.

Alos-Ferrer, C. και Ania, A. (2005): The Evolutionary Stability of Perfectly Competitive Behavior

Οι Alos-Ferrer, Ania διερευνούν την εξελικτική συμπεριφορά τόσο του παιγνίου Cournot, όσο και γενικότερα των συμμετρικών παιγνίων στα οποία το κέρδος ενός παίκτη είναι συνάρτηση της συνολικής στρατηγικής των αντιπάλων του μόνο (δεν εξαρτάται από το ποιός παίκτης κάνει τη συγκεκριμένη επιλογή, αλλά μονάχα από κάποια παράμετρο που εκφράζει συνολικά τις επιλογές των παικτών). Απέδειξαν ότι η ισορροπία κατά Walras αποτελεί μια ολικά σταθερή ESS (κατά Shaffer), τόσο στο παίγνιο Cournot, όσο και γενικότερα σε όλα τα «συμμετρικά και συνολικά» παίγνια.

Πιο συγκεκριμένα οι Alos-Ferrer και Ania δίνουν καταρχήν τον εξής ορισμό για τα «συμμετρικά και συνολικά» παίγνια: «Ένα γενικό συμμετρικό και συνολικό παίγνιο είναι ένα παίγνιο $\Gamma = (N, S, \pi)$ με N τον αριθμό των παικτών, S τον χόρο των (κοινών για κάθε παίκτη) καθαρών στρατηγικών, ο οποίος είναι υποσύνολο ενός πλήρως διατεταγμένου χώρου X, $\pi : S \times X \to R$ μια πραγματική συνάρτηση και $g : S^N \to X$ μια συμμετρική και γνησίως αύξουσα συνάρτηση, τέτοια ώστε οι συναρτήσεις κέρδους των παικτών να δίνονται από την $\pi_i(\mathbf{s}) \equiv \pi(s_i, g(\mathbf{s}))$ για κάθε $\mathbf{s} \in S^N$ και $\forall i \in N$ (δηλαδή οι συναρτήσεις κέρδους είναι οι ίδιες για κάθε παίκτη)» (Alos-Ferrer and Ania 2005).

Το παίγνιο Cournot με κοινές συναρτήσεις κόστους, ικανοποιεί τον παραπάνω ορισμό, δεδομένου ότι είναι συμμετρικό (δεν υπαρχει διαφοροποίηση αν «αντιμεταθέσουμε» τις επιλογές δύο παικτών) και το κέρδος του κάθε παίκτη εξαρτάται από την συνολική ποσότητα που επιλέγουν οι άλλοι παίκτες και όχι από τις μεμονωμένες τους επιλογές.

Γενικεύουν κατά κάποιο τρόπο τον παραπάνω ορισμό, ορίζοντας «οικογένειες συνολικών παιγνίων». «Μια οικογένεια συνολικών παιγνίων είναι μια συλλογή παιγνίων $\{\Gamma^n\}_{n=1}^{\infty}$, με $\Gamma = \{N, S, \pi\}$ είναι ένα γενικό συμετρικό και συνολικό παίγνιο, με σύνολική (aggregate) τέτοια ώστε $g_1(s) = s \ \forall s \in S$ και $\exists g : X \times S \to R$, τέτοια ώστε

$$g^{n+1}(s_1, s_2, ..., s_{n+1}) = g(g^n(s_1, s_2, ..., s_n), s_{n+1})$$

$\forall s_1, ..., s_{n+1} \in S$ και $\forall n \geq 1$» (Alos-Ferrer and Ania 2005).

Σημειώνουμε ότι ο ορισμός της «συνολικής» είναι επαγωγικός. Η συνθήκη $g^1(s) = s$ συνεπάγεται ότι ο περιορισμός της g^2 στο $S \times S$ συμπίπτει με την g^2 και συνεπώς είναι συμμετρική. Αυτή η επαγωγική κατασκευή, αφενός μας επιτρέπει να μιλάμε για «οικογένειες παιγνίων» με μεταβλητό αριθμό παικτών, αλλά με την ίδια στρατηγική δομή και αφετέρου μας επιτρέπει να υπολογίζουμε τα κέρδη των παικτών σαν συνάρτηση μονάχα της υποκειμενικής στρατηγικής του παίκτη και της «συνολικής» στρατηγικής των αντιπάλων του. Πραγματικά, αν έχουμε την οικογένεια $\{\Gamma^n\}_{n=1}^{\infty}$ και ορίσουμε την $\tilde{\pi} : S \times X \to R$ ως

$$\tilde{\pi}(s, x) = \pi(s, g(x, s))$$

Χρησιμοποιώντας τον ορισμό της «οικογένειας» μπορούμε να εκφράσουμε τα κέρδη του παιγνίου σαν συνάρτηση της στρατηγικής του παίκτη i και μιας συνολικής έκφρασης (συγκεκριμένα της «συνολικής» g^{n-1}) των στρατηγικών των υπολοίπων παικτών:

$$\pi(s_i, s_{-i}) = \pi(s_i, g^n(s_i, s_{-i})) = \tilde{\pi}(s_i, g^{n-1}(s_{-i}))$$

Αυτή η προσέγγιση χρησιμοποιείται για την απλοποίηση της ανάλυσης της αντιστοιχίας των βέλτιστων απαντήσεων (βλ. και Vives 2000).

Στη συνέχεια δίνουν τον ορισμό των υπο-τμηματικών (sub-modular) και υπέρ-τμηματικών (super-modular) παιγνίων (βλ. και Topkis 1998): «Λέμε ότι ένα συνολικό παίγνιο $\Gamma = (N, s, \pi)$ είναι super-modular (sub-modular) σε ατομική στρατηγική και σε συνολική (aggregate) αν το π έχει αύξουσες (αντίστοιχα φθίνουσες) διαφορές. Δηλαδή αν η

$$\pi(s'', x) - \pi(s', x)$$

Είναι αύξουσα (φθίνουσα στα υπο-τμηματικά) στο $x \in X$ $\forall s'' > s' \in S$.

Αν $X = R$ και η $\pi(x, s)$ είναι δύο φορές συνεχώς παραγωγίσιμη, τότε η π έχει άυξουσες (φθίνουσες) διαφορές αν και μόνο

$$\frac{\partial^2 \pi(s, x)}{\partial x \partial s} \geq (\leq) 0$$

Η έννοια των αυξουσών διαφορών εκφράζει την έννοια της συμπληρωματικότητας (complementarity) - το κίνητρο αύξησης της s σε μια αύξηση της x. Αντίστοιχα η έννοια των φθινουσών διαφορών εκφράζει την υποκατάσταση (substitutability) -το κίνητρο μείωσης της s σε μια μείωση της x και αντίστροφα.» (Alos-Ferrer and Ania 2005).

«Λέμε ότι ένα συνολικό παίγνιο $\Gamma = (N, s, \pi)$ είναι οιωνεί υπερ-τμηματικό (quasi super-modular) σε ατομική στρατηγική και σε συνολική (aggregate) αν η π ικανοποιεί τη συνθήκη μοναδικού περάσματος (single-crossing property). Στο $(s, x) \in S$. Δηλαδή αν $\forall s'' > s'$ και $\forall x'' > x'$

$$\pi(s'', x') \geq \pi(s', x') \Rightarrow \pi(s'', x'') \geq \pi(s', x'')$$
$$\pi(s'', x') > \pi(s', x') \Rightarrow \pi(s'', x'') > \pi(s', x'')$$

Λέμε ότι ένα συνολικό παίγνιο $\Gamma = (N, s, \pi)$ είναι οιωνεί υπο-τμηματικό (quasi sub-modular) σε ατομική στρατηγική και σε συνολική (aggregate) αν η π ικανοποιεί τη δυαδικη συνθήκη μοναδικού περάσματος (dual single-crossing property) στο $(s, x) \in S$. Δηλαδή αν $\forall s'' > s'$ και $\forall x'' > x'$

$$\pi(s'', x') \leq \pi(s', x') \Rightarrow \pi(s'', x'') \leq \pi(s', x'')$$
$$\pi(s'', x') < \pi(s', x') \Rightarrow \pi(s'', x'') < \pi(s', x'')$$

Αν δηλαδή ισχύουν οι προηγούμενες ανισότητες με αντίστροφη φορά. Η συνθήκη μοναδικού περάσματος (single-crossing property - SCP) είναι μια τακτική έκδοση της συμπληρωματικότητας πιο ασθενής από τις αύξουσες διαφορές. Αν η s'' είναι προτιμότερη της s' δεδομένου του $x = x'$, τότε θα είναι προτιμότερη και για $x = x'' > x$, αν και δεν μπορούμε να πούμε ότι το κίνητρο για την αντικατάσταση της s' με την s'' έχει αλλάξει. Ανάλογη σχέση υπάρχει μεταξύ της δυαδικής συνθήκης μοναδικού περάσματος (dual SCP) και της υποκατάστασης» (Alos-Ferer and Ania 2005).

Το παίγνιο Cournot είναι υπό-τμηματικό (sub-modular), στην ίδια στρατηγική και την ολική ποσότητα $Q = \sum_{i=1}^{n} q_i$, εφόσον η συνάρτηση ζήτησης είναι φθίνουσα ως προς το Q. Αν από την άλλη θέλουμε να δείξουμε ότι το παίγνιο είναι υπο-τμηματικό (sub-modular) στην q_i και τη στρατηγική συνολικά των αντιπάλων q_{-i}, χρειαζόμαστε την πρόσθετη προϋπόθεση των φθινουσών οριακών αποδόσεων. Είναι επίσης σύνηθες στη βιβλιογραφία να αντιμετωπίζεται το παίγνιο Cournot σαν super-modular, με κάποια αλλαγή στις μεταβλητές, όπως θίξαμε και στο κεφάλαιο 2 (βλ. και Amir 1996 ή Vives 1990).

Στη συνέχεια υιοθετούν μια επανδιατύπωση του ορισμού του Schaffer (1988) για την «εξελικτικά σταθερή στρατηγική», κατάλληλη για «συμμετρικά και συνολικά παίγνια». «Λέμε ότι η καθαρή στρατηγική s είναι μια εξελικτικά σταθερή στρατηγική του παιγνίου Γ αν $\forall s' \in S$ ισχύει

$$\Pi(s | s', s, \ldots, s) \geq \Pi(s' | s, \ldots, s)$$

Η s λέγεται «αυστηρά εξελικτικά σταθερή» αν η παραπάνω ανισότητα είναι αυστηρή (Alos-Ferrer, Ania 2005). Στην περίπτωση συμμετρικού παιγνίου δεν έχει σημασία το ποιός ακριβώς παίκτης υιοθετεί την «μεταλλαγμένη στρατηγική» s', οπότε $\Pi(s|s',s,...,s) = \Pi(s|s,s'.s,...,s)$ κ.ο.κ. για οποιαδήποτε αντιμετάθεση του προφίλ στρατηγικών των αντιπάλων. Έτσι ο ορισμός αυτός ισοδυναμεί με εκείνον του Schaffer (1988) που αναφέραμε σε προηγούμενη ενότητα.

Εφόσον ο πληθυσμός είναι πεπερασμένος, μια στρατηγική ESS δεν μεγιστοποιεί το ιδίον όφελος του πράκτορα. Αντίθετα, τα *σχετικά* κέρδη είναι αυτά που μεγιστοποιούνται. Μια απόκλιση από μια εξελικτικά σταθερή κατάσταση μπορεί να μειώσει την πιθανότητα επιβίωσης της στρατηγικής, αλλά όπως είπαμε παραπάνω, μπορεί να προκαλέσει ακόμα μεγαλύτερη μείωση στις πιθανότητες επιβίωσης των υπολοίπων στρατηγικών (spite effect). Όπως έδειξε ο Schaffer (1998) μια εξελικτικά σταθερή στρατηγική s ικανοποιεί την

$$s \in \arg\max_{s'} [\Pi(s'|s,s,...,s) - \Pi(s|s',s,...,s)]$$

Έτσι μια εξελικτικά σταθερή στρατηγική, αντιστοιχεί σε μια ισορροπία κατά Nash του παιγνίου, με σχετικά όμως κέρδη. Γενικά μια ESS πεπερασμένου πληθυσμού κατά Schaffer, δεν αντιστοιχεί σε μια ισορροπία κατά Nash του αρχικού παιγνίου, σε αντίθεση με μια ESS κατά Price και Young.

Εν συνεχεία υιοθετούν έναν διαφορετικό ορισμό για την «ολική ευστάθεια» από εκείνον του Schaffer (1988). «Λέμε ότι η ESS s είναι ολικά ευσταθής (globally stable) αν

$$\Pi\left(s \middle| s', \overset{m\ \phi o \rho \varepsilon \varsigma}{\ldots}, s', s, ..., s\right) \geq \Pi\left(s \middle| s', \overset{m-1\ \phi o \rho \varepsilon \varsigma}{\ldots}, s', s, ..., s\right)$$

για $1 \leq m \leq N-1$. Η s είναι «αυστηρά ολικά ευσταθής» αν η παραπάνω ανισότητα είναι αυστηρή». (Alos-Ferer, Ania 2005).

Όπως έδειξε ο Vega-Redondo, [Vega-Redondo (1996), “claim”], αν οι επιχειρήσεις υιοθετούν καθαρές στρατηγικές, σε ένα μοντέλο άμεσης κοινωνικής μάθησης, στο οποίο η κάθε επιχείρηση προσδιορίζει την ποσότητά της σε κάθε γύρο, και η συνολική ποσότητα προσδιορίζει την τιμή που θα χρησιμοποιηθεί για τις αποφάσεις των επιχειρήσεων στον επόμενο γύρο, ισχύει

$$\pi\left(q^w, q, \overset{n-k\ \phi o \rho \varepsilon \varsigma}{\ldots}, q, q^w, \overset{n}{\ldots}, q^w\right) = P((n-k)q + kq^w)q^w - C(q^w) > P((n-k)q + kq^w)q - C(q) = \pi\left(q, q, \overset{n-k\ \phi o \rho \varepsilon \varsigma}{\ldots}, q, q^w, \overset{n}{\ldots}, q^w\right)$$

$\forall q \in Q,\ 1 \leq k \leq n$. q^w είναι η ποσότητα που αντιστοιχεί στην Βαλρασιανή ισορροπία και η παραπάνω ανισότητα συνεπάγεται ότι η q^w είναι ευσταθής στρατηγική και μάλιστα αυστηρά (Alos-Ferrer, Ania 2005). Πιο συγκεκριμένα, βλέπουμε ότι εφόσον βρισκόμαστε σε μια κατάσταση στην οποία όλοι οι πάικτες υιοθετούν την οποιαδήποτε κοινή στρατηγική $q \in Q$, ακόμα και ένας παίκτης να υιοθετεί την q^w, η κατάσταση $(q^w, q, ..., q)$ δίνει μεγαλύτερο κέρδος στην q^w απ΄ ότι στην q. Έτσι στα πλαίσια, τόσο του εξελικτικού αλγορίθμου του Vega-Redondo, όσο και στα πλαίσια ενός γενετικού αλγορίθμου, η q^w θα τείνει να αυξηθεί σε συχνότητα στον πληθυσμό, ενώ η συχνότητα της q θα μειωθεί. Δεδομένου ότι τα παραπάνω ισχύουν και για $k > 1$, αυτή η διαδικασία θα συνεχιστεί, μέχρι $k = n$. Παρόλο που η έννοια της εξελικτικά σταθερής στρατηγικής και η ευστάθειά της, δεν καλύπτουν τις περιπτώσεις που οι παίκτες επιλέγουν διαφορετικές στρατηγικές και όχι μια κοινή στρατηγική q, το αποτέλεσμα αυτό αποτελεί μια ένδειξη για το τι πρόκειται να συμβεί στις σχετικές προσομοιώσεις και, δεδομένου ότι πράγματι αυτό είναι που συμβαίνει, μια λογική εξήγηση για τη σύγκλιση του πληθυσμού στην q^w.

Όπως υπονοήσαμε παραπάνω και αναλύσαμε και σε προηγούμενη ενότητα, τα παραπάνω ισχύουν ακόμα κι αν $q = q^N$, με q^N την ισορροπία κατά Nash. Έστω ότι βρισκόμαστε στην κατάσταση $(q_N, q_N, ..., q_N)$ και για οποιονδήποτε λόγο ένας παίκτης υιοθετεί την q_w. Παρόλο που η q_N είναι σαφέστατα καλύτερη επιλογή για τον εν λόγω παίκτη, με δεδομένο ότι οι αντιπάλοι του επιλέγουν την q_N , αφού

$$q_N = \arg\max_q \pi\left(q | q_N, ..., q_N\right)$$

Η κατάσταση $(q_w, q_N, ..., q_N)$ οδηγεί σε ένα παίγνιο με μεγαλύτερο κέρδος για τον παίκτη που παίζει την q_w από το κέρδος εκείνων που παίζουν την q_N. Έτσι, η συχνότητα της q_W στον πληθυσμό τείνει να αυξάνεται.

Αλλά και στην περίπτωση που βρισκόμαστε στην $(q_W, ..., q_W)$, ακόμα και αν κάποιος παίκτης επιλέξει τη βέλτιστη -για τη δεδομένη κατάσταση

$$q_o = \arg\max_q \pi\left(q | q_W, ..., q_W\right)$$

οι παίκτες που παίζουν q_w θα κερδίζουν περισσότερο από αυτόν στην $(q_o, q_w, ..., q_w)$, οπότε ο πληθυσμός θα τείνει να επιστρέψει στον $(q_w, ..., q_w)$.

Arifovic και Maaschek (2006): Revisiting Individual Evolutionary Learning in the Cobweb Model – An Illustration of the Virtual Spite Effect

Οι Arifovic και Maaschek προσπαθούν να αντικρούσουν τα επιχειρήματα του Vriend (2000), αναφορικά με το γεγονός της σύγκλισης του αλγόριθμου ατομικής μάθησης του τελευταίου στην ισορροπία Nash. Στην ανάλυσή τους καταλήγουν ότι η σύγκλιση στην ισορροπία Nash οφείλεται σε δύο λόγους: αφενός στον συγκεκριμένο τρόπο με τον οποίο υπολογίζεται η απόδοση των κανόνων παραγωγής και αφετέρου στις συγκεκριμένες συναρτήσεις κόστους που χρησιμοποίησε ο Vriend.

Πιο συγκεκριμένα ο βασικός λόγος, κατά τον Vriend που αποφεύγεται η σύγκλιση στην ισορροπία Walras είναι η έλλειψη του spite effect στην περίπτωση του της ατομικής μάθησης. Το spite effect δεν επηρρεάζει τη διαδικασία μάθησης και συνεπώς οι παίκτες μπορούν να συγκλίνουν στην ισορροπία κατά Nash. Αντίθετα, υποστηρίζουν οι Arifovic, Maaschek, το spite effect μπορεί να παρατηρηθεί και στην περίπτωση ατομικής μάθησης. Για να το δείξουν αυτό, συγκρίνουν τον «απλό» αλγόριθμο ατομικής μάθησης πολλαπλών πληθυσμών της Arifovic (1994) με τον αλγόριθμο που χρησιμοποίησε ο Vriend.

Αρχικά υιοθετούν τη συνάρτηση ζήτησης

$$P = a + bQ^c$$

και τη συνάρτηση κόστους

$$C(q_i) = K + kq_i$$

που χρησιμοποίησε ο Vriend ώστε να είναι συγκρίσιμοι οι αλγόριθμοι.

Με αυτές τις συναρτήσεις ζήτησης και κόστους, η ισορροπία Walras δίνεται από την

$$Q^W = \left(\frac{k - a}{b}\right)^{\frac{1}{c}}$$

ενώ η αντίστοιχη ισορροπία κατά Nash από την

$$Q^N = \left[\frac{k - a}{b\left(\frac{c}{m} + 1\right)}\right]^{\frac{1}{c}}$$

Οι σημαντικότερες διαφορές μεταξύ του «απλού» αλγορίθμου ατομικής μάθησης πολλαπλών πληθυσμών της Arifovic (1994) και του Vriend (2000) είναι (Arifovic and Maaschek, 2006):

- Στον αλγόριθμο της Arifovic (1994), ολόκληρος ο πληθυσμός υπόκειται σε αναπαραγωγή, ενώ στον αλγόριθμο του Vriend (2000), μονάχα ένα μέρος του πληθυσμού αναπαράγεται κάθε φορά.
- Στον αλγόριθμο της Arifovic(1994), μπορούμε να έχουμε δύο χρωμοσώματα που απεικονίζουν την ίδια ποσότητα, κάτι που δεν συμβαίνει στο classifier system του Vriend (2000).
- Ο τρόπος που επιλέγεται η ποσότητα που θα προσφέρει τελικά η επιχείρηση είναι τυχαίος, σύμφωνα με τον κανόνα της ρουλέττας (πιθανότητα επιλογής ανάλογη της προσαρμοστικότητας) στην περίπτωση της Arifovic (1994), ενώ στην περίπτωση του Vriend (2000), η επιλογή γίνεται με stochastic auction.
- Στον αλγόριθμο της Arifovic (1994) υπολογίζεται η προσαρμοστικότητα όλων των χρωμοσωμάτων, με τη χρήση υποθετικών κερδών, ενώ στον αλγόριθμο του Vriend (2000) αλλάζει η προσαρμοστικότητα των χρωμοσωμάτων εκείνων που συμμετέχουν ενεργά στην διαμόρφωση της κατάστασης στην αγορά.

- Τέλος στον αλγόριθμο της Arifovic (1994), ο επανυπολογισμός των προσαρμοστικοτήτων γίνεται σε κάθε γύρο της προσομοίωσης, ενώ στον αλγόριθμο του Vriend (2000), κάθε 100 γύρους.

Για τις τιμές των διαφόρων παραμέτρων που έθεσαν οι Arifovic, Maaschek (2006) βλ. το σχετικό άρθρο.

Οι ερευνητές προσπάθησαν να διαπιστώσουν ποιές είναι οι διαφορές εκείνες που οδηγούν στην «εξαφάνιση» του spite effect στον αλγόριθμο του Vriend (2000). Έτσι διενέργησαν προσομοιώσεις στις οποίες τροποποιούσαν τον αλγόριθμο της Arifovic (1994), κατά τρόπο ώστε οι διαφορές αυτές να εξαλείφονται σταδιακά. Διαπίστωσαν ότι ο απλός αλγόριθμος ατομικής μάθησης της Arifovic (1994) συγκλίνει σχεδόν σε όλες τις περιπτωσεις στην ισορροπία Walras. Το spite effect συνεχίζει δηλαδή να υφίσταται και στην περίπτωση της ατομικής μάθησης (οι Maaschek, Arifovic ονομάζουν το spite effect σε αυτήν την περίπτωση, *virtual spite effect*). Ωστόσο δύο συγκεκριμένα χαρακτηριστικά του αλγόριθμου του Vriend (2000) εμποδίζουν την σύγκλιση στην ισορροπία κατά Walras. Αφενός η υπόθεση μηδενικού οριακού κόστους που πρότεινε $(k = 0)$ και αφετέρου η μη αναπροσαρμογή της προσαρμοστικότητας των χρωμοσωμάτων σε κάθε γύρο της προσομοίωσης. « *Ας υποθέσουμε ότι [βρισκόμαστε στην Βαλρασιανή ισορροπία] και μια επιχείρηση υιοθετεί μια ποσότητα, χαμηλότερη από την ποσότητα ισορροπίας. Αυτό αυξάνει την τιμή πάνω από την τιμή ισορροπίας. Εφόσον υποθέτουμε μηδενικό οριακό κόστος, οπότε και μηδενική τιμή ισορροπίας, η τιμή θα είναι πλέον θετική. Ας πάρουμε πρώτα την επιχείρηση που άλλαξε τη στρατηγική της. Δεδομένου ότι δεν χρησιμοποιούνται υποθετικά κέρδη, μονάχα η προσαρμοστικότητα της επιχείρησης αυτής θα αλλάξει. Η προσαρμοστικότητα αυτή είναι χαμηλότερη από εκείνη που είχε η επιχείρηση, στην κατάσταση ισορροπίας, οπότε η επιχείρηση τείνει να επιστρέψει στην ποσότητα ισορροπίας κατά Walras.*

Ας πάρουμε τώρα τις άλλες επιχειρήσεις. Αυτές παίζουν ακόμα την στρατηγική ισορροπίας. Η μετάλλαξη της στρατηγικής της μεμονομένης επιχείρησης οδήγησε στην αύξηση των κερδών των επιχειρήσεων αυτών. η προσαρμοστικότητά των επιχειρήσεων αυτών δεν μεταβάλλεται, στον αλγόριθμο του Vriend. Έτσι η στρατηγική της ισορροπίας κερδίζει σχετική προσαρμοστικότητα σε σχέση με όλες τις υπόλοιπες. Παραμένει πιθανό να χρησιμοποιηθεί στις επόμενες περιόδους.

Συνδιάζοντας τα παραπάνω, διαπιστώνουμε ότι υπάρχει μεγάλη εξελικτική πίεση για επαναφορά στην ισορροπία, στην περίπτωση που μια επιχείρηση αποκλίνει, παράγοντας μικρότερη ποσότητα. Αυτό δεν συμβαίνει όμως στην περίπτωση που η ποσότητα είναι μεγαλύτερη.

Ας δούμε τι συμβαίνει όταν μια επιχείρηση αποκλίνει από την ισορροπία, παράγοντας ελαφρώς μεγαλύτερη ποσότητα. Κάτω από την υπόθεση μηδενικού οριακού κόστους (και άρα μηδενικής τιμής στην ανταγωνιστική ισορροπία), η τιμή θα γίνει αρνητική με αύξηση της συνολικής προσφοράς. Από *την πλευρά του παίκτη που μετέβαλλε την στρατηγική του, το κέρδος του είναι μικρότερο από το κέρδος που είχε παίζοντας την στρατηγική ισορροπίας και δεδομένου ότι η προσαρμοστικότητά του μεταβάλλεται, το ίδιο συμβαίνει και σε εκείνη. Συνεπώς η επιχείρηση πιθανώς θα επιστρέψει στην προηγούμενή της στρατηγική.*

Όσον αφορά τώρα τις υπόλοιπες επιχειρήσεις, η προσαρμοστικότητά της στρατηγικής ισορροπίας είναι σημαντικά χαμηλότερη στη νέα κατάσταση που προκύπτει μετά την μετάλλαξη. Η προσαρμοστικότητα του κανόνα αυτού πέφτει και κάνει την υιοθέτηση της στρατηγικής ισορροπίας λιγότερο πιθανή στο μέλλον. Καθώς συμβαίνει αυτό, αυξάνει η πιθανότητα πολλές από τις επιχειρήσεις να αποκλίνουν σε πειραματισμούς στο μέλλον.

Αυτό είναι το σημείο που η συγκεκριμένη υπόθεση μηδενικού οριακού κόστους γίνεται σημαντική. Όταν οι τιμές ωθούνται στην περιοχή των αρνητικών τιμών, λόγω μιας μετάλαξης της στρατηγικής μιας επιχείρησης από το σημείο ισορροπίας, η ποσότητα που μεγιστοποιεί το κέρδος (το μηδενίζει) είναι η $q = 0$. *Αυτό ενθαρύνει αρκετά μεγάλες περικοπές στην παραγωγή. Έτσι οι επιχειρήσεις που υιοθετούν χαμηλότερες ποσότητες, καθώς πειραματίζονται με μεταλλαγμένες στρατηγικές, θα έχουν μεγάλη αύξηση στην προσαρμοστικότητά τους, με δεδομένο ότι οι τιμές θα παραμείνουν αρνητικές.*

Δεν είναι αυτή η κατάσταση όταν η τιμή [ισορροπίας] είναι θετική. Εδώ, όταν μια απόκλιση ενθαρύνει μια περικοπή στην παραγωγή, δε χρειάζεται η περικοπή αυτή να είναι τόσο δραστική, όσο εκείνη που σχετίζεται με μια προσομοίωση με μηδενική τιμή ισορροπίας, αφού το μέγιστο επιτρεπτό κέρδος [σε μια πτώση της τιμής κάτω από το 0] δεν είναι [κατανάγκη] μηδενικό.

Ο καθορισμός του κόστους με τον τρόπο του Vriend, ενθαρύνει και ανταμείβει μεγάλες περικοπές στην ποσότητα. Η συνολική ποσότητα πέφτει σημαντικά κι έτσι εκείνοι που παίζουν τις στρατηγικές μειωμένης ποσότητας απολαμβάνουν τα υψηλότερα κέρδη. Καθώς οι μεγάλες περικοπές ανταμείβονται, η ισορροπία κατά Nash επωφελείται.

Από την άλλη, η διατήρηση της ισορροπίας κατά Nash απαιτεί διαρκή συντονισμό στις χαμηλότερες ατομικές ποσότητες. Συνεπώς θα ανέμενε κανείς ότι η προσομοίωση θα συνέκλινε πάλι στην ανταγωνιστική ισορροπία. Το γιατί αυτό δεν συμβαίνει οφείλεται στο γεγονός ότι δεν χρησιμοποιούνται υποθετικά κέρδη και δυνητικές προσαρμοστικότητες.

Για να επιστρέψει η προσομοίωση στα ανταγωνιστικά επίπεδα, οι επιχειρήσεις θα πρέπει να αρχίσουν να αυξάνουν τα επίπεδα παραγωγής. Η πιθανότητα να συμβεί κάτι τέτοιο, είναι άμεσα συνδεδεμένη με την προσαρμοστικότητα των στρατηγικών αυτών. Ωστόσο οι προσαρμοστικότητες των στρατηγικών μεταβάλλονται μονάχα όταν παίζονται οι στρατηγικές. Έτσι, όταν οι επιχειρήσεις σκέφτονται να παίξουν αυτές τις στρατηγικές, οι προσαρμοστικότητές που λαμβάνουν υπόψην είναι *εκείνες που είχαν οι στρατηγικές την τελευταία φορά που παίχτηκαν. Οι επιχειρήσεις θυμούνται μονάχα την προσαρμοστικότητα που είχαν οι στρατηγικές στην περιοχή των αρνητικών τιμών, η οποία ήταν ιδιαίτερα χαμηλή. Έτσι η πιθανότητα υιοθέτησης των στρατηγικών αυτών είναι μικρή.*

Έτσι, αν και υπάρχει ακόμα ένα virtual spite effect, εμποδίζεται από δύο παράγοντες: την αναπροσαρμογή της προσαρμοστικότητας των στρατηγικών που χρησιμοποιούνται μόνο και την υιοθέτηση της υπόθεσης μηδενικού οριακού κόστους.

Από την άλλη, όταν αναπροσαρμόζουμε τις προσαρμοστικότητες των στρατηγικών με βάση τα υποθετικά κέρδη που θα είχαν εάν εφαρμοζόταν, το virtual spite effect γίνεται ένας από τους σημαντικότερους προσδιοριστικούς παράγοντες της εξέλιξης του συστήματος. [...] Οι μεγαλύτερες ταχύτητες μάθησης, που χαρακτηρίζουν το μοντέλο στο οποίο όλες οι προσαρμοστικότητες αναπροσαρμόζονται με βάση υποθετικά κέρδη, επιτρέπει την ύπαρξη spite effect. Έτσι ένα πιο αργό και λιγότερο σύνθετο σύστημα, μπορεί στην συγκεκριμένη περίπτωση να μας βοηθήσει να πετύχουμε το επιδιωκόμενο αποτέλεσμα». (Arifovic, Maschek, 2006).

Οι Valee, Yildizoglou (2007) δείχνουν, όπως θα δούμε αμέσως παρακάτω, ότι η σημαντική συνθήκη είναι η μη αναθεώρηση της προσαρμοστικότητας των χρωμοσωμάτων με τη χρήση υποθετικών κερδών. Εφόσον δεν αναθεωρούνται οι προσαρμοστικότητες όλων των χρωμοσωμάτων, έχουμε σύγκλιση στην ισορροπία κατά Nash και αντίθετα, όταν αναθεωρούνται, έχουμε σύγκλιση στην ισορροπια κατά Walras.

Valee, Yildizoglou (2007), Convergence In Cournot Oligopoly with Social and Individual Learning

Οι Valee, Yildizoglou (2007) διερεύνησαν μαθηματικά τους αλγορίθμους της Arifovic (1994) και Vriend (2000) σε συμμετρικά παίγνια Cournot με φθίνουσες συναρτήσεις αντίστροφης ζήτησης και κυρτές και αύξουσες συναρτήσεις κόστους (άρα και κοίλες συναρτήσεις κέρδους). Διαπίστωσαν έτσι τις συνθήκες που οδηγούν στη σύγκλιση στη Βαλρασιανή ή την ισορροπία κατά Nash σε κάθε περίπτωση.

Πρώτα απ΄ όλα οι Valee, Yildizoglou, έδωσαν έναν μαθηματικό ορισμό για το spite effect: «χρησιμοποιούμε τον όρο spite effect όταν η μεταλλαγμένη στρατηγική j υποφέρει από αρνητική απόκλιση [στο κέρδος] $\Delta\pi_j < 0$, αλλά υποφέρει λιγότερο από τις μη μεταλλαγμένες στρατηγικές $(\Delta\pi_j < \Delta\pi_i < 0) \quad \forall i \neq j$. Ομοίως, στη θετική κατεύθυνση, μπορούμε να ερμηνεύσουμε το φαινόμενο, σαν το γεγονός ότι το καθαρό κέρδος για την μεταλλαγμένη επιχείρηση $(\Delta\pi_j > 0)$ είναι μεγαλύτερο από το κέρδος των μη μεταλλαγμένων στρατηγικών $(\Delta\pi_j > \Delta\pi_i) \quad \forall i \neq j$, κάτι που ισχύει πάντα εφόσον οι μη μεταλλαγμένες στρατηγικές υποφέρουν από αρνητικές αποκλίσεις. Τελικά σε ένα παίγνιο με πεπερασμένο αριθμό χρωμοσωμάτων, λέμε ότι υπάρχει spite effect όταν, μετά από μια απόκλιση της επιχείρησης j απ΄ την στρατηγική ισορροπίας,

$$\Delta \pi_i < \Delta \pi_j \qquad \forall\, i \neq j$$ » (Valee, Yildizoglou 2007)

Η σχέση μεταξύ spite effect, ESS (κατά Schaffer) και ισορροπίας κατά Nash προκύπτει άμεσα :

«Σε ένα πεπερασμένο και συμμετρικό παίγνιο, ένα σημείο αυστηρής ισορροπίας κατά Nash είναι εξελικτικά ευσταθές κατά Schaffer, εφόσον δεν υπάρχει spite effect» (για απόδειξη βλ. Valee and Yildizoglou 2007).

Στη συνέχεια αποδεκνύουν μια βοηθητική πρόταση, ώστε να διερευνήσουν την εξελικτική ευστάθεια των αλγορίθμων των Vriend και Arifovic που είδαμε παραπάνω:

« Σε ένα πεπερασμένο και συμμετρικό παίγνιο, το προφίλ στρατηγικών $\hat{q} = (\hat{q}_1,...,\hat{q}_n)$ είναι ESS κατά Schaffer και δεν υπάρχει spite effect, όταν

$$dq_i(p' dq_i + p - C') < 0 \qquad \forall\, i$$

με την $q_i = \hat{q}_i + dq_i$ να είναι μια τοπική μετάλλαξη της $\hat{q}_i$ $(dq_i \approx 0)$ ».

Αμέσως μετά αποδεικνύουν την πολύ σημαντική πρόταση: «Στην περίπτωση μιμητικής κοινωνικής μάθησης [το σχέδιο μάθησης των περισσοτέρων αλγορίθμων κοινωνικής μάθησης που περιγράψαμε παραπάνω, στο οποίο έχουμε έναν πληθυσμό για όλες τις επιχειρήσεις, η τιμή στην αγορά αλλάζει σε κάθε γύρο της προσομοίωσης, όπως και οι προσαρμοστικότητες των χρωμοσωμάτων], οι στρατηγικές που συνεπάγονται μια απόκλιση απ΄ τη Βαλρασιανή ισορροπία θα εξαφανιστούν [από τον πληθυσμό], ενώ οι στρατηγικές που οδηγούν σε μια απόκλιση απ΄ την ισορροπία κατά Nash, μπορούν να διαχυθούν στον πληθυσμό».

Στη συνέχεια συγκρίνουν τους αλγορίθμους κοινωνικής και ατομικής μάθησης της Arifovic (1994) και του Vriend (2000), τόσο στην περίπτωση που ο πληθυσμός των επιχειρήσεων είναι πεπερασμένος, όσο και άπειρος.

Στην περίπτωση κοινωνικής μάθησης, αποδεικνύουν ότι εφόσον ο πληθυσμός είναι άπειρος θα έχουμε σύγκλιση στην Βαλρασιανή ισορροπία. Στην περίπτωση πεπερασμένου πληθυσμού, θα πρέπει επιπρόσθετα να έχουμε θετικά κέρδη σε μια γειτονιά της Βαλρασιανής ισορροπίας, ώστε η σύγκλιση να είναι εγγυημένη.

Όσον αφορά τους αλγόριθμους ατομικής μάθησης, αποδεικνύουν ότι η αναθεώρηση όλων των προσαρμοστικοτήτων, με βάση τα υποθετικά κέρδη που θα έδινε η εφαμογή των συγκεκριμένων στρατηγικών, συνεπάγεται ότι η Βαλρασιανή ισορροπία είναι εξελικτικά ευσταθής, ενώ η ισορροπία κατά Nash, όχι. Αν από την άλλη πλευρά οι προσαρμοστικότητες αναθεωρούνται μονάχα όταν χρησιμοποιείται η στρατηγική και ο πληθυσμός «ανανεώνεται» κάθε GA-rate γύρους (Vriend 2000), τότε, το αναμενόμενο κέρδος της κάθε στρατηγικής j του πληθυσμού του πράκτορα i θα είναι

$$E[\pi_i, (q_{ij}, Q_{-i})] = E[p(q_{ij}, Q_{-i})]q_{ij} - C(q_{ij}) = \bar{p}q_{ij} - C(q_{ij})$$

με την $\bar{p}$ να δίνεται απ΄ την

$$\bar{p} = \sum_{l \neq j} p\left(q_{ij} \sum_{l} q_{lj} \right) f(q_{lj}|GArate)$$

όπου $f(q_{lj}|GArate)$ η συχνότητα που αναμένεται να χρησιμοποιηθεί η q_{ij}, η οποία εξαρτάται φυσικά από το GA-rate. Καθώς $GArate \to \infty$, κάθε στρατηγική στον πληθυσμό ενός πράκτορα θα τείνει να χρησιμοποιείται f_{ij} φορές, όση δηλαδή και η συχνότητά της στον πληθυσμό. Αυτό, σημειώνουν οι Valee και Yildizoglou, οδηγεί σε ένα αναμενόμενο κέρδος για την q_{ij}, που συνεπάγεται $\frac{\partial \bar{p}}{\partial q_{ij}} < 0$, κάτι που συνεπάγεται με τη σειρά του ότι οι στρατηγικές που χαρακτηρίζονται από την $\bar{p} + q_{ij}\frac{\partial \bar{p}}{\partial q_{ij}} = p$ θα κυριαρχούνται από τις στρατηγικές που βρίσκονται πλησιέστερα στη συνθήκη

$$\bar{p} + q_{ij}\frac{\partial \bar{p}}{\partial q_{ij}} < \bar{p}$$

κάτι που, όπως δείχνουν στην μελέτη τους, καθιστά πιο πιθανή τη σύγκλιση στην ισορροπία κατά Nash.

Για να διερευνήσουν περισσότερο το ζήτημα, διενεργούν προσομοιώσεις του υποδείγματος ατομικής μάθησης του Vriend (2000) για διαφορετικές τιμές του GArate. Για κάθε τιμή του GArate, κάνουν τρεις διαφορετικές προσομοιώσεις, μια για κάθε διαφορετικό τρόπο επιλογής της στρατηγικής που θα χρησιμοποιήσει η επιχείρηση στον τρέχοντα γύρο. Εξετάζουν την τυχαία επιλογή, την επιλογή με τον κανόνα της ρουλέττας και την επιλογή του καλύτερου χρωμοσώματος του πληθυσμού (δηλαδή εκείνου που έχει αποκτήσει την υψηλότερη προσαρμοστικότητα, είτε στις προηγούμενες «εποχές» του αλγορίθμου είτε στις τρέχουσες $< GArate$ εκτελέσεις, στις οποίες ο πληθυσμός των χρωμοσωμάτων-στρατηγικών της επιχείρησης είναι σταθερός).

Όπως διαπιστώνουμε από τα αποτελέσματα των προσομοιώσεων, για $GArate = 1$, και οι τρεις αλγόριθμοι συγκλίνουν σε μια περιοχή που σχετίζεται περισσότερο με την Βαλρασιανή ισορροπία.

Μια τιμή $GArate = 10$ είναι από την άλλη αρκετή, για να συγκλίνουμε και στις τρεις περιπτώσεις, στην περιοχή γύρω από την ισορροπία κατά Nash. Υψηλότερες τιμές της GA-rate (50,100) δίνουν ακόμα καλύτερη σύγκλιση γύρω από την ισορροπία κατά Nash. Οι τρεις τρόποι επιλογής της ποσότητας παραγωγής δεν επηρρεάζουν σημαντικά την σύγκλιση. Θα μπορούσαμε να πούμε ότι η επιλογή του καλύτερου χρωμοσώματος εμφανίζει ελαφρώς χειρότερα αποτελέσματα από τις άλλες δύο, για δεδομένη τιμή της GA-rate. Η τιμή αυτή της GA-rate όμως, είναι εκείνη που καθορίζει την σύγκλιση ή μη στην ισορροπία Nash (σε γενικές γραμμές μια επιλογή $GArate \geq 10$ έχει ικανοποιητικότατα αποτελέσματα).

Alkemade et al (2007): On Social Learning and Robust Evolutionary Algorithm Design in the Cournot Oligopoly Game

Οι Alkemade et al (2007), εισάγουν έναν νέο αλγόριθμο κοινωνικής μάθησης. Θεωρούν ότι η σύγκλιση των προηγούμενων αλγορίθμων κοινωνικής μάθησης στην Βαλρασιανή αντί της ισορροπίας κατά Nash, οφείλεται στο γεγονός ότι κάθε πράκτορας αντιπροσωπεύεται με ένα χρωμόσωμα στον κοινό πληθυσμό, αντί -όπως στην περίπτωση των αλγορίθμων ατομικής μάθησης της Arifovic (1994) και Vriend (2000), που έχουμε έναν πληθυσμό για κάθε πράκτορα- έναν μεγαλύτερο αριθμό χρωμοσωμάτων.

Εισάγουν λοιπόν έναν νέο γενετικό αλγόριθμο κοινωνικής μάθησης στον οποίο τα χρωμοσώματα στον κοινό πληθυσμό είναι περισσότερα από τον αριθμό των επιχειρήσεων, για να εξετάσουν μια περίπτωση ολιγοπωλίου Cournot με 4 παίκτες. Οι συναρτήσεις κόστους και (αντίστροφης) ζήτησης που χρησιμοποιούν είναι αρκετά απλές:

$$P = 256 - Q, \text{ με } Q = \sum_{i=1}^{4} q_i$$

$$c(q_i) = 56q_i$$

Συνεπώς η συνάρτηση κέρδους του κάθε πράκτορα θα είναι

$$\pi_i = (256 - Q)q_i - 56q_i$$

Οι παραπάνω εξισώσεις συνεπάγονται ότι η ποσότητα της Βαλρασιανής ισορροπίας θα είναι $q_W = 50$ και της ισορροπίας κατά Nash $q_N = 40$.

Ο γενετικός αλγόριθμος που χρησιμοποιούν έχει ως εξής:

1. Κάθε πράκτορας επιλέγει τυχαία ένα από τα χρωμοσώματα του πληθυσμού, σαν στρατηγική.
2. Υπολογίζεται η τιμή και η ποσότητα ισορροπίας για τις δεδομένες στρατηγικές
3. Η προσαρμοστικότητα του χρωμοσώματος ισούται με το κέρδος του παίκτη που το χρησιμοποίησε
4. Επανέλαβε τα προηγούμενα βήματα εως ότου χρησιμοποιηθούν όλα τα χρωμοσώματα. Στη συνέχεια ανανέωσε τον πληθυσμό κατά τα γνωστά (επιλογή, διασταύρωση, μετάλλαξη).

Ο μικρότερος πληθυσμός που χρησιμοποίησαν είχε 4 χρωμοσώματα (1 χρωμόσωμα για κάθε επιχείρηση), ενώ ο μεγαλύτερος 200. Το μήκος των χρωμοσωμάτων που δοκίμασαν ήταν 6-9 δυαδικών ψηφίων, με την ποσότητα να προσδιορίζεται απλά από την δυαδική τιμή του χρωμοσώματος (οπότε η μέγιστη ποσότητα ήταν 64 για την περίπτωση των 6 δυαδικών ψηφίων, κ.ο.κ). Η πιθανότητα εφαρμογής του τελεστή διασταύρωσης ήταν $q_{cros} = 1$ και η πιθανότητα μετάλλαξης (για κάθε bit) $p_m = 0.01$. Τέλος χρησιμοποίησαν την παράμετρο "initial density" για να αρχικοποίησουν τον πληθυσμό. Η τιμή της αρχικής πυκνότητας (initial density) προσδιορίζει την πιθανότητα, το κάθε bit του αρχικού χρωμοσώματος να πάρει την τιμή 1. Χρησιμοποίησαν τις τιμές 0.1, 0.5 και 0.9.

Οι προσομοιώσεις τους έδειξαν ξεκάθαρα ότι ένας πληθυσμός 100 χρωμοσωμάτων συγκλίνει στην περιοχή γύρω από την ισορροπία κατά Nash. Το ίδιο βέβαια συμβαίνει και για μεγαλύτερους πληθυσμούς. Η σύγκλιση είναι ανεξάρτητη από τις τιμές της "initial density" ή του μήκους του χρωμοσώματος.

Στην περίπτωση που ο πληθυσμός αποτελείται από 4 χρωμοσώματα, η συμπεριφορά είναι απρόβλεπτη, και διαφέρει τόσο για διαφορετικές τιμές της αρχικής πυκνότητας, όσο και για διαφορετικά μήκη χρωμοσώματος. Αυτό θα πρέπει να θεωρείται αναμενόμενο, δεδομένου ότι ο αριθμός των χρωμοσωμάτων είναι πολύ μικρός για να λειτουργήσει αποτελεσματικά ο γενετικός αλγόριθμος.

Ανακεφαλαίωση

Όπως είδαμε, τόσο οι αλγόριθμοι της Arifovic (1994), όσο και οι αλγόριθμοι των Dawid and Kopel (1998), Franke (1998), Vriend (2000- social learning), όσο και ο απλούστερος εξελικτικός αλγόριθμος του Vega-Redondo, συγκλίνουν στην Βαλρασιανή ισορροπία. Οι ερευνητές, πέρα από την παράθεση αποτελεσμάτων προσομοιώσεων, επιχειρηματολογούν κυρίως με βάση το "spite effect" για να αιτιολογήσουν την εξέλιξη αυτή. Οι Alos-Ferer and Ania (2005) και οι Valee and Yildizoglou (2007) δείχνουν ότι η έννοια της εξελικτικά ευσταθούς στρατηγικής (ESS) όπως τροποποιήθηκε από τον Schaffer (1998) για πεπερασμένους πληθυσμούς, ισχύει για την Βαλρασιανή ισορροπία. Βέβαια για να ισχύει η παρατήρηση αυτή, θα πρέπει ο εξελικτικός αλγόριθμος να χαρακτηρίζεται από την συνεχή ανανέωση των χρωμοσωμάτων του πληθυσμού και από μη στοχαστικά κέρδη. Αυτές οι συνθήκες δεν ισχύουν στους αλγορίθμους ατομικής μάθησης του Vriend (2000) και στον αλγόριθμο κοινωνικής μάθησης των Alkemade et al (2007), στους οποίους το κέρδος της κάθε στρατηγικής υπολογίζεται μονάχα τη στιγμή που αυτή χρησιμοποιείται, και εξαρτάται φυσικά από τις αντίπαλες στρατηγικές που επιλέγονται με τυχαίο τρόπο. Το κέρδος είναι δηλαδή τυχαία μεταβλητή. Το γεγονός αυτό, σε συνδιασμό με τον αυξημένο αριθμό επαναληψεων του παιγνίου (GA-Rate στον αλγόριθμο του Vriend και μέγεθος πληθυσμού σε σχέση με τον αριθμό των παικτών στον αλγόριθμο των Alkemade et al.) οδηγεί σε προσαρμοστικότητες που βρίσκονται κοντά στο αναμενόμενο κέρδος των χρωμοσωμάτων, όταν έρχονται αντιμέτωπα με τυχαία αντίπαλη στρατηγική. Οι διαφορές αυτές, οδηγούν τελικά στην σύγκλιση στην ισορροπία κατά Nash, τουλάχιστον για την τιμή και τη συνολική ποσότητα στην αγορά.

Σημειώνουμε εδώ ότι ούτως ή άλλως, η έννοια της εξελικτικής ευστάθειας δεν τεκμηριώνει, από μαθηματικής άποψης, την σύγκλιση ή μη σε μια κατάσταση ισορροπίας. Αποτελεί απλά μία αναγκαία συνθήκη στην περίπτωση μιας στρατηγικής για κάθε παίκτη και αναπροσαρμογής του προφίλ στρατηγικών μετά από κάθε παίγνιο, με προσαρμοστικότητα της κάθε στρατηγικής ανάλογη με το κέρδος που πέτυχε η στρατηγική στο εν λόγω παίγνιο. Ο γενικός τρόπος να αποδείξουμε μαθηματικά την σύγκλιση ή μη σε μια συγκεκριμένη κατάσταση (ή στις περιπτώσεις εργόδικων διαδικασιών Markov σε ένα σύνολο καταστάσεων που εμφανίζονται με δεδομένη συχνότητα καθώς ο κύκλος των επαναλήψεων τείνει στο άπειρο), είναι να προσεγγίσουμε τη διαδικασία Markov στην οποία οδηγεί ο αλγόριθμος. Βέβαια αυτό είναι κατ΄ αρχήν αδύνατο σε έναν γενετικό αλγόριθμο, αφού το πλήθος των δυνατών καταστάσεων είναι $2^{N \cdot l}$ και συνεπώς το πλήθος των πιθανοτήτων μετάβασης είναι 2^{2Nl}. Ακόμα και η στατιστική εκτίμηση των συχνοτήτων του διανύσματος σταθερών συχνοτήτων $\mathbf{w}$ (βλ. κεφ. 2) απαιτεί ένα ιδιαίτερα μεγάλο αριθμό παρατηρήσεων. Ωστόσο μπορούμε να προσεγγίσουμε το πρόβλημα αυτό στατιστικά, αν ορίσουμε κατάλληλα τις μεταβλητές κατάστασης, όπως θα δούμε παρακάτω.

5. Ανάλυση κυριότερων αλγορίθμων

5.1. Εισαγωγή

Στο προηγούμενο κεφάλαιο είδαμε ότι οι περισσότεροι γενετικοί αλγόριθμοι που εφαρμόστηκαν μέχρι στιγμής στο πρόβλημα της επίλυσης μέσω εξελικτικών διαδικασιών του παιγνίου Cournot, δεν συγκλίνουν στην ισορροπία κατά Nash, αλλά στην ισορροπία κατά Walras, που χαρακτηρίζει τις ανταγωνιστικές αγορές. Οι μοναδικοί αλγόριθμοι που, όπως ανέφεραν οι ερευνητές, οδηγούν στη σύγκλιση της αγοραίας τιμής και (ισοδύναμα, αφού είναι αμφιμονοσήμαντη συνάρτηση της τιμής) της συνολικής και της μέσης ποσότητας στις τιμές εκείνες που προβλέπει η έννοια της ισορροπίας κατά Nash, είναι οι αλγόριθμοι ατομικής μάθησης του Vriend (2000) και ο αλγόριθμος κοινωνικής μάθησης των Alkemade et al (2007).

Εμείς θα εφαρμόσουμε τους αλγόριθμους αυτούς, μαζί με εκείνον της Arifovic (1994), αλλά και του co-evolutionary programming (Price 1997), ο οποίος είχε παρόμοιες ιδιότητες όταν εφαρμόστηκε από τον Price στο παίγνιο του διλήμματος του φυλακισμένου, σε μια σειρά υποδειγμάτων Cournot με καθαρές στρατηγικές, ώστε να διερευνήσουμε όχι μόνο την μακροοικονομική αλλά και τη μικροοικονομική σύγκλιση των αλγορίθμων στις αντοίστιχες ισορροπίες. Θέλουμε δηλαδή να διερευνήσουμε την διαχρονική πορεία των ατομικών επιλογών των πρακτόρων, στην διάρκεια της προσομοίωσης, επιπρόσθετα με την σύγκλιση των βασικών μακροοικονομικών μεγεθών της τιμής και της ποσότητας στην αγορά. Το βασικό ερώτημα είναι βέβαια, το αν όλοι οι παίκτες συγκλίνουν σε μια συμπεριφορά η οποία χαρακτηρίζεται από την επιλογή της ποσότητας της ισορροπίας κατά Nash (Walras στην περίπτωση του αλγορίθμου της Arifovic) ή αν αυτό αφορά μονάχα τον αριθμητικό μέσο των ποσοτήτων που επιλέγονται από τους διαφορετικούς παίκτες σε δεδομένη γενιά της προσομοίωσης, κάτι που ισοδυναμεί με την, διαπιστωμένη από τους προηγούμενους ερευνητές σύγκλιση της τιμής και της ποσότητας στην αγορά στις τιμές που προβλέπονται από το αντίστοιχο υπόδειγμα (Nash για τον αλγόριθμο του Vriend και των Alkemade et al και Walras για την Arifovic).

Αφού διαχωρίσουμε τις περιπτώσεις εκείνες στις οποίες η μικροοικονομική σύγκλιση δεν είναι εφικτή, θα προσπαθήσουμε να τροποποιήσουμε τους αλγόριθμους κατάλληλα, ώστε να είναι δυνατό κάτι τέτοιο. Εφόσον είναι εφικτή και η σύγκλιση των μεμονωμένων επιλογών των πρακτόρων, αφενός μπορούμε να χαρακτηρίσουμε τους τροποποιημένους αλγόριθμους μάθησης σαν πιο αποτελεσματικούς, αφετέρου η ανισοκατανομή του εισοδήματος που παρατηρείται στο σύστημα ελαχιστοποιείται (στην περίπτωση συμμετρικού παιγνίου, οπότε οι αναμενόμενες ποσότητες για όλους τους παίκτες γίνονται ίσες με την κοινή στρατηγική Nash και συνεπώς ίσες μεταξύ τους) και το σημαντικότερο, δημιουργείται η δυνατότητα χρησιμοποίησης των αλγορίθμων σαν ευρετικών αλγορίθμων για την εύρεση της ισορροπίας κατά Nash.

Όπως θα διαπιστώσουμε, η μετατροπή των αλγορίθμων πολλών πληθυσμών (Vriend 2000, Price 1997) σε κοινωνικής από ατομικής μάθησης, μας δίνει αυτήν δυνατότητα, εφόσον βέβαια οι συναρτήσεις κόστους είναι κοινές για όλες τις επιχειρήσεις, ώστε να έχει κατ΄ αρχήν νόημα η διαδικασία συλλογικής μάθησης (ένα χρωμόσωμα που χρησιμοποιείται από μια επιχείρηση να έχει την ίδια αποτελεσματικότητα, όταν χρησιμοποιείται από κάποια άλλη).

Στο τελευταίο κεφάλαιο θα προσπαθήσουμε να εισάγουμε έναν γενετικό αλγόριθμο για την επίλυση μιας γενικότερης κατηγορίας μη συμμετρικών παιγνίων Cournot. Θα αφήσουμε τους γενετικούς αλγόριθμους εξελικτικής μάθησης και θα διερευνήσουμε τη λύση στο πρόβλημα, μέσω της διαδοχικής εφαρμογής γενετικών αλγορίθμων βελτιστοποίησης, η οποία θα μας οδηγήσει στον προσδιορισμό της ισορροπίας κατά Nash ενός παιγνίου Cournot. Θα αξιοποιήσουμε δηλαδή την σύγκλιση της ακολουθίας των διαδοχικών βέλτιστων απαντήσεων των παικτών που αναλύσαμε στο κεφάλαιο 2, ώστε να κατασκευάσουμε έναν αλγόριθμο που θα βρίσκει την βέλτιστη απάντηση του κάθε παίκτη, με δεδομένες τις στρατηγικές των αντιπάλων του. Μάλιστα, με τη χρήση ενός εξωτερικού βρόχου, είναι εφικτή, όπως θα δούμε, η βελτίωση της ακρίβειας της υπολογιζόμενης λύσης. Βέβαια για να είναι εφικτά όλα αυτά, θα πρέπει, επαναλαμβάνουμε, η ακολουθία των θεωρητικών διαδοχικών βέλτιστων απαντήσεων να συγκλίνει στην ισορροπία κατά Nash, ή τουλάχιστον να μην αποκλίνει ξεκάθαρα (θα δούμε και μια τέτοια περίπτωση).

Αρχικά θα εξετάσουμε τις ιδιότητες σύγκλισης των 3 κυριότερων αλγορίθμων που παρουσιάστηκαν στο κεφάλαιο 3: του αλγόριθμου ατομικής μάθησης πολλών πληθυσμών της Arifovic (1994), του αλγόριθμου ατομικής μάθησης του Vriend (2000) με τυχαία επιλογή της τρέχουσας στρατηγικής (random selection στο Valee and Yildizoglou, 2007) και του αλγόριθμου κοινωνικής μάθησης των Alkemade et al (2007). Επίσης θα εξετάσουμε και τον αλγόριθμο με το όνομα «co-evolutionary programming»,έναν αλγόριθμο ατομικής επίσης μάθησης, αρκετά παρόμοιο με εκείνον του Vriend, ο οποίος χρησιμοποιήθηκε από τον Price (1997) στο πρόβλημα τιμολόγησης των αγορών ηλεκτρικής ενέργειας.

Θα εφαρμόσουμε τους 4 αυτούς αλγόριθμους σε παίγνια Cournot που ικανοποιούν τις συνθήκες του Amir (1996), έχουν δηλαδή φθίνουσα και οιωνεί κοίλη αντίστροφη συνάρτηση ζήτησης και αύξουσα συνάρτηση κόστους. Θα επιλέξουμε το γραμμικό μοντέλο των Alkemade et al (2007), ένα πολυωνυμικό μοντέλο που αποτελεί συγκεκριμένη εκδοχή του μοντέλου του Vriend(2000), και σε ένα μοντέλο με ρητό εκθέτη που δεν έχει χρησιμοποιηθεί προηγουμένως. Η αντίστροφη συνάρτηση ζήτησης

$$P = AQ^{\frac{3}{2}} + B, \quad A < 0$$

είναι φθίνουσα και κοίλη και η συνάρτηση κόστους

$$c = xq + y, \qquad x > 0$$

είναι αύξουσα, οπότε ικανοποιούνται οι συνθήκες του Amir (1996) για την ύπαρξη NE σε καθαρές στρατηγικές.

Θα διερευνήσουμε τις ιδιότητες σύγκλισης των αλγορίθμων στα 3 αυτά παίγνια με καθαρές ισορροπίες κατά Nash. Υπενθυμίζουμε ότι ακόμα και αν θεωρήσουμε ότι η μελέτη μας αφορά ένα διακριτό και όχι ένα διακριτοποιημένο παίγνιο Cournot, θα έχουμε την ίδια ισορροπία κατά Nash (εφόσον η αρχική βρίσκεται στο χώρο των τιμών των χρωμοσωμάτων), σύμφωνα με τους Dubey et al. (2006). Θα πρέπει τέλος να προσδιορίσουμε κάποια κριτήρια για να αξιολογήσουμε την επίδοση των αλγορίθμων. Παρόλο που μέχρι τώρα τα κριτήρια ήταν κυρίως η μέση ποσότητα (ή τιμή) και η διακύμανσή της, μέσα στον χρονικό ορίζοντα των προσομοιώσεων (Vriend 2000, Arifovic and Maaschek 2006, Valee and Yildizoglou 2007), κρίνουμε ότι αυτό δεν είναι επαρκές. Οι λόγοι είναι δύο. Όπως είπαμε στο κεφάλαιο 2, όλοι οι γενετικοί αλγόριθμοι με τελευταίο εφαρμοζόμενο τελεστή αυτόν της μετάλλαξης, μπορούν να περιγραφούν με κανονικές αλυσίδες Markov. Αυτό ισχύει και για τους αλγόριθμους που θα μελετήσουμε εδώ. Από τη στιγμή όμως που η στοχαστική διαδικασία που περιγράφει την εξέλιξη των αλγορίθμων μας είναι αλυσίδα Markov, αυτό αυτόματα σημαίνει ότι οι παρατηρήσεις μας δεν είναι ανεξάρτητες, οπότε δεν μπορούμε να συνάγουμε στατιστικά συμπεράσματα με μόνα κριτήρια το μέσο και τη διακύμανση. Αλλά και το -πραγματικά τεράστιο- πλήθος των διακριτών καταστάσεων, καθιστά τεχνικά ανέφικτη την εκτίμηση, τόσο των πιθανοτήτων μετάβασης, όσο απλά και του διανύσματος σταθερών συχνοτήτων για όλες τις καταστάσεις (βλ. Basawa and Rao, 1980, για τις εκτιμήτριες μέγιστης πιθανοφάνειας αυτών των παραμέτρων). Ωστόσο (Kemeny and Snell, 1960) μπορούμε να εκτιμήσουμε το σταθερό διάνυσμα συχνοτήτων για μεμονωμένες καταστάσεις, ή και για σύνολα που αποτελούνται από ένα μεγάλο αριθμό καταστάσεων, απλά αθροίζοντας τις σχετικές τους συχνότητες μέσα στην προσομοίωση, ή για ακόμα καλύτερα αποτελέσματα, αθροίζοντας τις αφού έχει προχωρήσει κάπως η προσομοίωση, ώστε να βρισκόμαστε κατά το δυνατόν στην ισορροπία (βλ. και Basawa and Rao, 1980).

Τα παραπάνω μας επιτρέπουν, με τον κατάλληλο ορισμό αυτών των «συγκεντρωμένων» ή «ομαδικών» καταστάσεων (lumped states) να εκτιμήσουμε όχι μόνο συνολικές παραμέτρους, όπως η μέση τιμή και ποσότητα, αλλά και τις πιθανότητες να βρεθεί ένας ή περισσότεροι παίκτες σε μια κατάσταση αρκετά κοντά στην επιθυμητή (ποσότητα Nash ή Walras).

5.2. Οι αλγόριθμοι

Ο αλγόριθμος ατομικής μάθησης της Arifovic, που παρουσιάσαμε και στην ενότητα 3 χαρακτηρίζεται από το γεγονός ότι κάθε επιχείρηση έχει το δικό της πληθυσμό χρωμοσωμάτων και τον δικό της γενετικό αλγόριθμο για την εξέλιξη του πληθυσμού αυτού, ο οποίος μεταβάλλει τα χρωμοσώματα σε κάθε γύρο της προσομοίωσης, επηρρεαζόμενος από την τιμή που επικρατεί στην αγορά. Ο αλγόριθμος έχει ως εξής (Yildizoglou and Vallee, 2007):

- Αρχικά προσδιορίζονται οι i πληθυσμοί , ένας για κάθε παίκτη, που αποτελούνται από j χρωμοσώματα, που αντιστοιχούν στις ποσότητες q_{ij} .
- Κάθε παίκτης i επιλέγει τυχαία ένα από τα χρωμοσώματα του πληθυσμού του. Έτσι προσδιορίζεται η τιμή στην αγορά από την $P = a - b\sum_{i=1}^{n} q_{ij}$.
- Εκτελείται ο ακόλουθος βρόχος για δεδομένο αριθμό βημάτων (διάρκεια προσομοίωσης)
    - Το υποθετικό κέρδος του κάθε χρωμοσώματος υπολογίζεται με βάση την τιμή στην αγορά και την εξίσωση κέρδους $\pi_{ij} = pq_{ij} - c(q_{ij})$.
    - Τα χρωμοσώματα του πληθυσμού αναπαράγονται με τη χρήση των τελεστών crossover και mutation κατά τα πρότυπα του «κανονικού γενετικού αλγορίθμου» (βλ. κεφάλαια 2-3) και με προσαρμοστικότητα ανάλογη με τα υποθετικά κέρδη.
    - Οι προσαρμοστικότητες των χρωμοσωμάτων - απογόνων υπολογίζονται όπως παραπάνω και κάθε παίκτης επιλέγει τυχαία, με πιθανότητα ανάλογη με την προσαρμοστικότητα, ένα χρωμόσωμα σαν την ποσότητα που προσφέρει στον επόμενο γύρο.
    - Με βάση τις παραπάνω ποσότητες, επανυπολογίζεται η νέα τιμή στην αγορά.

Η μοναδική διαφορά που θα έχει ο αλγόριθμός που θα χρησιμοποιήσουμε εδώ, από αυτόν της Arifovic, είναι η διατεταγμένη προσαρμοστικότητα - ordered fitness function - (δηλαδή το χρωμόσωμα με το χαμηλότερο κέρδος θα έχει προσαρμοστικότητα ίση με 1, εκείνο με το αμέσως χαμηλότερο 2, κ.ο.κ.). Η επιλογή μιας τέτοιας συνάρτησης προσαρμοστικότητας είναι αρκετά διαδεδομένη και μας δίνει την δυνατότητα να έχουμε έναν νεωτερισμό, χωρίς να μεταβάλλεται η ουσία του προβλήματος. Την ίδια τακτική θα ακολουθήσουμε και για τους υπόλοιπους αλγορίθμους. Ο κώδικας Matlab που υλοποιήσαμε για τον αλγόριθμο της Arifovic (όπως εφαρμόζεται στο γραμμικό υπόδειγμα) βρίσκεται στο παράρτημα Α-1. Με μικρές αλλαγές στις συναρτήσεις ζήτησης και κόστους προκύπτουν και οι υλοποιήσεις που χρησιμοποιήσαμε για την εφαρμογή του αλγορίθμου στα υπόλοιπα υποδείγματα.

Ο αλγόριθμος ατομικής μάθησης του Vriend, όπως παρουσιάζεται από τους Yildizoglou and Vallee (2007) έχει ως εξής

- Επιλέγονται με τυχαίο τρόπο τα χρωμοσώματα-στρατηγικές q_{ij} των πληθυσμών των i επιχειρήσεων.
- Για δεδομένο αριθμό περιόδων εκτέλεσε τον βρόχο

- Αν η τρέχουσα περίοδος είναι ακέραιο πολλαπλάσιο της παραμέτρου GArate, μετέβαλλε τους πληθυσμούς με εφαρμογή των τελεστών του γενετικού αλγορίθμου (διασταύρωσης, μετάλλαξης κλπ.)
- Αν η τρέχουσα περίοδος δεν είναι ακέραιο πολλαπλάσιο της GArate, επέλεξε με κάποιο τρόπο κάποια στρατηγική από τον πληθυσμό της κάθε επιχείρησης, σαν την τρέχουσα στρατηγική της q_{ij}^{*}. Το κέρδος της στρατηγικής προσδιορίζεται από το αποτέλεσμα του παιγνίου Cournot μεταξύ των τρεχουσών στρατηγικών όλων των επιχειρήσεων $\pi_{i}\left(q_{ij}^{*t}\right)=\pi_{i}\left(q_{ij}^{*},Q_{-i}^{*}\right)$. Δηλαδή οι τρέχουσες στρατηγικές για όλες τις επιχειρήσεις χρησιμοποιούνται για να προσδιοριστεί η τρέχουσα τιμή, η οποία οδηγεί στον υπολογισμό του κέρδους της κάθε στρατηγικής ξεχωριστά.

Οι διαφορές λοιπόν μεταξύ των αλγορίθμων ατομικής μάθησης της Arifovic (1994) και του Vriend (2000), είναι

a) Στον αλγόριθμο του Vriend, το κέρδος κάθε στρατηγικής προσδιορίζεται μονάχα όταν χρησιμοποιείται η στρατηγική, σε αντίθεση με το μοντέλο της Arifovic, που χρησιμοποιούνται υποθετικά κέρδη, για τον προσδιορισμό της προσαρμοστικότητας όλων των στρατηγικών των πληθυσμών, σε κάθε περίοδο της προσομοίωσης

b) Στον αλγόριθμο του Vriend, οι πληθυσμοί μεταβάλλονται κάθε GArate περιόδους, ενώ στον αλγόριθμο της Arifovic, σε κάθε περίοδο.

Όσον αφορά τώρα, τον τρόπο επιλογής της τρέχουσας στρατηγικής από τον πληθυσμό στρατηγικών της κάθε επιχείρησης, οι Yildizoglou and Vallee (2007), έδειξαν, όπως αναφέραμε και στο κεφάλαιο 3, ότι τόσο η μέθοδος επιλογής με πιθανότητα ανάλογη με το τρέχον κέρδος, όσο και μια εντελώς τυχαία επιλογή είναι ελαφρώς καλύτερες, όσον αφορά την εξελικτική σύγκλιση του αλγορίθμου στην ισορροπία κατά Nash, από την επιλογή της καλύτερης (best of) στρατηγικής, εκείνης δηλαδή με το μέγιστο τρέχον κέρδος, που επίσης λειτουργεί ικανοποιητικά.

Εμείς αποφασίσαμε να υιοθετήσουμε την τυχαία επιλογή, με ίσες πιθανότητες για όλα τα χρωμοσώματα του πληθυσμού. Επίσης, όπως και σε όλους τους αλγόριθμους, χρησιμοποιούμε διατεταγμένη προσαρμοστικότητα (ordered fitness). Ο κώδικας Matlab βρίσκεται στο παράρτημα Α-2.

Ο αλγόριθμος του coevolutionary programming, που εφαρμόστηκε από τον Price(1997) στο πρόβλημα τιμολόγησης αγορών ηλεκτρικής ενέργειας, έχει πολλές ομοιότητες με τον αλγόριθμο του Vriend. Όπως περιγράφεται από τους Son and Baldick (2004), ο αλγόριθμος έχει ως εξής:

1. Επέλεξε τις παραμέτρους για τον αριθμό των επιχειρήσεων, τον αριθμό των χρωμοσωμάτων σε κάθε πληθυσμό, την πιθανότητα μετάλλαξης κλπ.

2. Αρχικοποίησε με τυχαίο τρόπο τους i πληθυσμούς (ένας για κάθε επιχείρηση, πάντα) με τυχαίο τρόπο.
3. Επέλεξε τυχαία στρατηγική για κάθε παίκτη, και εκτέλεσε το παίγνιο μεταξύ των επιλεγμένων στρατηγικών. Το κέρδος της κάθε στρατηγικής είναι το κέρδος του παίκτη που την χρησιμοποίησε, στο παίγνιο.
4. Επανέλαβε το βήμα (3) μέχρι να αποδοθεί κέρδος σε όλες τις στρατηγικές.
5. Μετασχημάτισε τους πληθυσμούς, εφαρμόζοντας τους τελεστές του γενετικού αλγόριθμου.
6. Επανέλαβε τον βρόχο (3-5) για καθορισμένο αριθμό γενεών.

Βλέπουμε ότι οι αλγόριθμοι είναι σχεδόν ίδιοι. Η μόνη τους διαφορά είναι, ότι ενώ ο Vriend χρησιμοποιεί την σταθερά GArate για να προσδιορίσει το χρονικό σημείο στο οποίο θα αλλάξουν οι πληθυσμοί, ο Price χρησιμοποεί απλά τη συνθήκη, ότι θα πρέπει όλα τα χρωμοσώματα να έχουν «συμμετάσχει» σε κάποιο παίγνιο. Η υλοποίηση που θα κάνουμε (παράρτημα Α-3 για το υπόδειγμα με ρητό εκθέτη) εκτελεί τον αλγόριθμο αυτόν ιδιαίτερα οικονομικά (από άποψη χρόνου) αφού χρησιμοποιούμε την δυνατότητα του Matlab να μεταθέτει στοιχεία διανυσμάτων, με τυχαίο τρόπο. Έτσι χρησιμοποιούμε τυχαίες μετάθεσεις των αριθμών $1,\ldots,Population\_Size$, μια για κάθε παίκτη και απλά, αντιστοιχίζουμε τις $i-$ στές στρατηγικές, στα διανύσματα των τυχαίων μεταθέσεων για να πάρουμε ένα μοναδικό, τυχαίο «τουρνουά» για όλες τις στρατηγικές όλων των παικτών.

Τέλος ο αλγόριθμος των Alkemade et al (2007), παρουσιάστηκε στο κεφάλαιο 3. Ο κώδικας Matlab που υλοποιήσαμε βρίσκεται στο παράρτημα Α-4.

5.3. Τα υποδείγματα

Το υπόδειγμα που χρησιμοποιούν οι Alkemade et al (2007) αποτελεί μια ειδική, απλή περίπτωση των υποδειγμάτων των Arifovic και Vriend. Στο υπόδειγμα αυτό, που παρουσιάσαμε, όπως και τα προηγούμενα, στην ενότητα 3, οι συναρτήσεις κόστους και αντίστροφης ζήτησης δίνονται από τις

$$P = a - Q, \qquad \text{με } Q = \sum_{i=1}^{n} q_i$$
$$c(q_i) = xq_i$$

για n επιχειρήσεις. Το υπόδειγμα προκύπτει απ΄ το υπόδειγμα της Arifovic για $y = 0, b = 1$. Οι Alkemade et al. (2007) θέτουν $a = 256$ και $x = 56$. Η συνάρτηση αντίστροφης ζήτησης είναι φθίνουσα και κοίλη και η συνάρτηση κόστους γραμμική και αύξουσα. Ικανοποιούνται έτσι οι συνθήκες του Amir (1996), οπότε υπάρχει ισορροπία Nash σε καθαρές στρατηγικές.

Οι ισορροπίες κατά Nash και Walras προκύπτουν εύκολα. Για την Βαλρασιανή ισορροπία έχουμε (θεωρούμε P σταθερή)

$$\frac{d\pi}{dq} = \frac{d(Pq - c)}{dq} = P - \frac{dc}{dq} = a - nq - x$$

λόγω συμμετρίας του παιγνίου Cournot, $Q = nq$ στην ισορροπία. Πρέπει

$$\frac{d\pi}{dq} = 0 \Leftrightarrow q^W = \frac{a - x}{n}$$

και

$$\left.\frac{d\pi}{dq}\right|_{q = q^W} = -n < 0$$

που ισχύει $\forall q$. Για $a = 256$, $x = 56$ και $n = 4$ είναι $q^w = 50$ και για $n = 20$ $q^w = 10$.

Για την ισορροπία κατά Nash έχουμε:

$$\frac{d\pi_i}{dq_i} = \frac{d(Pq_i - c(q_i))}{dq_i} = \frac{dP}{dq_i}q_i + P - \frac{dc(q_i)}{dq_i} = -q_i + a - nq_i - x$$

και

$$\frac{d\pi_i}{dq_i} = 0 \Leftrightarrow q^N = \frac{a - x}{n + 1}$$

ενώ

$$\left.\frac{d\pi_i}{dq_i}\right|_{q_i = q^N} = -n - 1 < 0$$

που ισχύει πάντα. Αν $a = 256$, $x = 56$ και $n = 4$ είναι $q^N = 40$. Για $n = 20$, $q^N = 9.5238$

Το πολυωνυμικό μοντέλο θα έχει συνάρτηση ζήτησης φθίνουσα και κοίλη και αύξουσα συνάρτηση κόστους και οι δύο πολυωνυμικές. Ένα πολυώνυμο 3ου βαθμού

$$P = AQ^3 + B$$

με A<0, έχει τις επιθυμητές ιδιότητες για την αντίστροφη συνάρτηση ζήτησης, ενώ θα επιλέξουμε γραμμική συνάρτηση κόστους

$$c = xq + y$$

με $x > 0$. Η ισορροπία κατά Walras δίνεται από την

$P - \frac{dc}{dq} = 0 \Leftrightarrow AQ^3 + B - x = 0 \Leftrightarrow An^3q^3 + B - x = 0$ (στην ισορροπία $Q = nq$) και τελικά

$$q^W = \frac{1}{n}\left(\frac{x - B}{A}\right)^{\frac{1}{3}}$$

με $\frac{d^2}{dq^2}\left(An^3q^3 + B - x\right) = 2An^3q^2 < 0$ για $A < 0$ και άρα έχουμε μέγιστο.

Η ισορροπία κατά Nash δίνεται από την

$$\frac{d\pi}{dq}=0 \Leftrightarrow \frac{dP}{dQ}\frac{dQ}{dq}q+P-\frac{dc}{dq}=0 \Leftrightarrow 3AQ^2q+AQ^3+B-x=0$$ και στην ισορροπία

$\left(3an^2+An^3\right)q^3+B-x=0$ οπότε $q^N=\left(\frac{x-B}{3An^2+An^3}\right)^{\frac{1}{3}}$.

Επιλέγουμε $A=-1$ και $q^N=20$ για $n=20$ επιχειρήσεις, οπότε $B-x=7{,}36\times 10^7$ και $q^w=20.9538$, ενώ για $n=4$ επιχειρήσεις $q^w=104.7690$ και $q^N=86.9401$. Τέλος θέτουμε $x=10$, $y=10$, οπότε και $B=7{,}36\cdot 10^7+10$.

Στο μοντέλο με ρητό εκθέτη,

$$\begin{aligned} P&=AQ^{\frac{3}{2}}+B \quad & A<0 \text{ και}\\ c&=xq+y & x>0 \end{aligned}$$

με τα A,x να έχουν οριστεί κατ΄ αυτόν τον τρόπο για να έχουμε φθίνουσα και κοίλη αντίστροφη συνάρτηση ζήτησης και αύξουσα συνάρτηση κόστους, η ισορροπία κατά Walras δίνεται από την

$P-\frac{dc}{dq}=0 \Leftrightarrow AQ^{\frac{3}{2}}+B-x=0 \Leftrightarrow Q=\left(\frac{x-B}{A}\right)^{\frac{2}{3}}$ και στην ισορροπία $q^w=\frac{1}{n}\left(\frac{x-B}{A}\right)^{\frac{2}{3}}$. Επίσης στο

q^w, $\frac{d}{dq}\left(P-\frac{dc}{dq}\right)=\frac{3An^{\frac{3}{2}}}{2}\sqrt{q}<0$, αφού $A<0$, οπότε έχουμε μέγιστο.

Η ισορροπία κατά Nash δίνεται από

$$\frac{d\pi}{dq}=0 \Leftrightarrow \frac{dP}{dQ}\frac{dQ}{dq}q+P-\frac{dc}{dq}=0 \Leftrightarrow \frac{3A}{2}q\sqrt{Q}+AQ^{\frac{3}{2}}+B-x=0$$ και στην ισορροπία

$$\frac{3A\sqrt{n}}{2}q^{\frac{3}{2}}+An^{\frac{3}{2}}q^{\frac{3}{2}}=x-B \Leftrightarrow q^N=\left[\frac{2(x-B)}{3An^{\frac{1}{2}}+2An^{\frac{3}{2}}}\right]^{\frac{2}{3}}.$$

Θέτουμε $A=-1$ και $B-x=8200$ οπότε για $n=20$, $q^N=19.3749$ και $q^W=20.332$. Για $n=4$ είναι $q^N=82.2143$ και $q^W=101.6598$. Τέλος θέτουμε $x=100$, οπότε $B=8300$.

5.4. Τα κριτήρια αξιολόγησης

Όπως είπαμε και παραπάνω, τα βασικά κριτήρια που χρησιμοποιούνται στην αξιολόγηση της διαδικασίας σύγκλισης των αλγορίθμων είναι η μέση τιμή και ποσότητα και οι διακυμάνσεις γύρω από αυτές, στη διάρκεια της προσομοίωσης. Ωστόσο, όπως είδαμε και στην ενότητα 2, οι στοχαστικές διαδικασίες που χαρακτηρίζουν την εξέλιξη των γενετικών αυτών αλγορίθμων, είναι κανονικές αλυσίδες Markov (τουλάχιστον ο αλγόριθμος της Arifovic, βλ. αμέσως παρακάτω τι συμβαίνει για τους υπόλοιπους αλγορίθμους), οπότε οι παρατηρήσεις στις διάφορες χρονικές στιγμές δεν είναι ανεξάρτητες και συνεπώς η μέση τιμή και η διακύμανση δεν προσφέρεται για την εξαγωγή συμπερασμάτων. Αντίθετα αυτό που μπορεί να γίνει στις (κανονικές) διαδικασίες Markov, είναι η εκτίμηση τόσο των πιθανοτήτων μετάβασης, όσο και των συνιστωσών του διανύσματος σταθερών συχνοτήτων. Από εκεί μπορούμε να εκτιμήσουμε και τις υπόλοιπες παραμέτρους της αλυσίδας, όπως οι μέσοι χρόνοι μετάβασης κλπ.

Ένα δεύτερο πρόβλημα προκύπτει σχετικά με την ομογένεια της αλυσίδας Markov, των αλγορίθμων του Vriend, Alkemade et al και co-evolutionary programming. Δεδομένου ότι οι πληθυσμοί των γενετικών αλγορίθμων μεταβάλλονται σε συγκεκριμένες περιόδους, οι πιθανότητες μετάβασης στις συγκεκριμένες αυτές περιόδους, είναι εκείνες που διαμορφώνουν την αλυσίδα Markov του αλγορίθμου, για τις οποίες ιχύουν όσα είπαμε στο κεφαλαίου 2 για τις πιθανότητες μετάβασης που προκύπτουν για τους τελεστές των κανονικών γενετικών αλγορίθμων, ενώ η διαδικασία στις επόμενες περιόδους είναι σταθερή -ο πίνακας μετάβασης είναι ο μοναδιαίος. Αυτό λοιπόν που πραγματικά συμβαίνει, μεταξύ δύο διαδοχικών καταστάσεων των πληθυσμών (μεταξύ των περιόδων $kGArate$ και $(k+1)GArate$) είναι ότι, σε όλες τις ενδιάμεσες καταστάσεις, συγκεκριμενοποιείται η τυχαία μεταβλητή που δίνει την προσαρμοστικότητα του κάθε χρωμοσώματος, μέσω των τυχαίων «συνταιριασμάτων» των στρατηγικών σε παίγνια Cournot (για την ακριβή εξίσωση της προσαρμοστικότητας, βλ. Yildizoglou and Vallee, 2007). Με βάση τις κατ΄ αρχήν πιθανότητες της τυχαίας αυτής μεταβλητής, υπολογίζονται και οι πιθανότητες μετάβασης από την κατάσταση k στην $k+1$. Δεδομένου τώρα, ότι οι πιθανότητες και οι πιθανές τιμές της προσαρμοστικότητας εξαρτώνται αποκλειστικά, από τα χρωμοσώματα που βρίσκονται στους πληθυσμούς των επιχειρήσεων, στην περίοδο $kGArate$, η στοχαστική διαδικασία που περιγράφει τις καταστάσεις των πληθυσμών στις περιόδους $GArate, 2GArate, ..., kGArate, ...$ είναι διαδικασία Markov και μάλιστα ομογενής. Λαμβάνοντας υπόψη αυτά που είπαμε στο κεφάλαιο 2 για τους κανονικούς γενετικούς αλγορίθμους και με δεδομένο ότι μια τυχαία προσαρμοστικότητα δεν αλλάζει σε τίποτε την ανάλυση του Riechmann (1999), διαπιστώνουμε ότι, περιορίζοντας τις καταστάσεις της διαδικασίας των αλγορίθμων στις περιόδους στις οποίες μεταβάλλονται οι πληθυσμοί, έχουμε μια κανονική αλυσίδα Markov, όπως στον αλγόριθμο της Arifovic (1994).

Αν έχουμε μια κανονική αλυσίδα Markov με χώρο καταστάσεων S, πιθανότητες μετάβασης

$$P_{ij} = P[X_k = j | X_{k-1} = i] \qquad i, j \in S$$

και αρχικές πιθανότητες

$$p_u^{(0)} = P[X_0 = u] \qquad u \in S$$

και έχουμε μια πραγμάτωση της στοχαστικής διαδικασίας

$$X_1, X_2, ..., X_n$$

οι εκτιμήτριες μέγιστης πιθανοφάνειας των P_{ij} είναι ανεξάρτητες των αρχικών πιθανοτήτων και δίνονται από τις (Basawa and Rao, 1980)

$$\hat{P}_{ij} = \frac{n_{ij}}{n}$$

τον λόγο δηλαδή του αριθμού των μεταβάσεων από την κατάσταση i στην κατάσταση j, ως προς το μήκος n της πραγμάτωσης. Παρόμοια οι εκτιμήτριες των συνιστωσών του διανύσματων σταθερών συχνοτήτων

$$\boldsymbol{\pi} = (\pi_1, ..., \pi_m) \qquad \text{με } m = |S|$$

δίνονται από τις σχετικές συχνότητες

$$\frac{n_i}{n}$$

Είναι επίσης θεμιτό, όπως σημειώνουν οι Basawa και Rao (σελ.54), να πάρουμε τις παρατηρήσεις από ένα σημείο k και μετά, οπότε η αλυσίδα θα έχει συγκλίνει επαρκώς στην ισορροπία, οπότε θα έχουμε μεγαλύτερη ακρίβεια (θα αποφύγουμε τον αρχικό «θόρυβο» που παρατηρείται πριν η αλυσίδα συγκλίνει επαρκώς στην ισορροπία- ο οποίος ούτως ή άλλως δεν είναι σημαντικός, εφόσον έχουμε μεγάλο αριθμό παρατηρήσεων).

Δυστυχώς στην περίπτωσή μας, ο αριθμός των καταστάσεων είναι πολύ μεγάλος, κάτι που σημαίνει ότι και ο αριθμός των παρατηρήσεων που χρειαζόμαστε για μια ικανοποιητική εκτίμηση, ακόμα και του $\boldsymbol{\pi}$ είναι ανάλογα υψηλός. Για καθέναν από τους σύνθετους γενετικούς μας αλγόριθμους με n πληθυσμούς m χρωμοσωμάτων μήκους l, το πλήθος των διαφορετικών καταστάσεων (όπως προσδιορίζεται από τα χρωμοσώματα του καθενός από τους πληθυσμούς) είναι 2^{nml}. Άρα ακόμα και στην περίπτωση που έχουμε έναν πληθυσμό (όπως στους αλγόριθμους κοινωνικής μάθησης) με έναν ελάχιστο αριθμό 20 χρωμοσωμάτων (που θεωρείται ανεπαρκής για την ικανοποιητική σύγκλιση ενός γενετικού αλγόριθμου) τότε το πλήθος των δυνατών καταστάσεων θα είναι μεγαλύτερο από 2^{20} κατά κάποιες τάξης μεγέθους (τόσες όσο το μήκος του κάθε χρωμοσώματος, που συνεπάγεται τον αριθμό των εφικτών τιμών που θέλουμε να διερευνήσουμε). Προφανώς δεν είναι εφικτή η εκτίμηση όλων των πιθανοτήτων μετάβασης ή και των συχνοτήτων στην ισορροπία ακόμα.

Αυτό που μπορούμε όμως να κάνουμε είναι να εξετάσουμε σύνολα καταστάσεων ή ομαδικές καταστάσεις (lumped states). Μπορούμε δηλαδή να διαμερίσουμε το χώρο των δυνατών καταστάσεων σε κλάσεις ισοδυναμίας που θα αποτελούνται από περισσότερες από μια καταστάσεις και να μελετήσουμε τη συμπεριφορά του αλγορίθμου, όσον αφορά τις ομαδικές αυτές καταστάσεις. Παρόλο που, εν γένει, ένας διαμερισμός σε ομαδικές καταστάσεις οδηγεί σε μια στοχαστική διαδικασία που έχει απωλέσει την ιδιότητα Markov (Basawa and Rao 1980), μπορούμε ακόμα να εκτιμήσουμε τις συνιστώσες του διανύσματος σταθερών συχνοτήτων και τις πιθανότητες μετάβασης 1ης τάξης (εφόσον ξεκινήσουμε αρκετά κοντά στην ισορροπία), χωρίς να χρειαστεί να ανατρέξουμε στην αρχική αλυσίδα Markov : « Εφόσον η διαδικασία Markov είναι κανονική και οι συνθήκες για διατήρηση της Μαρκοβιανής ιδιότητας (lumpability) δεν ισχύουν για τον διαμερισμό

$$A_1, A_2, ..., A_n \text{ με } \bigcup A_i = S \text{ και } A_i \cap A_j = \emptyset \quad \forall\, i, j$$

και ξεκινήσουμε τη διαδικασία από την κατάσταση ισορροπίας, τότε η

$$\hat{p}_{ij} = \Pr_a\left[X_{n+1} \in A_j \middle| X_n \in A_i\right]$$

είναι η ίδια για όλα τα n. Οπότε η μήτρα $\hat{P} = \hat{p}_{ij}$ μπορεί να θεωρηθεί σαν η μήτρα μετάβασης ενός βήματος. Ομοίως το διάνυσμα

$$\hat{a}_i = \Pr_a\left[X_n \in A_i\right]$$

Είναι σταθερό για όλα τα n και οι συνιστώσες του δεν είναι παρά το άθροισμα των συνιστωσών του διανύσματος σταθερών συχνοτήτων για όλες τις καταστάσεις που ανήκουν στο A_i (οι εκτιμήτριες της μέγιστης πιθανοφάνειας του διανύσματος σταθερών συχνοτήτων είναι ανεξάρτητες από την αρχική κατάσταση, χωρίς αυτό να σημαίνει ότι είναι ανώφελο να ξεκινήσουμε από μια προχωρημένη χρονική στιγμή του αλγορίθμου, που θα μας φέρει πιο κοντά στην κατάσταση ισορροπίας, όπως προαναφέραμε). Παρόμοια μπορούμε να ορίσουμε τη μήτρα των πιθανοτήτων μετάβασης δεύτερης τάξης. Δεν ισχύει

$$P_2 = P^2$$

αν η διαδικασία των ομαδικών καταστάσεων δεν είναι αλυσίδα Markov. Μπορούμε επίσης να ορίσουμε τον αναμενόμενο χρόνο πρώτης μετάβασης $M = \{m_{ij}\}$ στην ομαδική διαδικασία, αλλά για να τον βρούμε θα πρέπει να αντρέξουμε στην αρχική αλυσίδα Markov και να υπολογίσουμε τον χρόνο μετάβασης από κάθε κατάσταση $i \in A_i$ στην ομαδική κατάσταση A_j, m_{iA_j}. Για να το πετύχουμε αυτό, μπορούμε να μετατρέψουμε τις καταστάσεις της A_j σε καταστάσεις απορρόφησης (absorbing states) και να εκτιμήσουμε το μέσο χρόνο απορόφησης. Θα είναι

$$\hat{m}_{ij} = \sum_{k \in A_i} a_k^* m_{k,A_j}$$

με a_k^* την k-οστή συνιστώσα του a^i (Kemeny and Snell 1960).

Βλέπουμε λοιπόν ότι μπορούμε να εκτιμήσουμε τις συνιστώσες του διανύσματος σταθερής κατάστασης για την διαδικασία των ομαδοποιημένων καταστάσεων, απλά αθροίζοντας τις εκτιμήτριες για τις συχνότητες όλων των μεμονωμένων καταστάσεων που αποτελούν την ομαδοποιημένη κατάσταση. Παρόμοια μπορούμε να εκτιμήσουμε τις πιθανότητες μετάβασης ενός βήματος. Για την εκτίμηση των πιθανοτήτων μετάβασης, θα πρέπει να ξεκινήσουμε σε μια κατάσταση ισορροπίας (σε μια κατάσταση δηλαδή, για την οποία $P^n \approx A$ - με A την μήτρα σταθερής κατάστασης). Εφόσον πάντως ο αριθμός παρατηρήσεων είναι αρκετά υψηλός, η παραπάνω συνθήκη δεν είναι αναγκαία στην πράξη.

Όσον αφορά τώρα τον υπολογισμό των αναμενόμενων χρόνων πρώτης μετάβασης, από δεδομένη αρχική κατάσταση σε δεδομένη τελική κατάσταση της αλυσίδας Markov, θα πρέπει να γνωρίζουμε τον πίνακα P_{ij} (Kemeny and Snell 1960). Δεδομένου ότι αυτό δεν ισχύει στην περίπτωσή μας, αυτό που επίσης μπορούμε να κάνουμε είναι να μετατρέψουμε τις καταστάσεις της όποιας (απλής ή και ομαδοποιημένης ακόμα) κατάστασης A_j σε καταστάσεις απορρόφησης (absorbing states) και να εκτελέσουμε την διαδικασία N φορές, για δεδομένη αρχική κατάσταση. Δεδομένου ότι οι N εκτελέσεις θα είναι, εν γένει, ανεξάρτητες, μπορούμε κατ΄ αυτόν τον τρόπο να εκτιμήσουμε τον αναμενόμενο χρόνο πρώτης μετάβασης στην ομαδική κατάσταση A_j, από αρχική κατάσταση i (Kemeny and Snell 2006).

Αφού ο κύριος στόχος μας είναι να διερευνήσουμε την σύγκλιση των αλγορίθμων στην ισορροπία Nash (ή Walras στην περίπτωση του αλγορίθμου της Arifovic), λογικό είναι να ορίσουμε τις ομαδικές καταστάσεις μας κατά ένα τέτοιο τρόπο, ώστε να σχετίζονται με την απόσταση από την κατάσταση αυτή. Θα χρησιμοποιήσουμε την, αρκετά διαδεδομένη, έννοια της απόστασης Hamming (βλ. και κεφάλαιο 2) για να μετρήσουμε την απόσταση δύο χρωμοσωμάτων. Η απόσταση αυτή ορίζεται ως το πλήθος των bits που διαφέρουν στα δύο χρωμοσώματα. Π.χ. τα χρωμοσώματα

100110 και 100000

έχουν απόσταση Hamming 2.

Από τη στιγμή που έχουμε περισσότερους από έναν πληθυσμούς, με ένα -εξωγενώς καθορισμένο- αριθμό χρωμοσωμάτων ο καθένας, ο ορισμός των καταστάσεων, με βάση τη συνολική απόσταση όλων των χρωμοσωμάτων από το χρωμόσωμα- στόχο καθισταται προβληματικός, αφού εξαρτάται από τον αριθμό των επιχειρήσεων και των χρωμοσωμάτων του πληθυσμού της κάθε επιχείρησης (με εξαίρεση τον αλγόριθμο των Alkemade et al (2007), ο οποίος έχει έναν μόνο πληθυσμό). Πιο συγκεκριμένα, αν κάθε χρωμόσωμα έχει μήκος l, έχουμε m χρωμοσώματα στον πληθυσμό της κάθε επιχείρησης και n επιχειρήσεις, τότε η μέγιστη συνολική απόσταση (στην περίπτωση που όλα τα χρωμοσώματα είναι το ακριβώς αντίστροφο χρωμόσωμα από το χρωμόσωμα-στόχος) είναι

$$D_{\max} = lmn$$

Αυτό είναι και το πλήθος των διαφορετικών ομαδικών καταστάσεων, αν διαμερίσουμε τον συνολικό χώρο των δυνατών καταστάσεων των αλγορίθμων, με βάση την συνολική απόσταση. Αν δηλαδή θεωρήσουμε ότι όλες οι δυνατές καταστάσεις των πληθυσμών των επιχειρήσεων που απέχουν από το χρωμόσωμα-στόχος κατά D πρέπει να ανήκουν στο ίδιο σύνολο. Δεδομένου τώρα ότι ένας γενετικός αλγόριθμος χρειάζεται τουλάχιστον 20-30 χρωμοσώματα στον πληθυσμό του για να λειτουργήσει αποδοτικά (το πλήθος των χρωμοσωμάτων πρέπει να είναι μεταξύ του l και του $2l$, κατά τον Alander(1992)), διαπιστώνουμε ότι, το πλήθος των ομαδικών αυτών καταστάσεων είναι πολύ μεγάλο για να είναι πρακτικά εφικτή η εκτίμηση και του διανύσματος σταθερών συχνοτήτων ακόμα.

Έτσι λοιπόν θα προτιμήσουμε να ορίσουμε τις συνολικές καταστάσεις σύμφωνα με την μέση απόσταση. Δύο καταστάσεις του χώρου των πληθυσμών των αλγορίθμων θα ανήκουν στο ίδιο σύνολο A_i αν η μέση απόσταση των χρωμοσωμάτων τους από το χρωμόσωμα-στόχο κυμαίνεται μεταξύ του $i-1$ και του i. Συμβατικά ορίζουμε την κατάσταση A_0 σαν την ιδανική μοναδική κατάσταση εκείνη που όλα τα χρωμοσώματα, όλων των πληθυσμών των επιχειρήσεων είναι ίσα με το χρωμόσωμα-στόχο. Η μοναδική απλή κατάσταση στην ανάλυσή μας, λοιπόν, θα είναι εκείνη στην οποία όλα τα χρωμοσώματα, όλων των πληθυσμών, είναι η ποσότητα ισορροπίας (Nash ή Walras, ανάλογα με το υπόδειγμα). Μπορούμε να υπολογίσουμε και τον αναμενόμενο χρόνο μεταξύ δύο διαδοχικών επισκέψεων στην κατάσταση αυτή, πολύ απλά, αφού είναι η αντίστροφη, της αντίστοιχης συνιστώσας του διανύσματος σταθερών συχνοτήτων $\frac{1}{\pi_i}$ (Kemeny and Snell 1960). Έτσι λοιπόν μπορούμε να ελέγξουμε και την ευστάθεια των αλγορίθμων.

Δεν είναι δύσκολο να διαπιστώσουμε ότι το πλήθος των ομαδικών καταστάσεων είναι ακριβώς $l+1$. Από τη στιγμή που η μέγιστη απόσταση ενός χρωμοσώματος από το χρωμόσωμα-στόχο είναι (στην περίπτωση που όλα τα bits είναι διαφορετικά) l, αυτή είναι και η μέγιστη μέση απόσταση $d_{\max}$. Έτσι θα έχουμε τις ομαδικές καταστάσεις $A_0, A_1, ..., A_l$. Προφανές είναι επίσης ότι ο ομαδοποίηση αυτή αποτελεί διαμερισμό του χώρου των αρχικών καταστάσεων.

5.5. Προσομοιώσεις.

Θα φροντίσουμε πάντα ώστε η μέγιστη ποσότητα που μπορεί να απεικονίσει το χρωμόσωμα (η ποσότητα που αντιστοιχεί στην τιμή του χρωμοσώματος 2^l-1) να είναι η τριπλάσια από την ποσότητα στόχο (ισορροπία κατά Nash ή Walrash). Ο λόγος που το κάνουμε αυτό δεν είναι μόνο η «άνεση χώρου», αλλά η πολύ σημαντική προϋπόθεση, για τις αναλύσεις μας των ομαδικών καταστάσεων, η ποσότητα - στόχος να βρίσκεται μέσα στο σύνολο των τιμών που μπορεί να πάρει ένα χρωμόσωμα. Δεν αρκεί, στην περίπτωσή μας, να έχουμε μια προσέγγιση της τιμής ισορροπίας στον πληθυσμό μας. Όλοι οι ορισμοί των ομαδικών καταστάσεων και οι μετρήσεις των αποστάσεων, αλλά και οι εκτιμήσεις των συχνοτήτων, μέσων χρόνων μετάβασης κ.ο.κ., τόσο για τις ομαδικές καταστάσεις, όσο και για την επιθυμητή κατάσταση καθεαυτή, θέτουν την απαίτηση ύπαρξης συγκεκριμένου χρωμοσώματος με τη συγκεκριμένη αυτή τιμή. Αν τώρα θέσουμε

$$q_{\max} = 2q^*$$

τότε, απ΄ τη στιγμή που η ποσότητα στην οποία αντιστοιχεί ένα χρωμόσωμα x δίνεται από την

$$q = \frac{q_{\max} \cdot x}{2^l - 1}$$

θα έχουμε

$$q = \frac{q_{\max}}{2} \Rightarrow x = \frac{2^l - 1}{2}$$

και, δεδομένου ότι ο $2^l - 1$ είναι περιττός, κάτι τέτοιο δεν μπορεί να συμβαίνει.

Το χρωμόσωμα $2^l - 1$ έχει δυαδική τιμή

$$2^l - 1 = \underbrace{11...1}_{l}(bin)$$

Ομοίως

$$q = \frac{q_{\max}}{3} \Rightarrow x = \frac{2^l - 1}{3}$$

Ο αριθμός 3 είναι $11(bin)$. Η ακέραια διαίρεση δύο δυαδικών αριθμών γίνεται όπως και στην περίπτωση των δεκαδικών (Socha and Norton 1994). Αν εκτελέσουμε την διαίρεση $1101:11$ για παράδειγμα, έχουμε πηλίκο 100 και υπόλοιπο 1 (δυαδικά πάντα). Παρατηρούμε λοιπόν, ότι η διαίρεση

$$\underbrace{11...1}_{l}:11$$

θα είναι ακριβής, αν και μόνο αν το l είναι άρτιος, οπότε το πηλίκο θα αποτελείτα από l «01» bit-strings

$$\underbrace{11...1}_{l}:11 = \underbrace{0101...01}_{l}$$

Εφόσον έχουμε για παράδειγμα $l = 20$, η τιμή-στόχος (η ισορροπία κατά Nash ή Walrash) θα αντιστοιχεί στο 20-μπιτο χρωμόσωμα με μηδενική τιμή στα άρτια bits και τιμή 1 στα περιττής τάξης bits.

Μπορούμε και να αποδείξουμε την εξίσωση

$$\underbrace{11...1}_{l}:11 = \underbrace{0101...01}_{l}$$

Πολύ εύκολα, με επαγωγή: για $l = 2$ προφανώς

$$11:11 = 1 = 01$$

και αν υποθέσουμε ότι ισχύει για $l = 2n$, τότε για $l = 2(n+1)$, παρατηρούμε ότι το

$$\underbrace{11...1}_{2n+2}$$

προκύπτει από το

$$\underbrace{11...1}_{2n}$$

αν κάνουμε «left-shift» 2 φορές και προσθέσουμε 11. Είναι δηλαδή

$$\underbrace{11...1}_{2n+2} = \underbrace{11...1}_{2n} \cdot 100 + 11$$

οπότε

$$\underbrace{11...1}_{2n+2} : 11 = \left(\underbrace{11...1}_{2n} \cdot 100 + 11 \right) : 11 = \underbrace{01...01}_{2n} \cdot 100 + 11 : 11 = \underbrace{01...01}_{2n} \cdot 100 + 01 = \underbrace{01...01}_{2n+2}$$

ο.ε.δ.

5.5.1. Εφαρμογή του αλγόριθμου της Arifovic στο γραμμικό υπόδειγμα

Ξεκινάμε με την εφαρμογή του αλγόριθμου της Arifovic(1994) στο υπόδειγμα των Alkemade et al. (2007). Θα εκτελέσουμε 10.000 βήματα στην προσομοίωση και το μέγεθος του κάθε χρωμοσώματος θα είναι 20 bits. Αρχικά εξετάζουμε την περίπτωση 4 επιχειρήσεων με 30 χρωμοσώματα στον πληθυσμό της κάθε μιας. Η πιθανότητα μετάλλαξης που επιλέξαμε (μετά από τη διαπίστωση ότι εναλλακτικές τιμές, όπως οι $0{,}1$, $0{,}001$ και $0{,}005$ δεν έχουν τα καλύτερα αποτελέσματα όσον αφορά τη σύγκλιση της συνολικής ποσότητας) είναι $p_{mut} = 0{,}01$. Επίσης $p_{cr} = 1$. Θα διερευνήσουμε τη σύγκλιση στην ισορροπία Walras, που όπως προαναφέραμε είναι $q^W = 50$, οπότε $q_{\max} = 150$.

Η μέση ποσότητα των 4 επιχειρήσεων στην αγορά, παρουσιάζει μεγάλες διακυμάνσεις, στον χρονικό ορίζοντα της προσομοίωσης. Η μέση τιμή της στα 10.000 βήματα είναι 51,9213 και η τυπική της απόκλιση 11,6035. Συνεπώς η μέση ποσότητα είναι αρκετά κοντά στην μέση ποσότητα της ισορροπίας κατά Walras, παρόλο που η τυπική απόκλιση είναι υψηλή.

Βλέπουμε ότι οι 4 επιχειρήσεις δεν ισορροπούν στην ίδια ποσότητα, και συνεπώς δεν έχουμε ισορροπία κατά Walras στο μικροοικονομικό επίπεδο:

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	47,4878	28,2839
2	51,5447	26,3824
3	52,1950	27,7354
4	56,4576	28,7703

η παραπάνω ανισορροπία αντικατοπτρίζεται και στις εκτιμήτριες του διανύσματος σταθερών συχνοτήτων των 21 ομαδικών καταστάσεων ($A_0,...,A_{20}$)

0	0	0	0	0	0	0	0,0046	0,0626	0,1795	0,3952	0,2672	0,0789	0,0119	0,0001	0	0	0	0	0	0

Βλέπουμε δηλαδή ότι σε κανένα από τα 10.000 βήματα δεν βρεθήκαμε σε κατάσταση με μέση απόσταση από την ιδανική (όλα τα χρωμοσώματα ίσα με την ποσότητα της Βαλρασιανής ισορροπίας) μικρότερη από 6 bits.

Αυξάνοντας τον αριθμό των χρωμοσωμάτων στους πληθυσμούς στα 60, παρατηρούμε ότι η βελτίωση δεν είναι ιδιαίτερα θεαματική. Η μέση τιμή της μέσης ποσότητας που προσφέρουν οι 4 επιχειρήσεις, στα 10.000 βήματα της προσομοίωσης είναι

$$\frac{1}{40.000}\sum_{t=1}^{10000}\sum_{i=1}^{4} q_{it} = 51,4591$$

ενώ η τυπική απόκλισή της είναι

$$\left(\frac{1}{10000}\sum\left(\frac{1}{4}\sum_{i=1}^{4} q_{it} - \frac{1}{40.000}\sum_{t=1}^{10000}\sum_{i=1}^{4} q_{it}\right)^2\right)^{\frac{1}{2}} = 10,3130$$

παρότι υπάρχει μια μικρή βελτίωση, η μείωση δεν είναι ιδιαίτερα σημαντική.

Βλέπουμε και εδώ ότι οι 4 επιχειρήσεις δεν ισορροπούν στην ίδια ποσότητα, και συνεπώς δεν έχουμε ισορροπία κατά Walras στο μικροοικονομικό επίπεδο:

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	26,7474	18,4525
2	50,3585	17,0803
3	46,9187	19,4287
4	81,8118	19,6082

οι εκτιμήτριες του διανύσματος σταθερών συχνοτήτων των 21 ομαδικών καταστάσεων ($A_0,...,A_{20}$)

0	0	0	0	0	0	0	0,0013	0,0741	0,2558	0,3583	0,2729	0,0348	0,0028	0	0	0	0	0	0	0

Βλέπουμε δηλαδή ότι σε κανένα από τα 10.000 βήματα δεν βρεθήκαμε σε κατάσταση με μέση απόσταση από την ιδανική (όλα τα χρωμοσώματα ίσα με την ποσότητα της Βαλρασιανής ισορροπίας) μικρότερη από 7 bits, ενώ μόνο σε 13 περιόδους βρεθήκαμε στην κατάσταση μέσης απόστασης μεταξύ 7 και 8 bits.

Αν τώρα αυξήσουμε τον αριθμό των χρωμοσωμάτων στα 200, έχουμε μέση τιμή 50,9331 και τυπική απόκλιση 9,1566. Οι μέσες ποσότητες και οι αποκλίσεις για τις 4 επιχειρήσεις είναι

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	52,4763	15,2923
2	57,2339	15,9801
3	49,3282	16,3993
4	44,6939	15,2426

και οι εκτιμήτριες του διανύσματος σταθερών συχνοτήτων

0	0	0	0	0	0	0	0	0,0557	0,6194	0,3188	0,0061	0	0	0	0	0	0	0	0	0

Γενικά παρατηρούμε ότι η μεταβολή του αριθμού των χρωμοσωμάτων δεν έχει σημαντικές επιδράσεις στην σύγκλιση των ποσοτήτων των μεμονωμένων επιχειρήσεων. Μπορούμε πάντως να παρατηρήσουμε ότι, ενώ επιφέρει μια μικρή βελτίωση στην τυπική απόκλιση μεταξύ των μέσων ποσοτήτων των επιχειρήσεων, ουσιαστικά απομακρύνει -επίσης οριακά- τις ποσότητες αυτές από την Βαλρασιανή ισορροπία, όπως φαίνεται από τις χαμηλότερες συχνότητες στις καταστάσεις μικρών αποστάσεων στην εκτιμήση του σταθερού διανύσματος συχνοτήτων. Πάντως το σίγουρο είναι ότι σε καμμία περίπτωση η σύγκλιση δεν μπορεί να χαρακτηριστεί ικανοποιητική. Παρατηρούμε επίσης ότι επιβεβαιώνεται τόσο ο κανόνας των «30 χρωμοσωμάτων», όσο και ο κανόνας του Alander (1992), αφού το μέγεθος του χρωμοσώματος, στην περίπτωσή μας είναι 20 bits.

Ας μελετήσουμε τώρα την περίπτωση 20 επιχειρήσεων, οπότε η Βαλρασιανή ποσότητα για την κάθε επιχείρηση είναι $q^W = 10$. Για 30 χρωμοσώματα στους πληθυσμούς, η μέση ποσότητα στις προσομοιώσεις ήταν 10,2968 και η τυπική απόκλιση 1,7125.

Οι μέσες ποσότητες και οι τυπικές αποκλίσεις τους, ξεχωριστά για τις 20 επιχειρήσεις ήταν

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	6,3381	5,4480
2	7,3215	4,9394
3	7,0163	5,2616
4	7,1721	5,8921
5	8,8732	6,5954
6	8,8003	6,3217
7	9,6895	6,9013
8	10,4321	7,2179
9	9,2880	7,1450
10	10,4426	7,4112
11	10,4615	7,3717
12	9,9684	7,6574
13	10,3128	7,5635

14	10,7954	7,8515
15	12,5978	8,2281
16	12,3493	8,2468
17	13,2849	8,2707
18	12,9088	8,5127
19	13,1239	8,4513
20	14,7600	8,6745

Οι εκτιμήτριες του διανύσματος σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0,0046	0,6002	0,3925	0,0027	0	0	0	0	0	0	0	0

Παρατηρούμε ότι η κατάσταση είναι παρόμοια με την περίπτωση των 4 επιχειρήσεων.

Για λόγους ασφαλείας, ελέγχουμε και τα αποτελέσματα για 50 χρωμοσώματα στους πληθυσμούς. Η μέση ποσότητα στην αγορά είναι 10,3266 και η τυπική της απόκλιση 1,8637. Ο πίνακας των ποσοτήτων των μεμονωμένων επιχειρήσεων είναι

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	5,2677	3,6589
2	6,3049	3,5754
3	9,6578	3,8877
4	9,3825	5,2393
5	9,4345	4,5464
6	9,3202	4,6014
7	11,5379	5,8716
8	8,8444	4,7568
9	10,4981	5,8990
10	9,4985	5,4640
11	10,0442	6,7099
12	10,8939	6,3065
13	11,4442	7,0540
14	11,5784	6,8248
15	11,5721	7,0449
16	10,6022	7,1725
17	12,9186	7,5339
18	13,1812	7,6258
19	12,5419	7,6691
20	12,0096	7,8424

και οι εκτιμήτριες των συχνοτήτων στην ισορροπία είναι

0	0	0	0	0	0	0	0	0	0,0051	0,7068	0,288	0,0001	0	0	0	0	0	0	0	0

Όπως βλέπουμε δεν υπάρχουν σημαντικές διαφορές.

5.5.2. Εφαρμογή του αλγορίθμου της Arifovic στο πολυωνυμικό μοντέλο

Όπως σε όλες τις εφαρμογές του αλγόριθμου της Arifovic, έτσι κι εδώ θα διερευνήσουμε τη σύγκλιση στην Βαλρασιανή ισορροπία. Θέτουμε $p_{mut} = 0{,}01$ $p_{cr} = 1$ και $l = 20$. Ξεκινάμε με $n = 4$ και 30 χρωμοσώματα στον κάθε πληθυσμό. $q_{\max} = 3q^W = 314{,}3070$.

Η μέση ποσότητα στην αγορά είναι 109,1381 και η τυπική της απόκλιση 24,4340. Για τις μεμονωμένες επιχειρήσεις έχουμε:

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	117,8133	60,6076
2	108,7017	57,4365
3	108,5176	49,5079
4	101,5198	51,6820

Το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0,0029	0,0462	0,1879	0,4376	0,2871	0,0373	0,001	0	0	0	0	0	0

Ελέγχουμε και την περίπτωση με 50 χρωμοσώματα στους πληθυσμούς: η μέση ποσότητα είναι 108,7213, η τυπική της απόκλιση 21,7422 και οι αντίστοιχες για τις μεμονομένες επιχειρήσεις

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	93,9067	41,3519
2	129,3700	48,4475
3	111,4694	37,1168
4	100,1391	43,1436

Το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0,0329	0,2099	0,2971	0,2679	0,1588	0,0333	0,0001	0	0	0	0	0	0

Για την περίπτωση τώρα των 20 επιχειρήσεων, με 30 χρωμοσώματα στον πληθυσμό της κάθε επιχείρησης, η μέση ποσότητα είναι 21,5606, η τυπική της απόκλιση 3,5970, οι αντίστοιχες τιμές για τις μεμονωμένες επιχειρήσεις

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	10,9937	8,6340
2	15,7589	11,1095
3	16,0333	11,1193
4	16,6001	12,2316
5	16,9216	11,9966

6	18,4669	12,7230
7	18,6253	14,1289
8	18,4820	13,7028
9	19,6210	14,7023
10	21,4743	15,1088
11	24,6979	16,7655
12	23,9749	16,2333
13	24,4901	16,6844
14	23,1379	16,9761
15	24,3291	17,1530
16	26,5954	16,7206
17	27,3239	17,2451
18	27,1440	17,2053
19	28,5799	17,0876
20	27,9607	18,0050

και το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων

0	0	0	0	0	0	0	0	0	0,006	0,6612	0,332	0,0008	0	0	0	0	0	0	0

Ελέγχουμε και την περίπτωση με 50 χρωμοσώματα στους πληθυσμούς των 20 επιχειρήσεων.

Η μέση ποσότητα είναι 21,7453, η τυπική της απόκλιση 3,8584 και οι αντίστοιχες τιμές για τις μεμονωμένες επιχειρήσεις

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	27,9905	9,2673
2	18,4408	8,0430
3	16,2743	8,3925
4	17,0001	9,4056
5	18,8416	11,2096
6	18,4893	10,9961
7	28,3217	13,4510
8	21,4425	11,7925
9	18,8345	9,7696
10	19,7899	12,7239
11	21,3786	12,5331
12	23,0934	13,5397
13	21,1094	13,6065
14	24,3916	14,5872
15	20,7773	13,2139
16	24,3045	15,4689

17	21,6676	14,7460
18	22,9789	15,0225
19	25,7263	15,9124
20	24,0528	16,2652

Το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0,0201	0,7348	0,2451	0	0	0	0	0	0	0	0

5.5.3. Εφαρμογή του αλγορίθμου της Arifovic στο μοντέλο με ρητό εκθέτη

Όπως σε όλες τις εφαρμογές του αλγόριθμου της Arifovic, έτσι κι εδώ θα διερευνήσουμε τη σύγκλιση στην Βαλρασιανή ισορροπία. Θέτουμε $p_{mut} = 0{,}01$ $p_{cr} = 1$ και $l = 20$. Ξεκινάμε με $n = 4$ και 30 χρωμοσώματα στον κάθε πληθυσμό. $q_{\max} = 3q^W = 304{,}794$.

Η μέση ποσότητα στην αγορά είναι 105,6356 και η τυπική της απόκλιση 23,5168. Για τις μεμονωμένες επιχειρήσεις έχουμε:

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	115,8250	53,0506
2	99,7182	54,2406
3	111,9511	58,2062
4	95,0271	55,9633

Το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0,0174	0,1876	0,3907	0,3188	0,0812	0,0043	0	0	0	0	0	0

Ελέγξαμε και την περίπτωση με 50 χρωμοσώματα στους πληθυσμούς και δεν υπήρξαν σημαντικές διαφορές.

Για την περίπτωση τώρα των 20 επιχειρήσεων, με 30 χρωμοσώματα στον πληθυσμό της κάθε επιχείρησης, η μέση ποσότητα είναι 20,9317, η τυπική της απόκλιση 3,4304, οι αντίστοιχες τιμές για τις μεμονωμένες επιχειρήσεις

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	16,3942	12,0690
2	11,9587	9,2895
3	13,0068	11,4939
4	14,6362	11,2123
5	14,6323	12,3353
6	18,2542	12,7541

7	15,9473	12,4385
8	20,1375	14,4212
9	17,8157	14,4650
10	20,5347	14,8321
11	24,3146	15,7743
12	23,5678	15,7331
13	19,3012	14,9434
14	25,5770	16,7807
15	26,6338	17,2674
16	23,7389	15,9107
17	29,2404	17,3805
18	26,8610	17,2258
19	28,0081	17,3637
20	28,0740	17,2653

και το εκτιμώμενο διάνυσμα σταθερών συχνοτήτων

0	0	0	0	0	0	0	0	0	0,0099	0,6293	0,3604	0,0004	0	0	0	0	0	0	0	0

Η περίπτωση με 50 χρωμοσώματα στους πληθυσμούς των 20 επιχειρήσεων, δεν είχε σημαντικές διαφοροποιήσεις.

5.5.4. Εφαρμογή του αλγόριθμου του Vriend στο γραμμικό υπόδειγμα

Γενικά θα υιοθετήσουμε $GArate = 50$, όπως προτείνουν οι Valee and Yildizoglou (2007).

Αρχικά μελετάμε την περίπτωση των $n = 4$ επιχειρήσεων. Διαπιστώσαμε ότι η σύγκλιση στην ισορροπία κατά Nash δεν επηρρεάζεται σημαντικά, από διαφορές στην τιμή του τελεστή διασταύρωσης. Για 25.000 γενεές, $p_{mut} = 0{,}01$, $GArate = 50$, $l = 20$ και 30 χρωμοσώματα στους πληθυσμούς οι εκτιμήτριες του διανύσματος σταθερών συχνοτήτων ήταν:

$p_{cr} = 0{,}1$

0	0	0	0	0	0	0	0	0,028	0,624	0,304	0,044	0	0	0	0	0	0	0	0	0

$p_{cr} = 0{,}5$

0	0	0	0	0	0	0	0	0,004	0,482	0,504	0,01	0	0	0	0	0	0	0	0	0

$p_{cr} = 0{,}75$

0	0	0	0	0	0	0	0	0	0	0,044	0,48	0,462	0,014	0	0	0	0	0	0	0

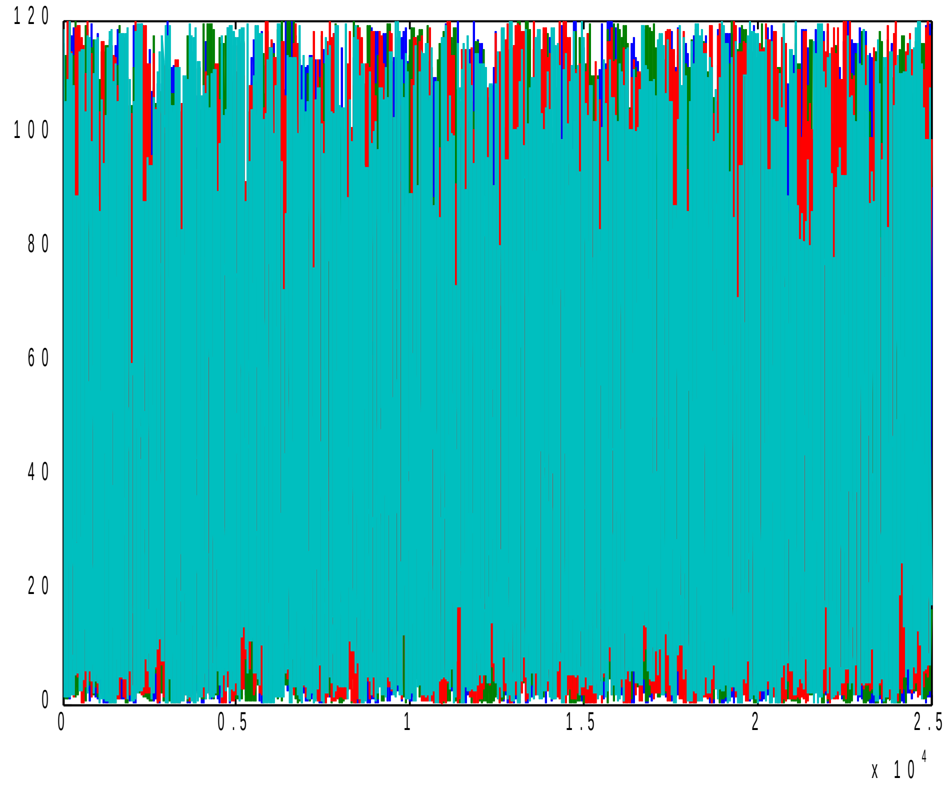
$p_{cr} = 1$

0	0	0	0	0	0	0	0	0	0	0,186	0,204	0,508	0,102	0	0	0	0	0	0	0

Στη συνέχεια επιλέξαμε $p_{cr} = 1$. Αντίθετα διαφορετικές πιθανότητες μετάλλαξης, μπορούν να επηρεάσουν το αποτέλεσμα. Για τις ίδιες τιμές με παραπάνω για τον αριθμό γενεών, χρωμοσωμάτων και GA-rate

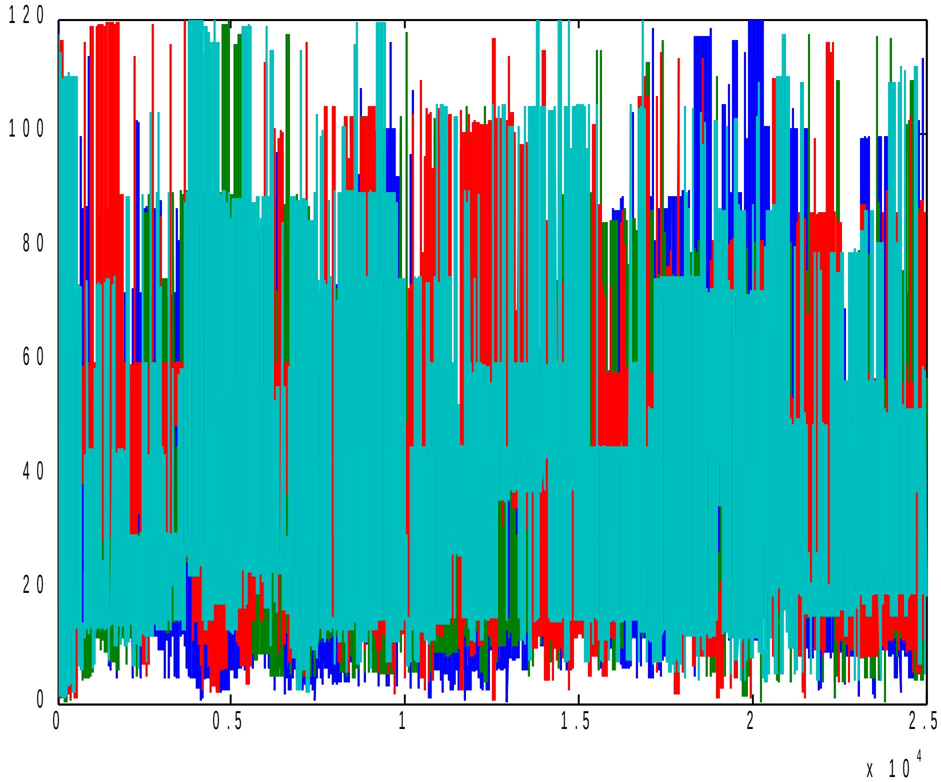
p_{mut} = 0,1

0	0	0	0	0	0	0	0	0	0,016	0,678	0,306	0	0	0	0	0	0	0	0	0



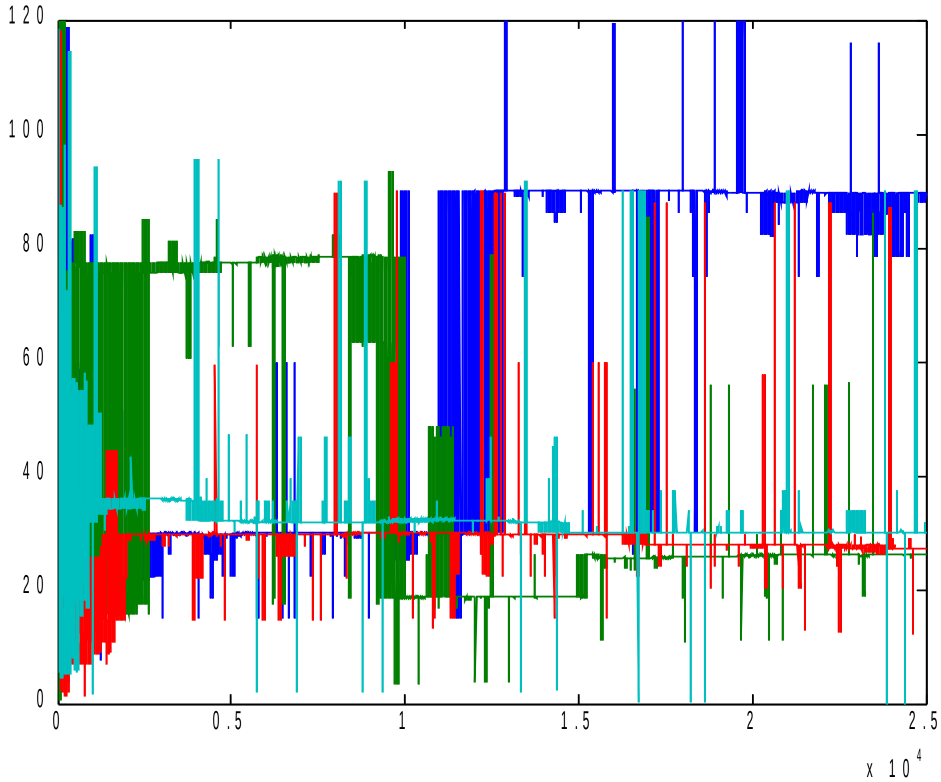
p_{mut} = 0,01

0	0	0	0	0	0	0	0	0,002	0,228	0,45	0,248	0,072	0	0	0	0	0	0	0	0



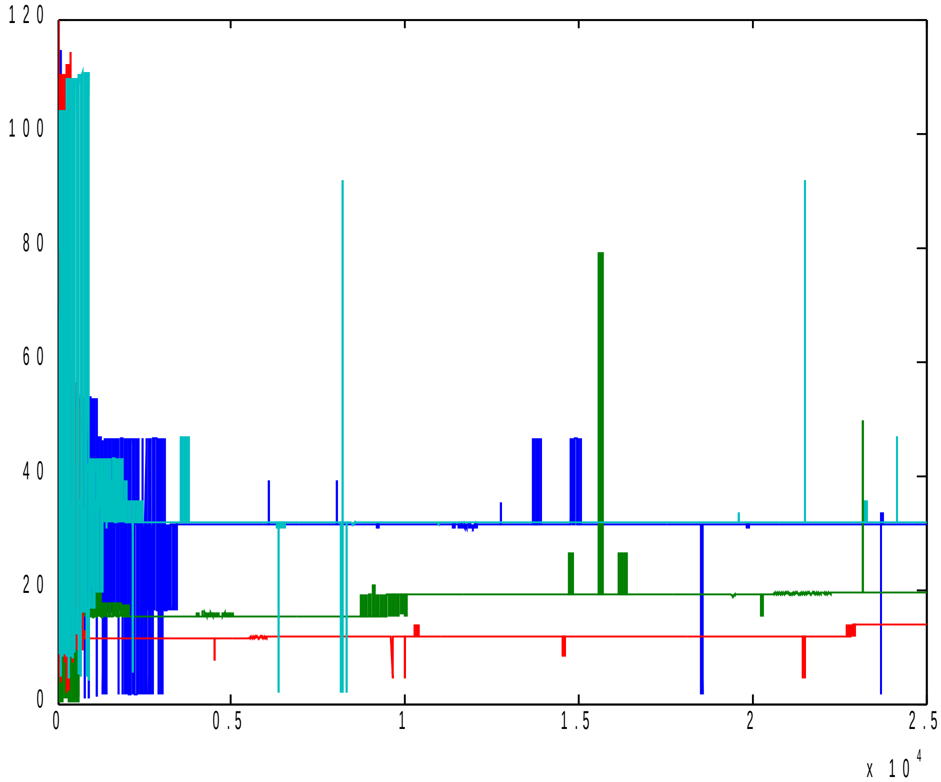
p_{mut} = 0,001

0	0	0	0	0	0	0	0	0	0,066	0,556	0,332	0,046	0	0	0	0	0	0	0	0



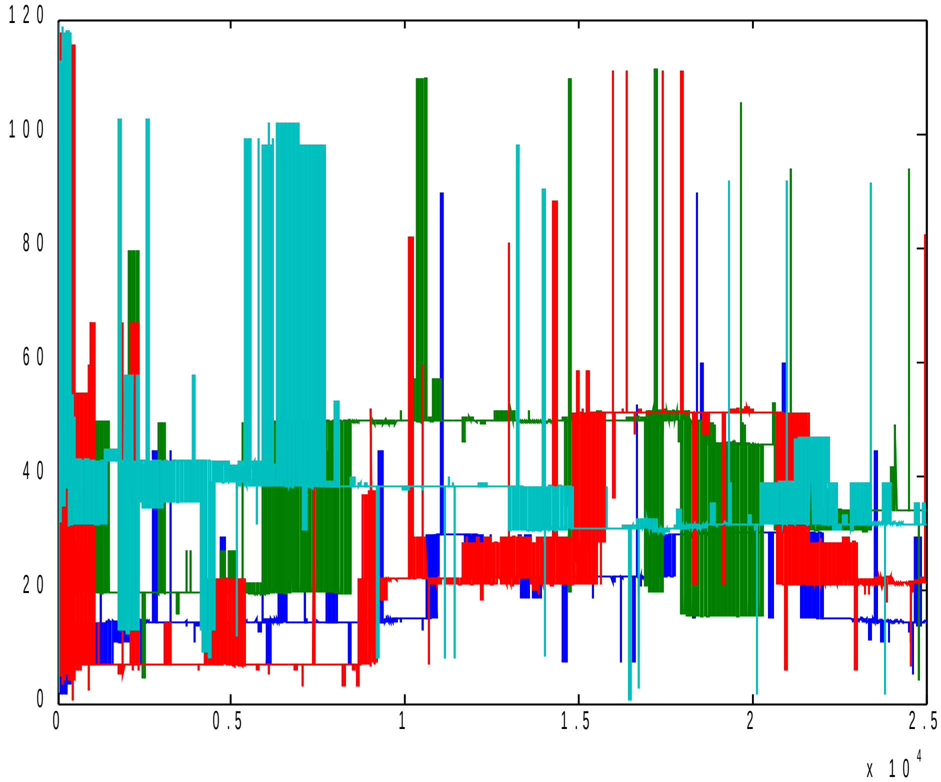
p_{mut} = 0,0001

0	0	0	0	0	0	0	0	0	0	0,956	0,044	0	0	0	0	0	0	0	0	0



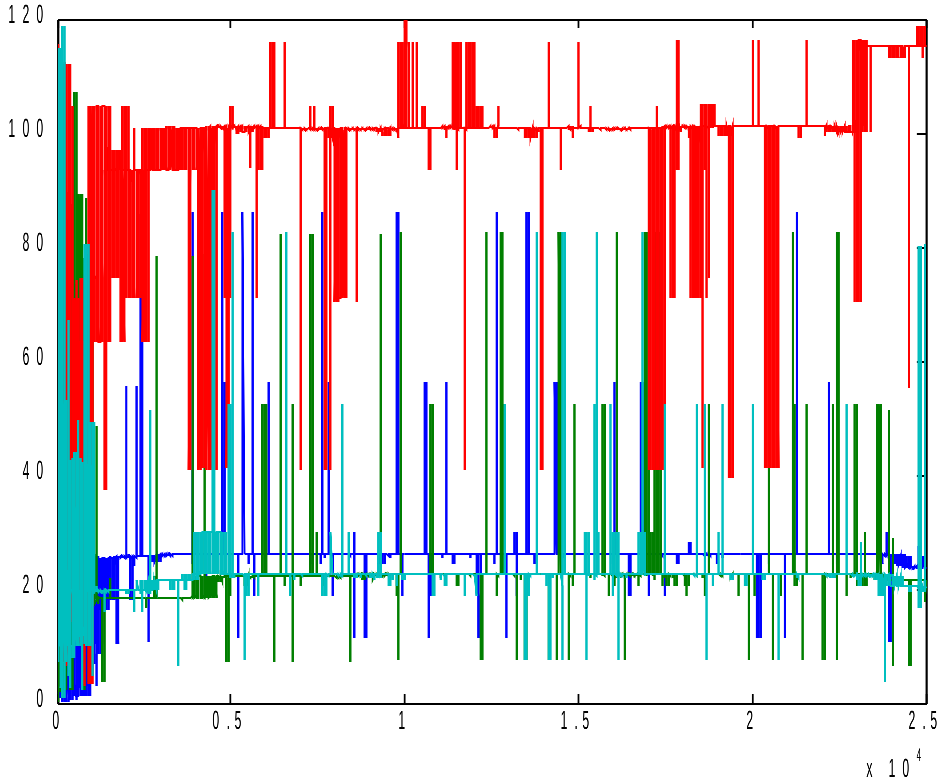
Ο αριθμός των χρωμοσωμάτων στον πληθυσμό για τις ίδιες τιμές παραμέτρων, $chrom = 20$

0	0	0	0	0	0	0	0	0	0,266	0,668	0,066	0	0	0	0	0	0	0	0	0



$chrom = 30$

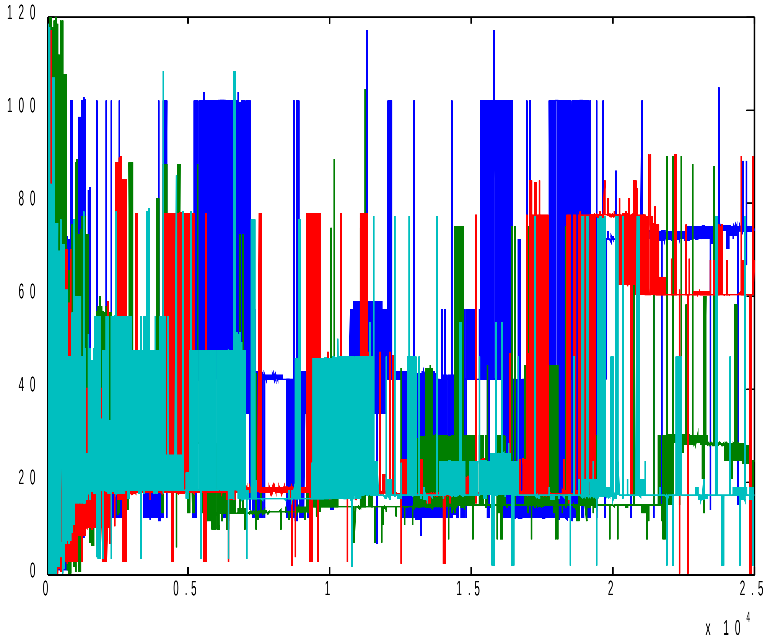
0	0	0	0	0	0	0	0	0	0	0,664	0,326	0,01	0	0	0	0	0	0	0	0



Παρατηρούμε μάλιστα εδώ, όπου οι τιμές των παραμέτρων είναι ακριβώς οι ίδιες με την περίπτωση για $p_{mut} = 0{,}01$ παραπάνω, ότι οι παίκτες «κολλάνε» και πάλι, σε διαφορετικές ποσότητες αυτήν την φορά

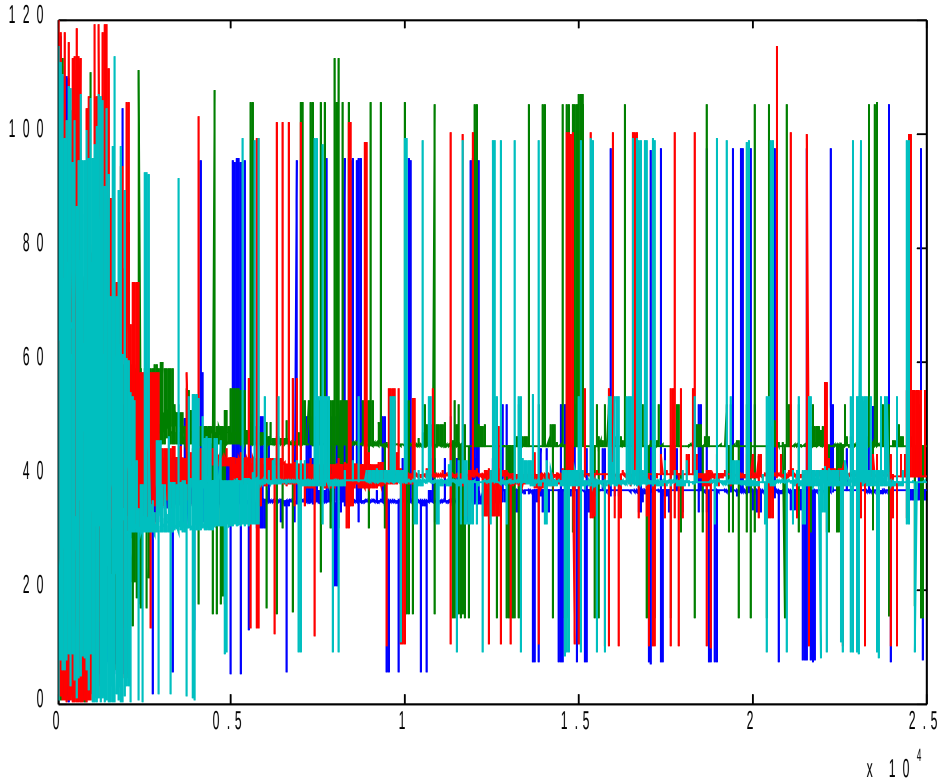
$chrom = 50$

0	0	0	0	0	0	0	0	0	0,028	0,13	0,436	0,406	0	0	0	0	0	0	0	0



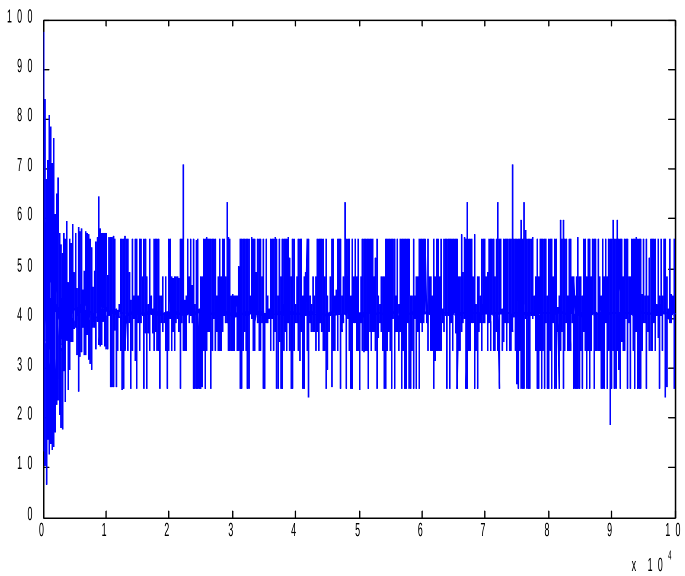
$chrom = 100$

0	0	0	0	0	0	0	0,094	0,39	0,3	0,19	0,026	0	0	0	0	0	0	0	0	0

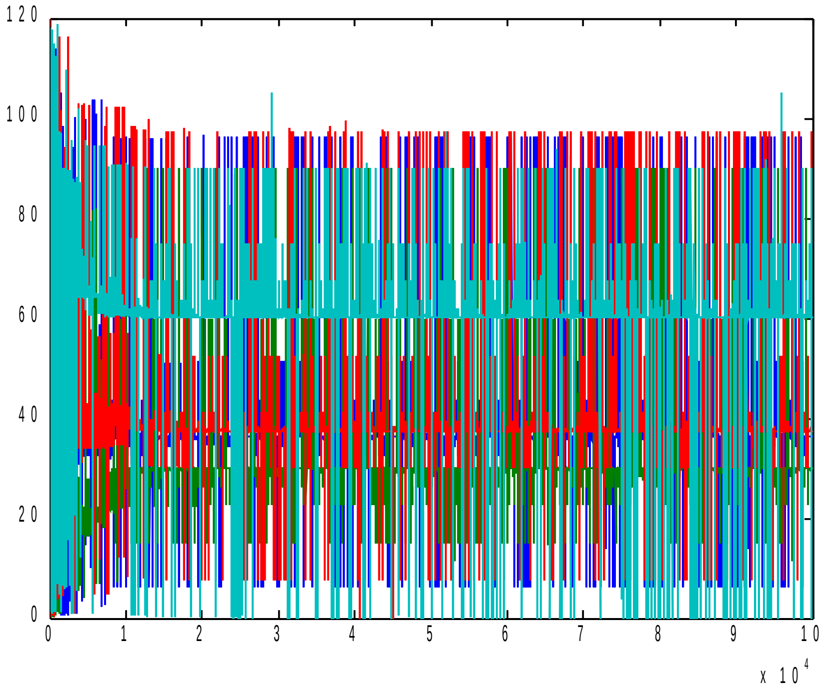


Βλέπουμε μια βελτίωση για μεγαλύτερους πληθυσμούς. Η σημαντικότερη παρατήρηση είναι βέβαια ότι η διαφοροποίηση, μεταξύ των σημείων σύγκλισης των παικτών είναι σημαντική, όπως φαίνεται τόσο από το διάγραμμα, όσο και από το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων.

Τελικά, για $GArate = 50$, $p_{mut} = 0{,}01$, $p_{cr} = 1$, 100 χρωμοσώματα στον πληθυσμό και 100.000 γενεές έχουμε μέση ποσότητα στην αγορά 41,0843 και τυπική απόκλιση 2,6216



Για τις μεμονωμένες επιχειρήσεις



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	36,6576	5,4517
2	29,5050	5,2253
3	37,8722	4,8318
4	60,3023	6,2368

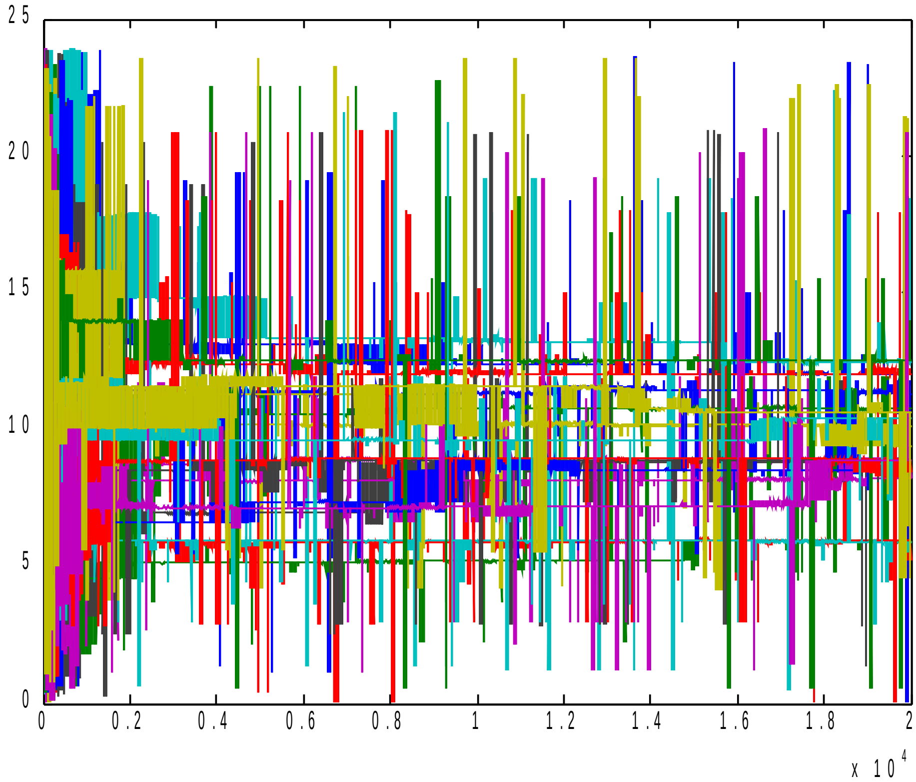
ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0,0925	0,6685	0,239	0	0	0	0	0	0	0	0	0

Για την περίπτωση των 20 επιχειρήσεων τώρα, οπότε $q^N = 9{,}5238$, για $GArate = 50$, $p_{cr} = 1$, $p_{mut} = 0{,}01$, $l = 20$ και 20.000 γενεές δοκιμάζουμε πρώτα διαφορετικούς πληθυσμούς.

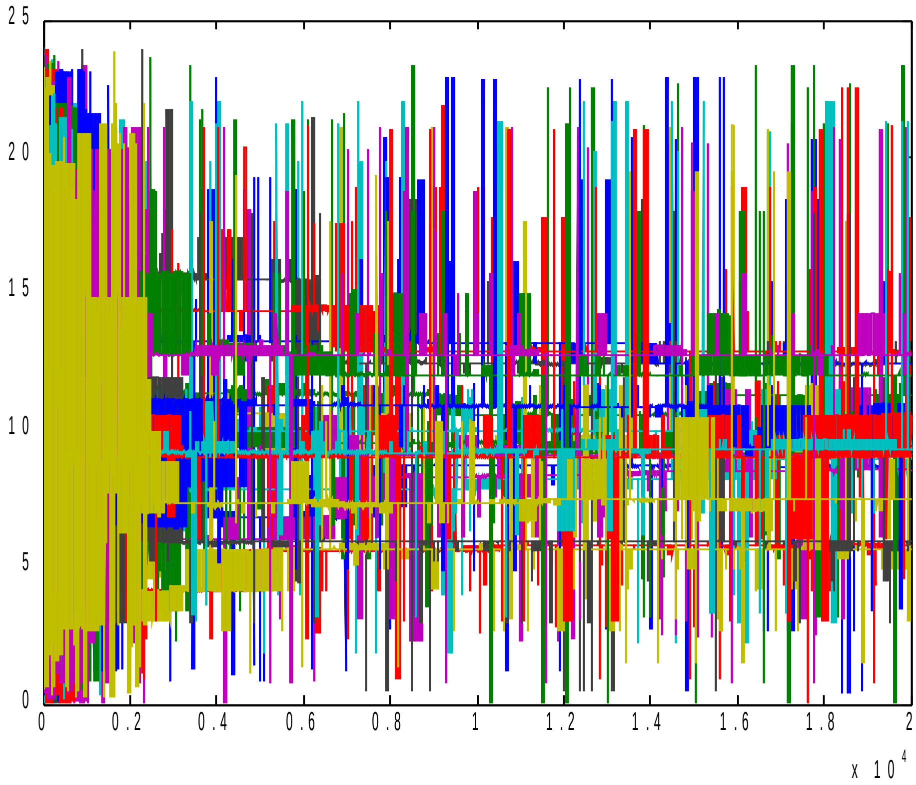
$chrom = 20$

0	0	0	0	0	0	0	0	0	0,5975	0,4025	0	0	0	0	0	0	0	0	0



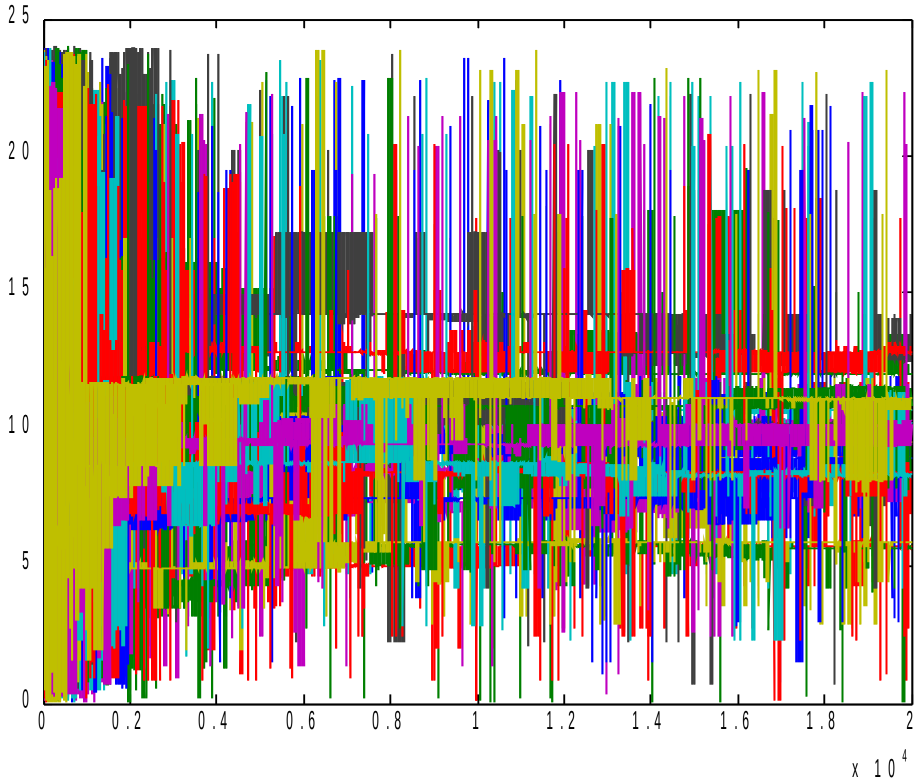
chrom = 30

0	0	0	0	0	0	0	0	0	0	0,62	0,38	0	0	0	0	0	0	0	0	0	



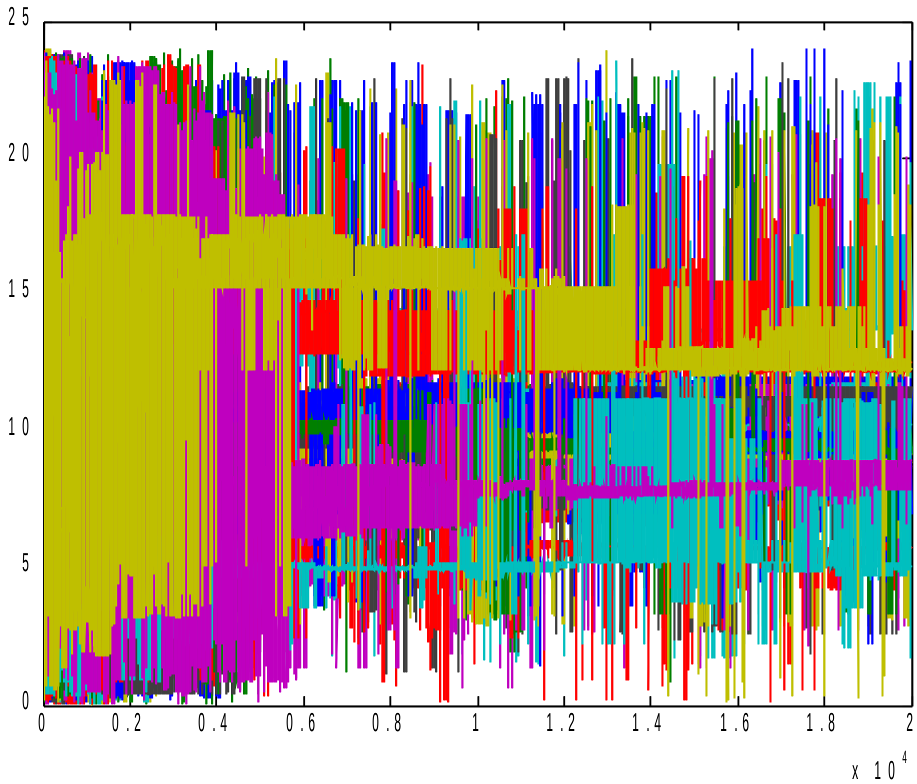
chrom = 50

0	0	0	0	0	0	0	0	0	0,2675	0,7275	0,005	0	0	0	0	0	0	0	0	0



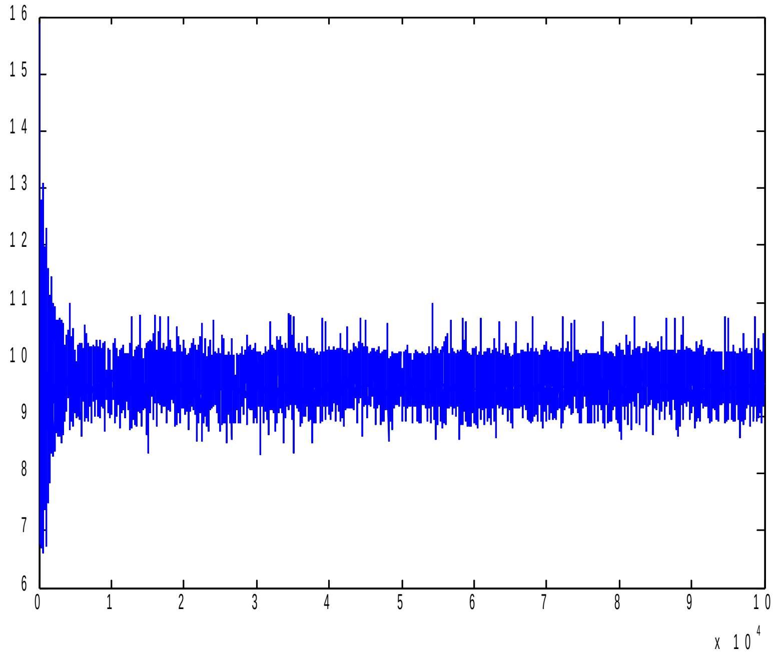
chrom = 100

0	0	0	0	0	0	0	0	0	0,245	0,6425	0,1125	0	0	0	0	0	0	0	0	0

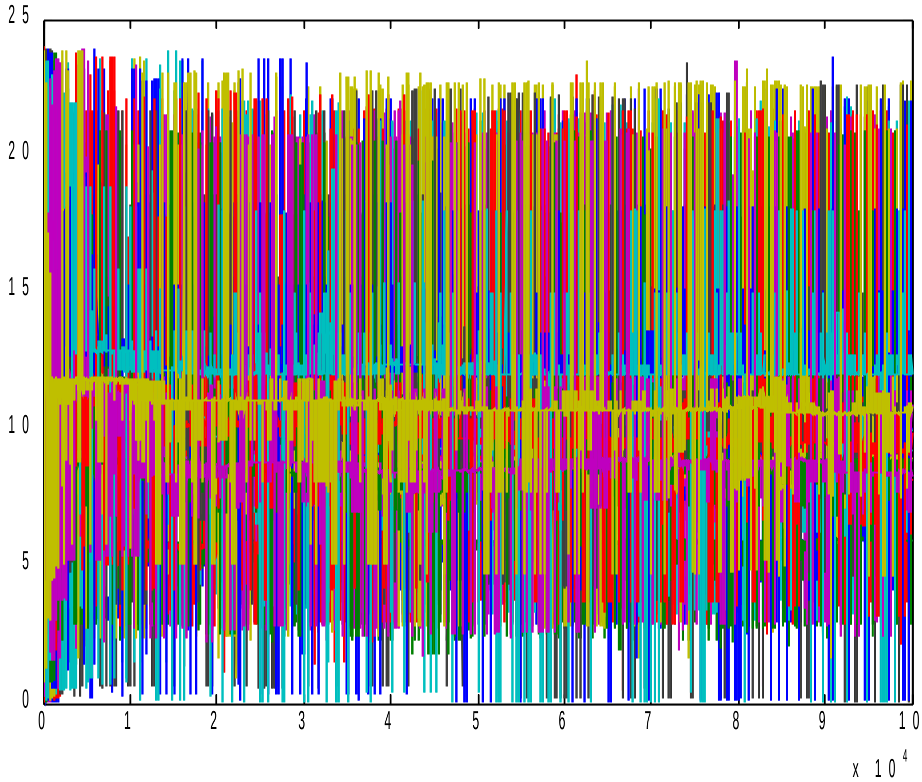


Εδώ τα αποτελέσματα για 50 χρωμοσώματα ήταν ελαφρώς καλύτερα από των 100 χρωμοσωμάτων, για αυτό θα τα επιλέξουμε για την τελική εκτέλεση.

Έτσι για $GArate = 50$, $p_{mut} = 0{,}01$, $p_{cr} = 1$, 50 χρωμοσώματα στους πληθυσμούς και 100.000 γενεές έχουμε μέση ποσότητα στην αγορά 9,5257 και τυπική απόκλιση 0,2153



Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	5,7133	1,0403
2	8,5611	0,8896
3	9,6171	1,0115
4	9,7837	1,3095
5	9,4951	1,1546
6	10,6780	0,9165
7	12,4103	0,9480
8	10,4449	0,9904
9	6,0609	1,3467
10	7,8769	1,7361
11	9,1093	1,3928
12	8,7747	1,0017
13	8,6715	0,8775
14	10,1284	1,0694
15	12,3121	1,0986
16	8,6190	0,9355
17	9,9472	0,9173
18	12,4606	1,7583
19	8,9311	1,2803
20	10,9182	0,9289

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

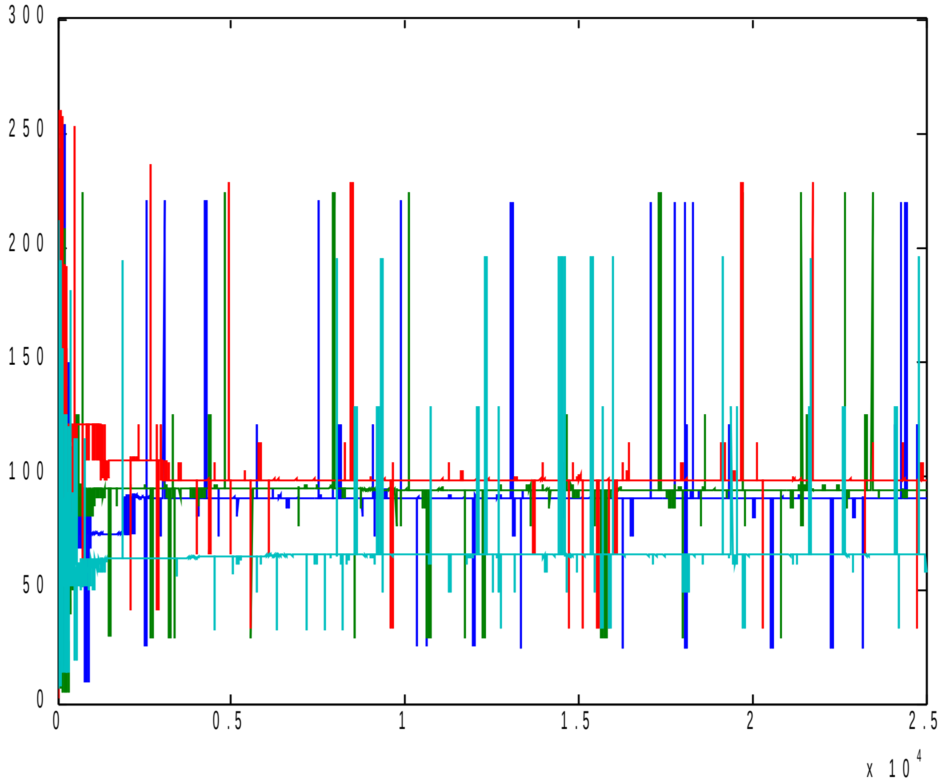
0	0	0	0	0	0	0	0	0	0	0,866	0,134	0	0	0	0	0	0	0	0	0

5.5.5. Εφαρμογή του αλγόριθμου του Vriend στο πολυωνυμικό υπόδειγμα

Εφαρμόζουμε τον αλγόριθμο του Vriend στο πολυωνυμικό υπόδειγμα (βλ. ενότητα 3). Για την περίπτωση των 4 επιχειρήσεων και για $GArate = 50$, $p_{mut} = 0.,01$, $l = 20$, $p_{cr} = 1$ και αριθμό γενεών 25.000, έχουμε για διαφορετικούς πληθυσμούς:

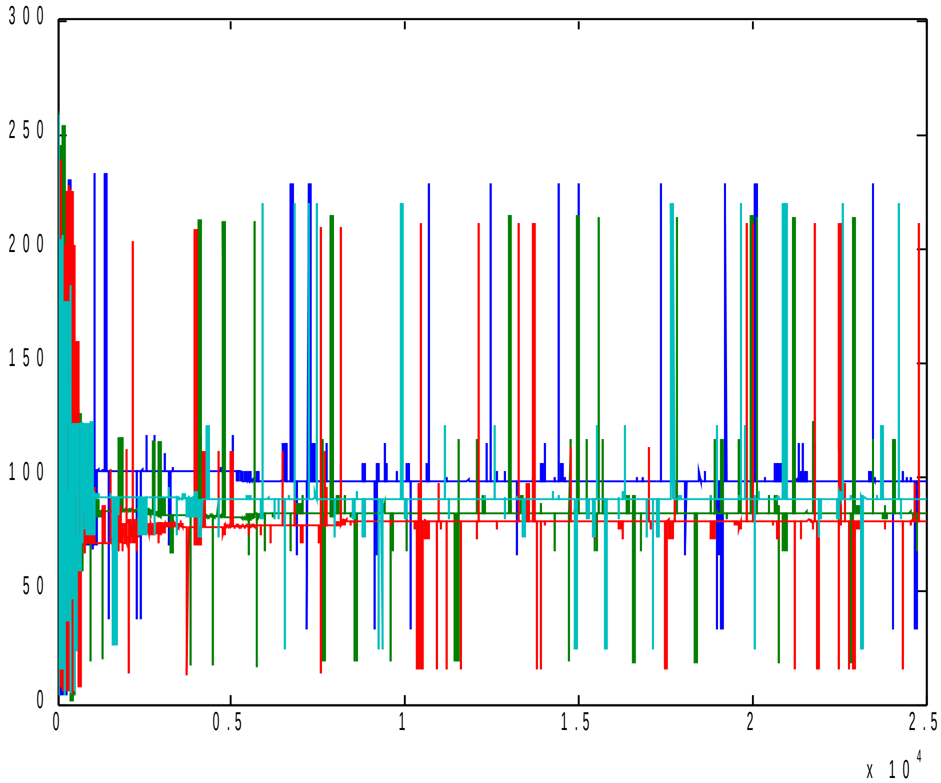
$chrom = 20$

0	0	0	0	0	0	0	0	0	0,002	0,598	0,4	0	0	0	0	0	0	0	0	0



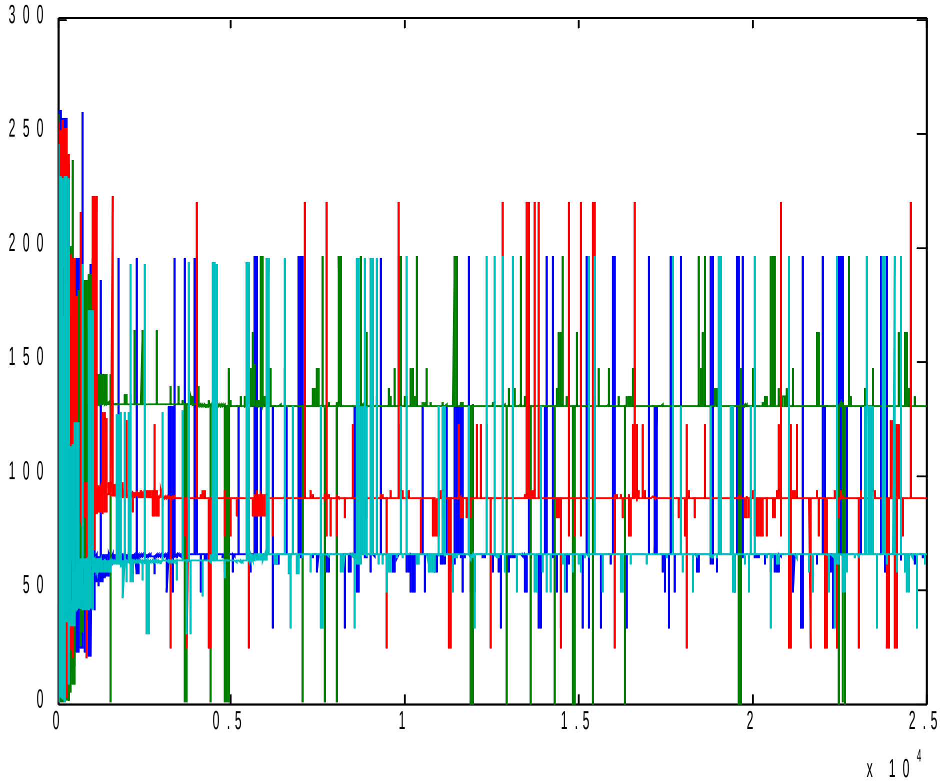
chrom = 30

0	0	0	0	0	0	0	0	0,11	0,122	0,584	0,184	0	0	0	0	0	0	0	0	0



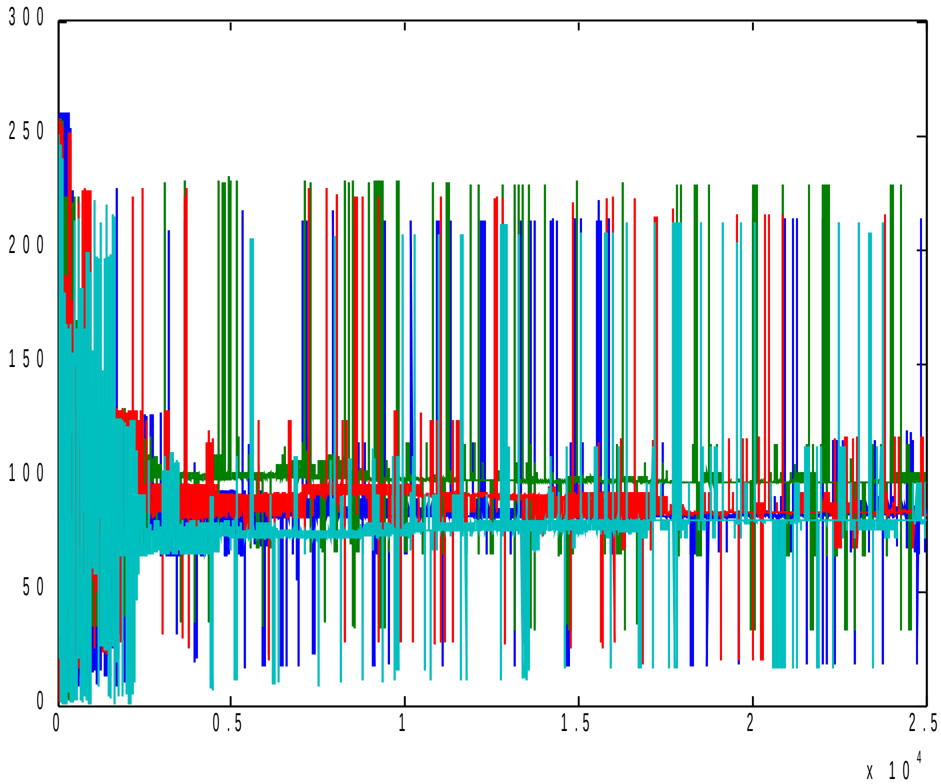
chrom = 50

0	0	0	0	0	0	0	0	0	0,168	0,394	0,438	0	0	0	0	0	0	0	0	0



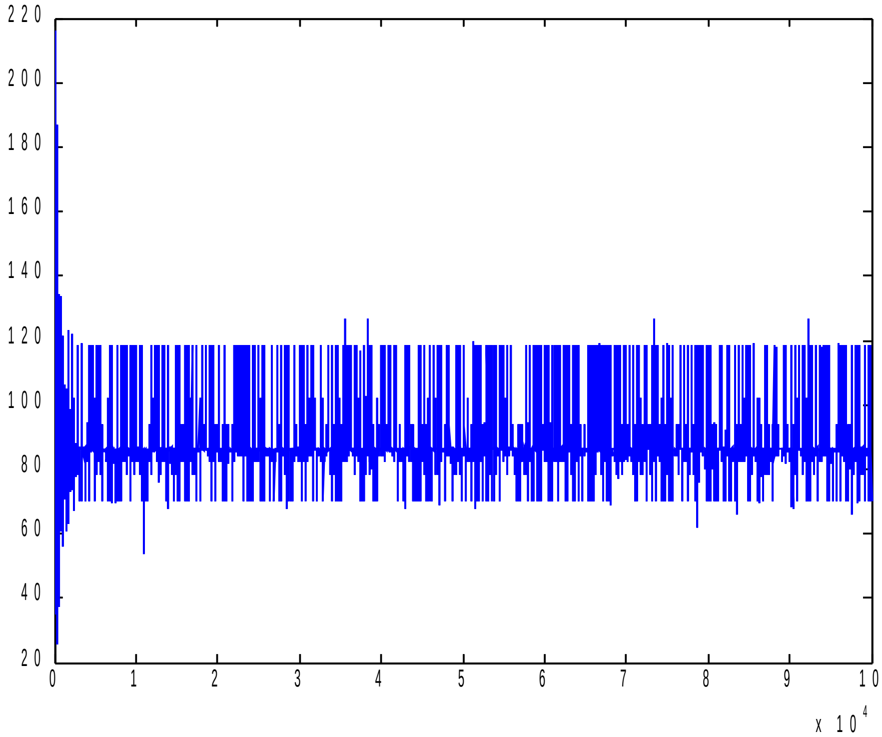
chrom = 100

0	0	0	0	0	0	0	0	0,306	0,528	0,166	0	0	0	0	0	0	0	0	0	0

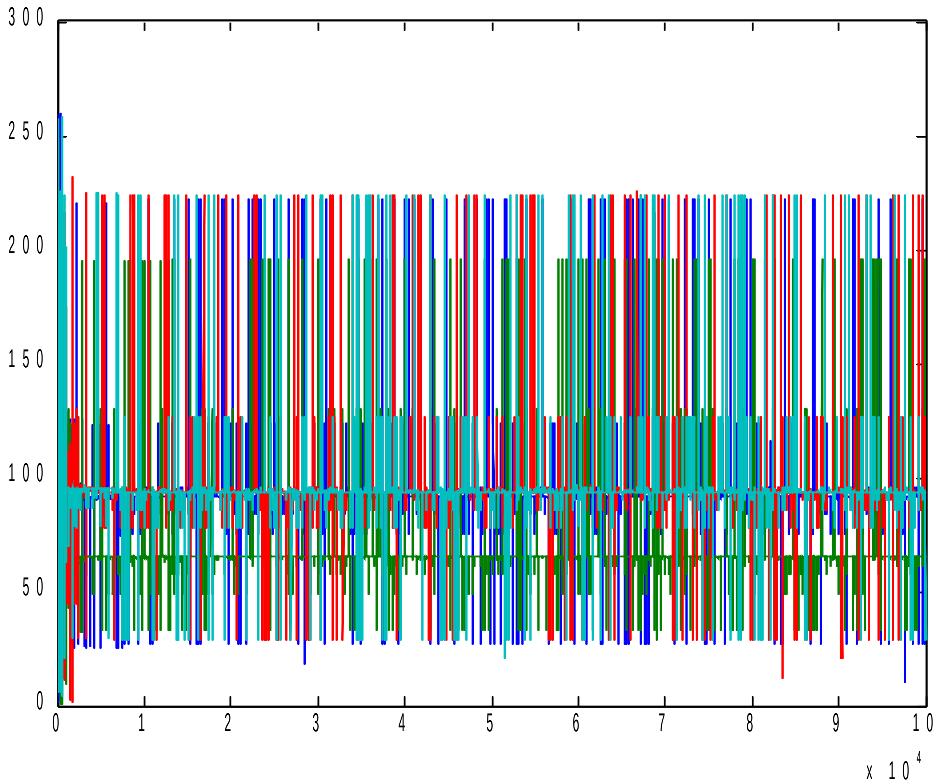


Βλέπουμε ότι όσο μεγαλύτερο πληθυσμό έχουμε, τόσο το καλύτερο (κάτι αναμενόμενο, βέβαια). Εμείς θα επιλέξουμε 50 χρωμοσώματα για την τελική εκτέλεση. Έχουμε λοιπόν, για $GArate = 50$,

$p_{cr} = 1$, $p_{mut} = 0{,}01$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 100.000 γενεές, μέση ποσότητα στην αγορά 86,2807 και τυπική απόκλιση 3,9776.



Για τις μεμονωμένες επιχειρήσεις έχουμε



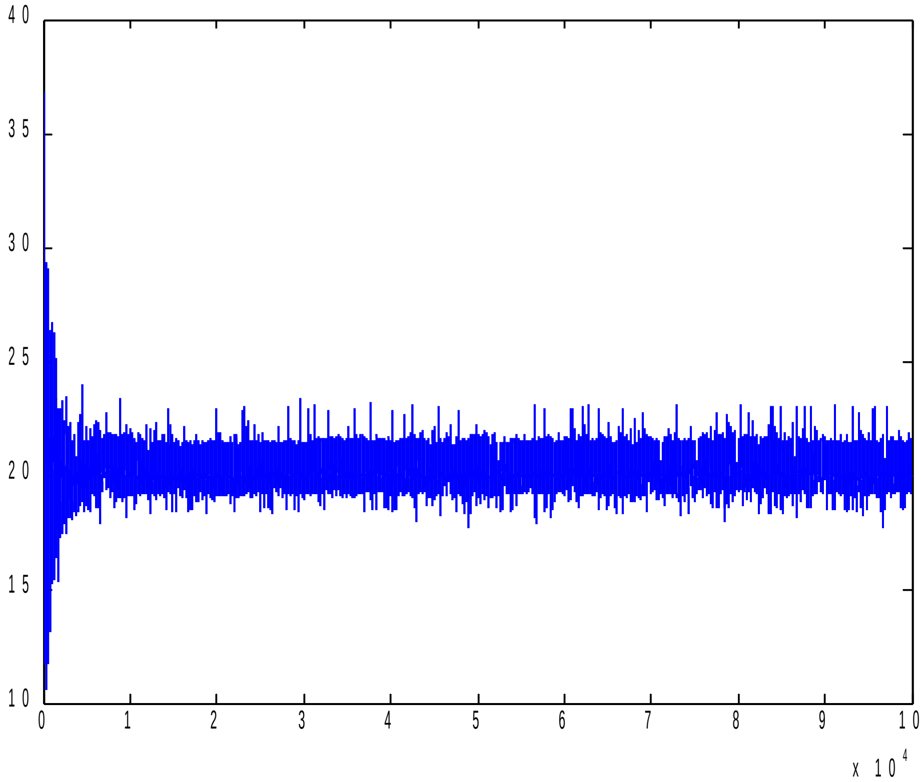
Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση

1	91,8309	7,8282
2	65,3700	7,9961
3	93,9287	7,4739
4	93,9933	7,3539

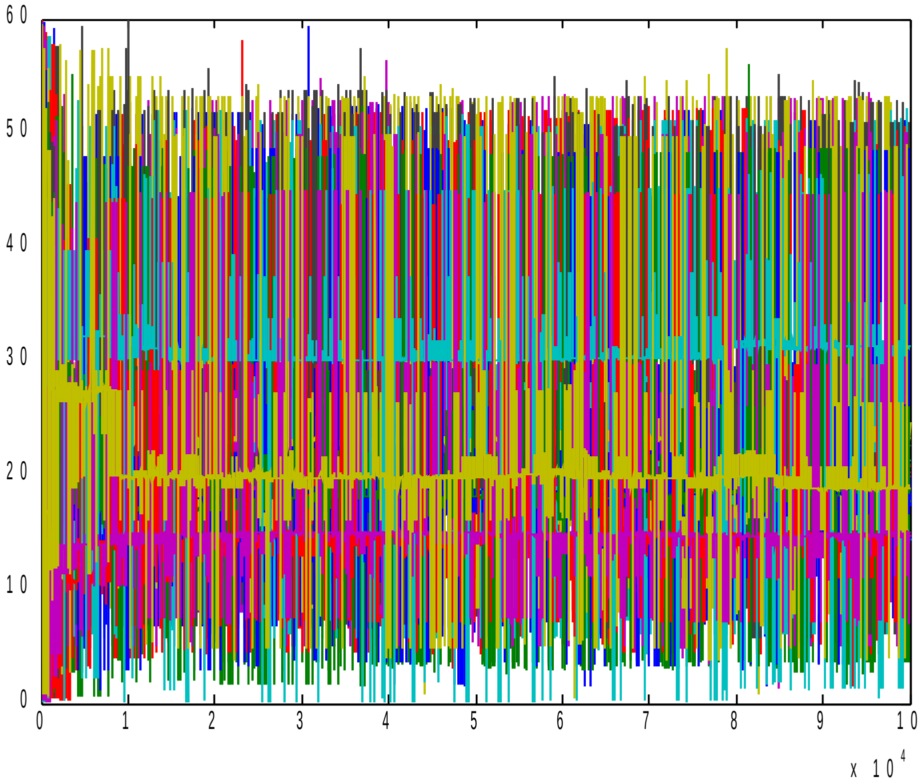
ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0,8725	0,0775	0,05	0	0	0	0	0	0	0	0	0

Για την περίπτωση των 20 επιχειρήσεων τώρα, με $l = 20$, $p_{cr} = 1$, $p_{mut} = 0{,}01$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 100.000 γενεές, έχουμε μέση ποσότητα στην αγορά 20,0320 και τυπική απόκλιση 0,5597



Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	21,4005	3,0632
2	17,8883	2,1360
3	21,4405	2,5044
4	14,7491	2,3672
5	22,9000	2,3624
6	23,2626	2,0226
7	23,4781	2,6210
8	14,5969	2,2615
9	18,4771	2,6851
10	20,2756	3,0438
11	20,7799	2,6258
12	20,0468	2,1132
13	23,2623	2,3004
14	21,0713	2,4309
15	18,3519	2,0457
16	18,5202	2,9243
17	14,2906	2,3960
18	30,7261	2,6651
19	14,7737	2,1866
20	20,3478	2,8470

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

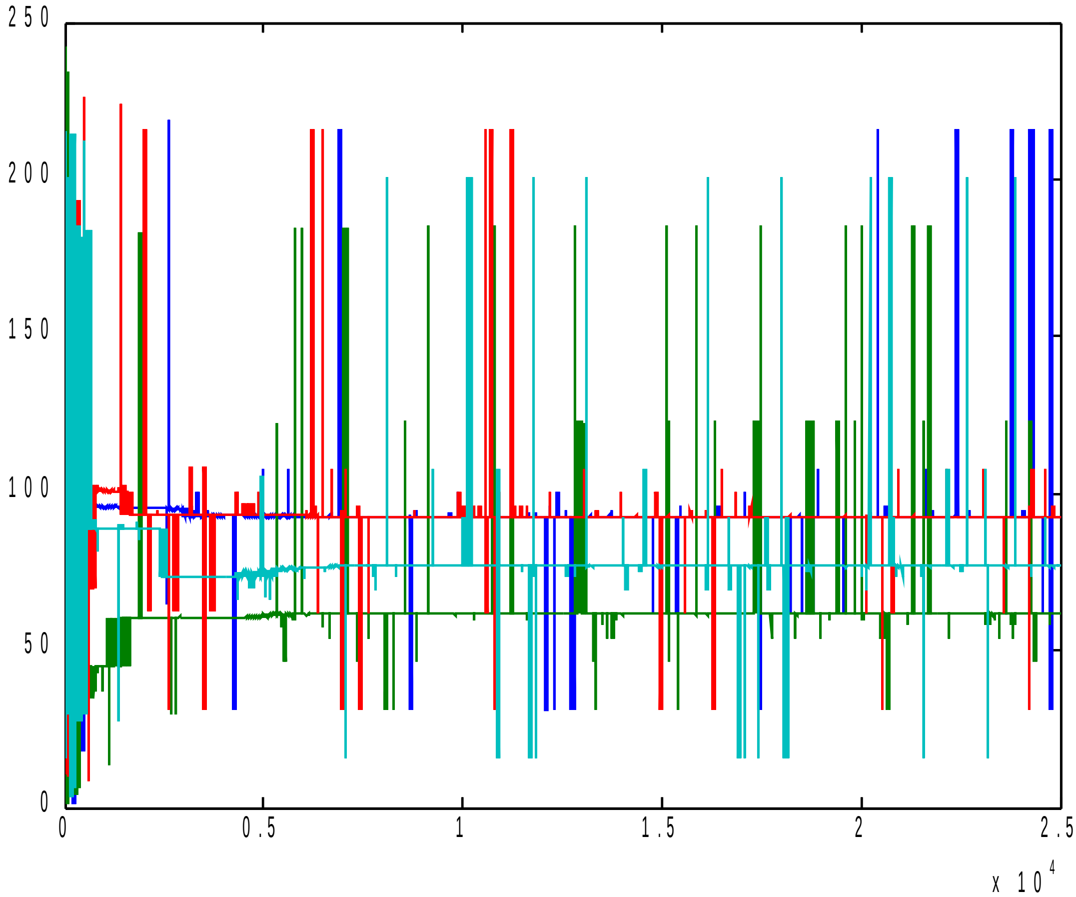
0	0	0	0	0	0	0	0	0	0,414	0,586	0	0	0	0	0	0	0	0	0	0

5.5.6. Εφαρμογή του αλγόριθμου του Vriend στο υπόδειγμα με ρητό εκθέτη

Εφαρμόζουμε τον αλγόριθμο του Vriend στο υπόδειγμα με ρητό εκθέτη (βλ. ενότητα 3). Για την περίπτωση των 4 επιχειρήσεων ($q^N = 82{,}143$) και για $GArate = 50$, $l = 20$, $p_{mut} = 0{,}01$, $p_{cr} = 1$ και αριθμό γενεών 25.000, έχουμε για διαφορετικούς πληθυσμούς:

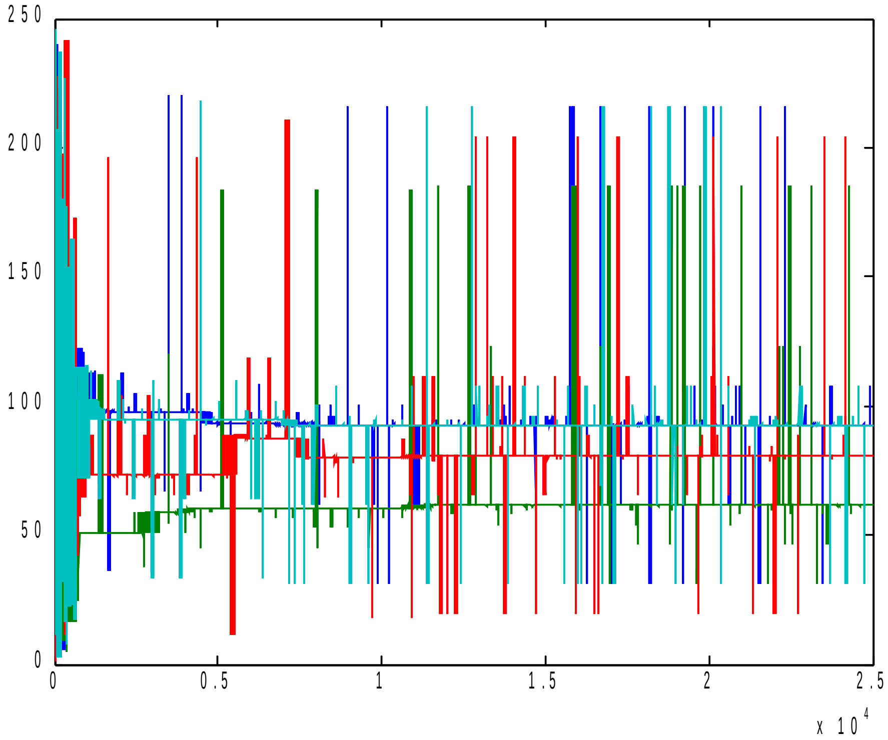
$chrom = 20$

0	0	0	0	0	0	0	0	0	0,03	0,95	0,02	0	0	0	0	0	0	0	0	0



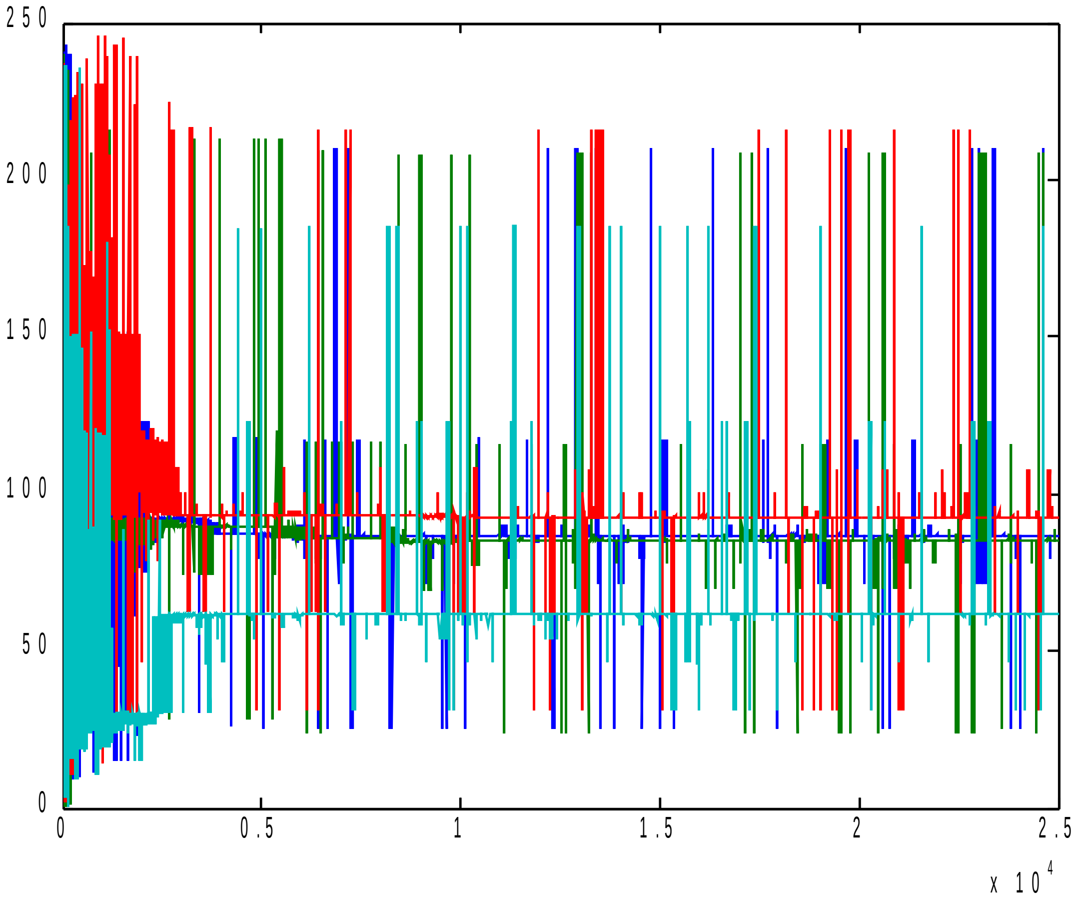
$chrom = 30$

0	0	0	0	0	0	0	0	0	0	0,706	0,242	0,052	0	0	0	0	0	0	0	0



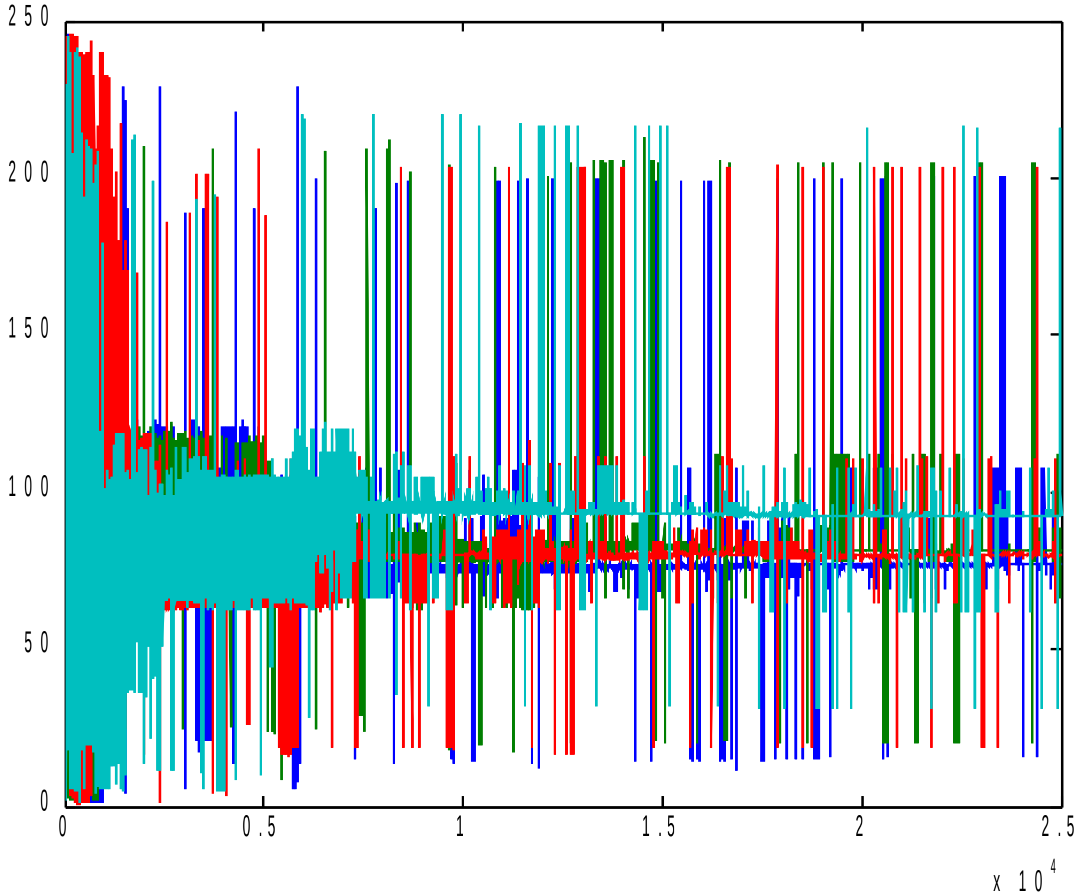
chrom = 50

0	0	0	0	0	0	0	0	0	0,084	0,916	0	0	0	0	0	0	0	0	0	0	

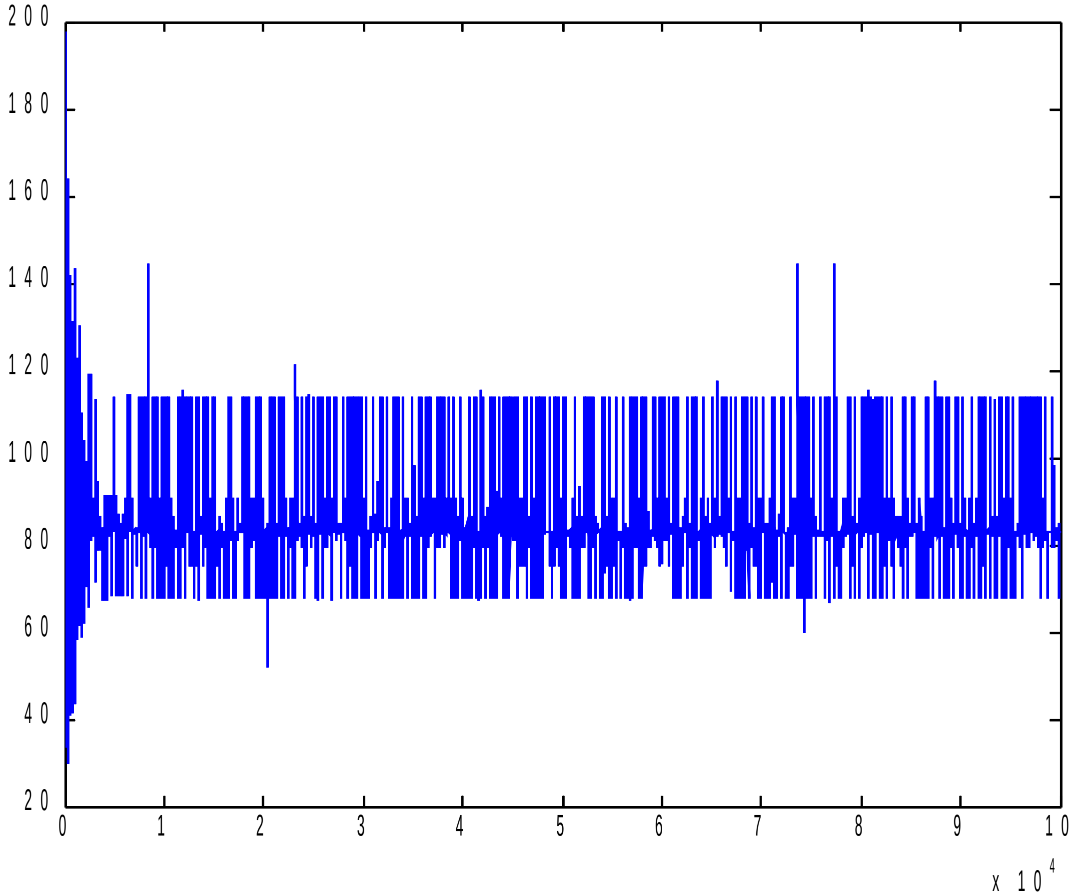


$chrom = 100$

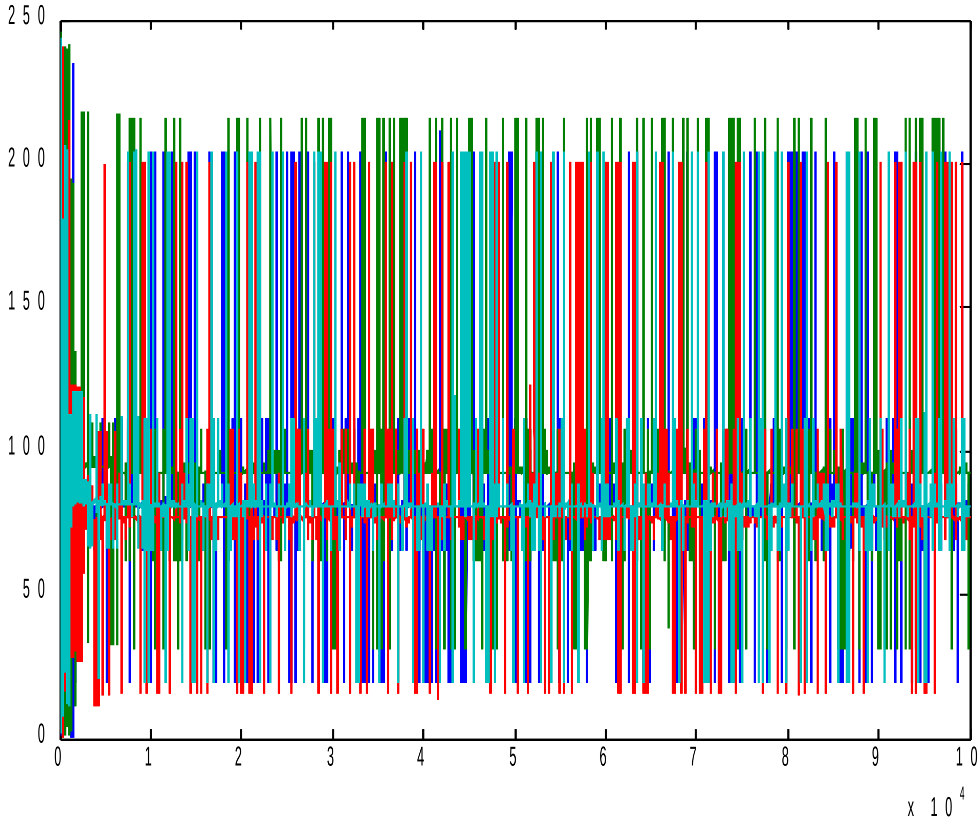
0	0	0	0	0	0	0	0	0,172	0,686	0,142	0	0	0	0	0	0	0	0	0	0



Βλέπουμε κι εδώ ότι, ενώ υπάρχει μια τάση βελτίωσης, καθώς αυξάνεται το μέγεθος του πληθυσμού, οι διαφορές δεν είναι σημαντικές. Και εδώ θα επιλέξουμε 50 χρωμοσώματα για την τελική εκτέλεση. Έχουμε λοιπόν, για $GArate = 50$, $l = 20$, $p_{cr} = 1$, $p_{mut} = 0.01$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 100.000 γενεές, μέση ποσότητα στην αγορά 83,0711 και τυπική απόκλιση 4,0927.



Για τις μεμονωμένες επιχειρήσεις έχουμε

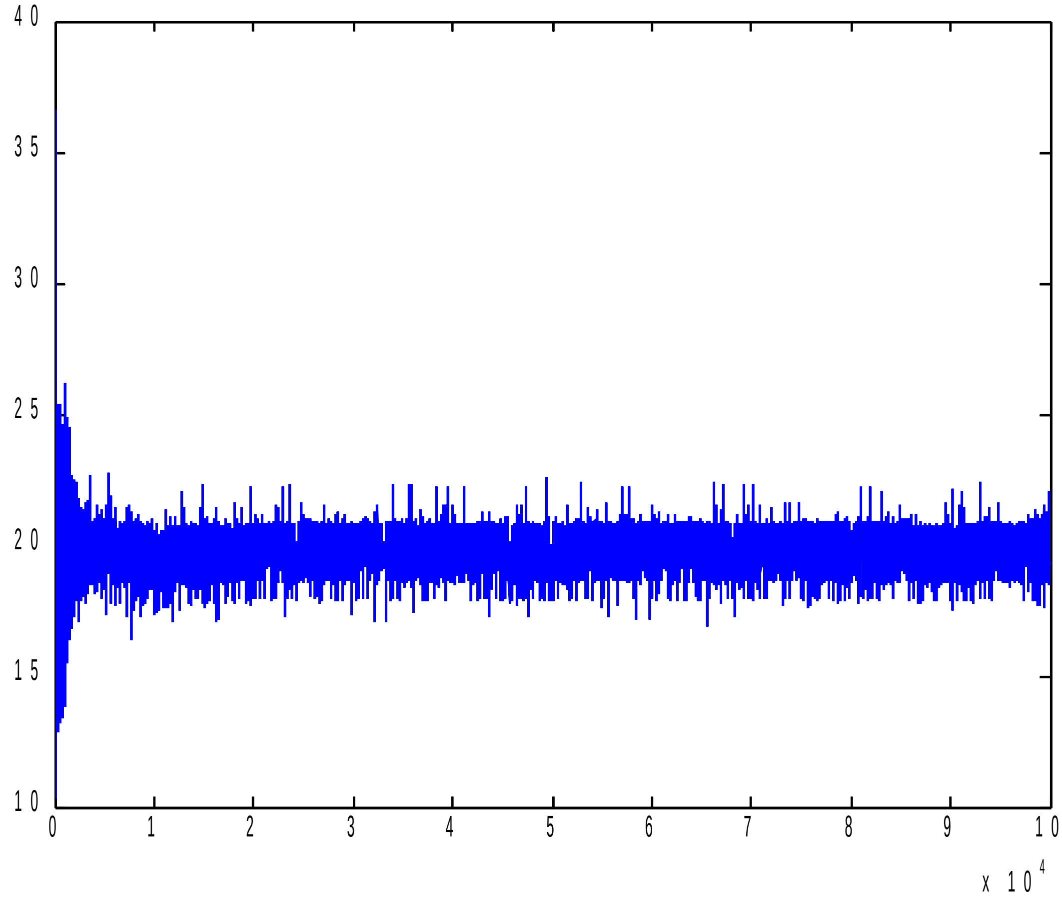


Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	81,2366	7,1881
2	92,7713	9,6418
3	76,9724	8,1094
4	81,3040	6,9604

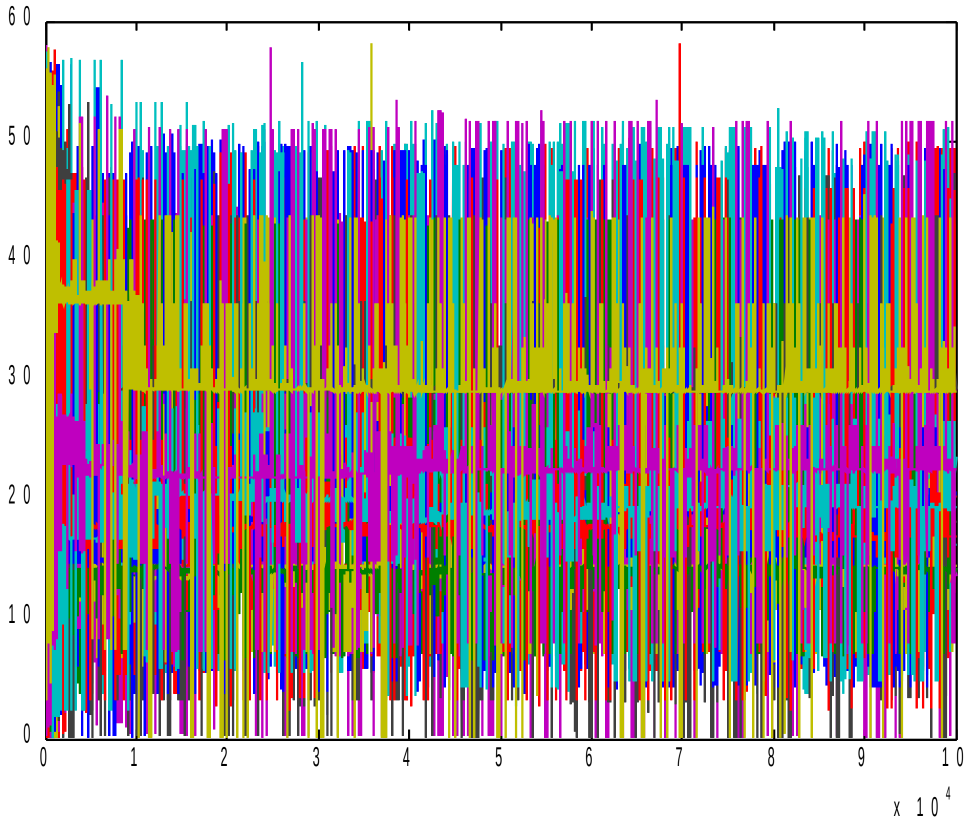
ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0,0285	0,049	0,9135	0,006	0,003	0	0	0	0	0	0	0	0	0

Για την περίπτωση των 20 επιχειρήσεων τώρα, με $p_{cr} = 1$, $l = 20$, $p_{mut} = 0{,}01$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 100.000 γενεές, έχουμε μέση ποσότητα στην αγορά 19,4040 και τυπική απόκλιση 0,5253



Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	20,4243	2,2586
2	14,1129	1,9388
3	20,0382	2,2268
4	19,9101	2,4193
5	29,2775	2,4056
6	14,4096	1,9426
7	17,8487	2,2765
8	20,5519	1,9299
9	14,4543	1,9554
10	14,3400	2,4959
11	22,8482	2,4549
12	14,1442	2,2169
13	14,3234	2,2774
14	29,1251	2,5354
15	18,5381	2,4472
16	14,1380	1,9355
17	17,8946	2,4555
18	19,3300	2,1789
19	22,4842	2,1271
20	29,8863	2,7351

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0	0,8375	0,1625	0	0	0	0	0	0	0	0	0

5.5.7. Εφαρμογή του Co-evolutionary programming στο γραμμικό υπόδειγμα

Εδώ, το *GArate* ισούται με τον αριθμό χρωμοσωμάτων στων πληθυσμό. Αυτό θα επηρρεάζει τη συμπεριφορά των αλγορίθμων για διαφορετικό αριθμό χρωμοσωμάτων. Επίσης ο αριθμός «επαναλήψεων», δηλαδή εκτελέσεων παιγνίων Cournot ισούται με το γινόμενο του αριθμού γενεών επί τον αριθμό χρωμοσωμάτων στον πληθυσμό. Έτσι για 10.000 γενεές πληθυσμών, και 50 χρωμοσώματα σε κάθε πληθυσμό, έχουμε 500.000 εκτελέσεις του βασικού βρόχου του αλγορίθμου (και συνεπώς 500.000 παίγνια).

Αρχικά μελετάμε την περίπτωση των $n = 4$ επιχειρήσεων. Διαπιστώσαμε ότι η σύγκλιση στην ισορροπία κατά Nash δεν επηρρεάζεται σημαντικά, από διαφορές στην τιμή του τελεστή διασταύρωσης. Για 1.000 γενεές, $l = 20$, $p_{mut} = 0{,}01$, και 50 χρωμοσώματα στους πληθυσμούς οι εκτιμήτριες του διανύσματος σταθερών συχνοτήτων ήταν:

$p_{cr} = 0{,}1$

0	0	0	0	0	0	0	0	0,047	0,898	0,05	0,005	0	0	0	0	0	0	0	0	0

$p_{cr} = 0{,}5$

0	0	0	0	0	0	0,09	0,256	0,646	0,003	0,005	0	0	0	0	0	0	0	0	0	0

$p_{cr} = 0{,}75$

0	0	0	0	0	0	0	0,135	0,822	0,037	0,006	0	0	0	0	0	0	0	0	0	0

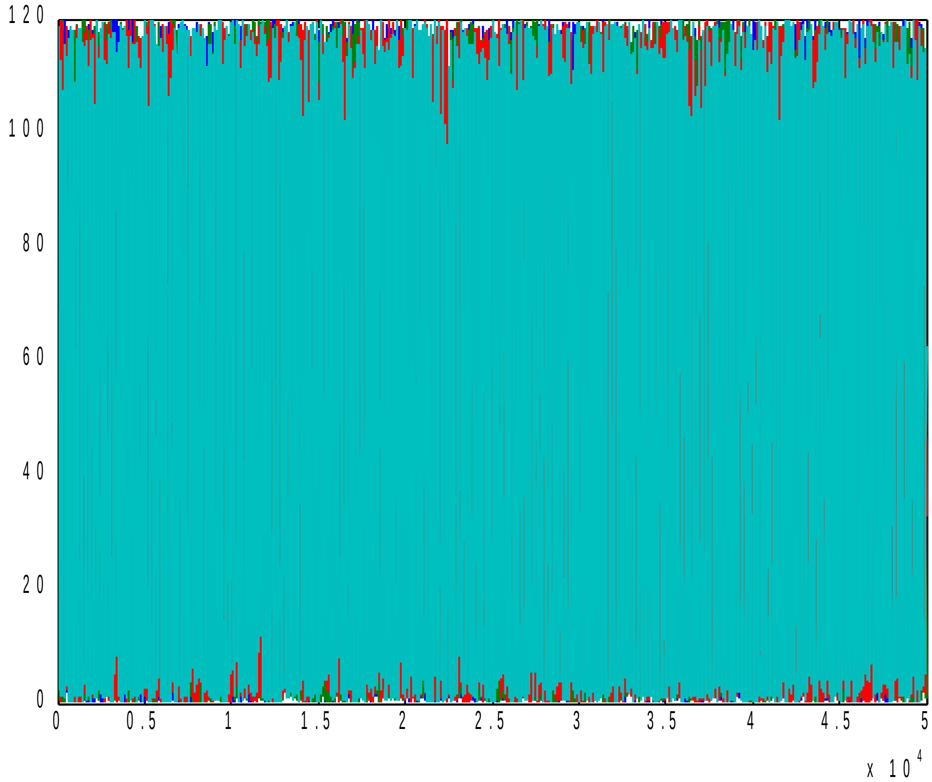
$p_{cr} = 1$

0	0	0	0	0	0	0	0	0,148	0,81	0,041	0,001	0	0	0	0	0	0	0	0	0

Στη συνέχεια επιλέξαμε $p_{cr} = 1$. Αντίθετα διαφορετικές πιθανότητες μετάλλαξης, μπορούν να επηρεάσουν το αποτέλεσμα. Για τις ίδιες τιμές με παραπάνω για τον αριθμό γενεών και χρωμοσωμάτων

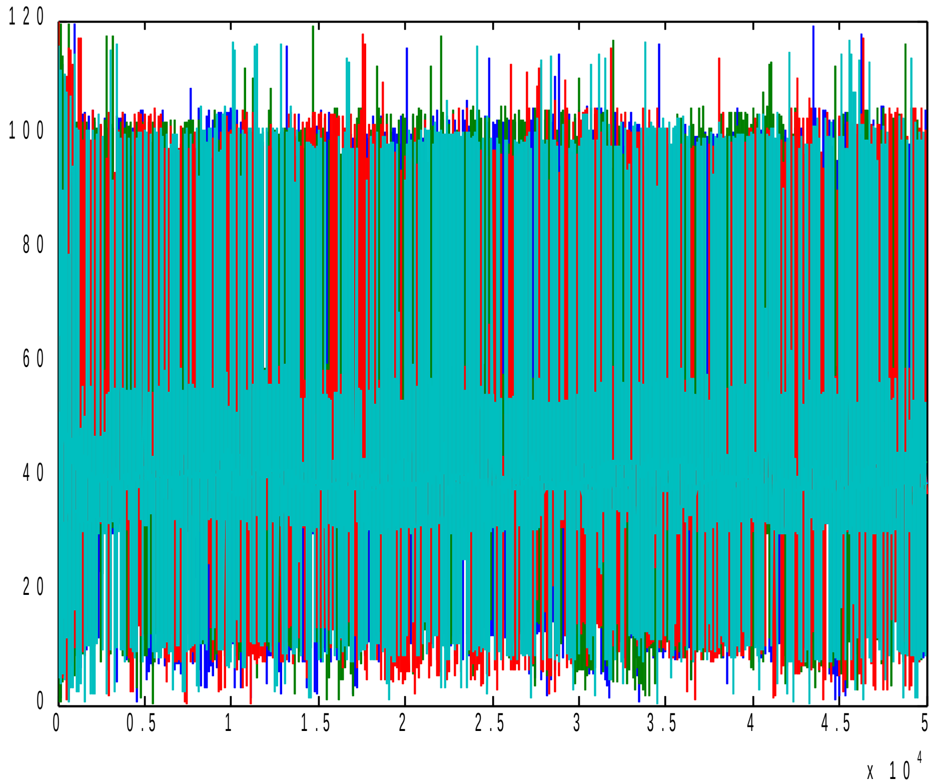
$p_{mut} = 0{,}1$

0	0	0	0	0	0	0	0	0	0,002	0,795	0,203	0	0	0	0	0	0	0	0	0



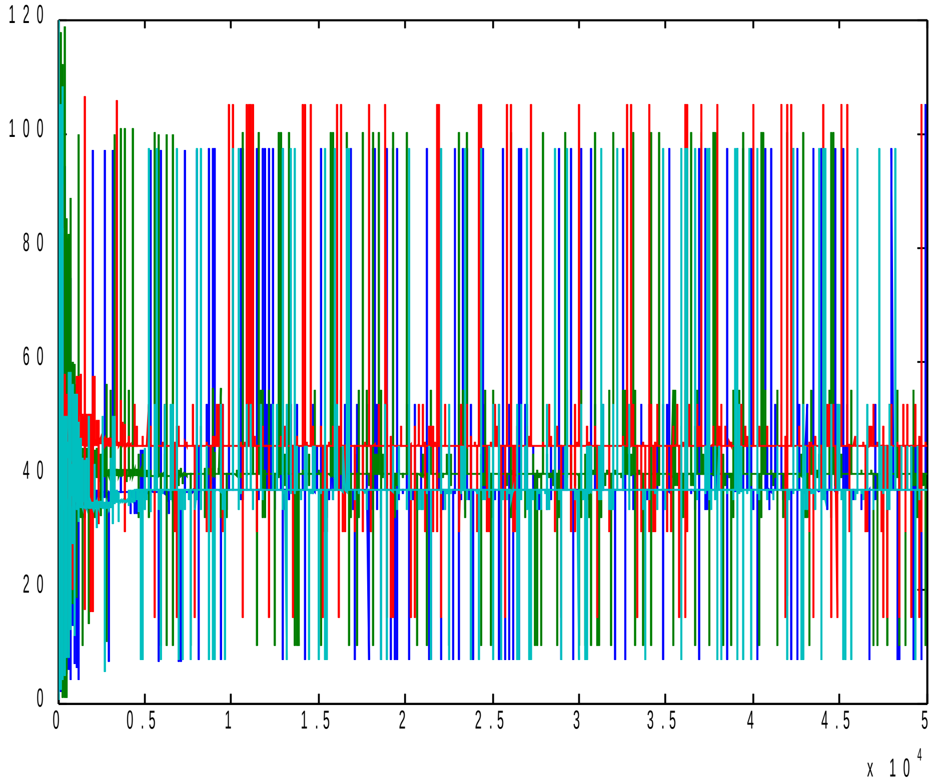
p_{mut} = 0,01

0	0	0	0	0	0	0	0,005	0,328	0,476	0,18	0,011	0	0	0	0	0	0	0	0	0



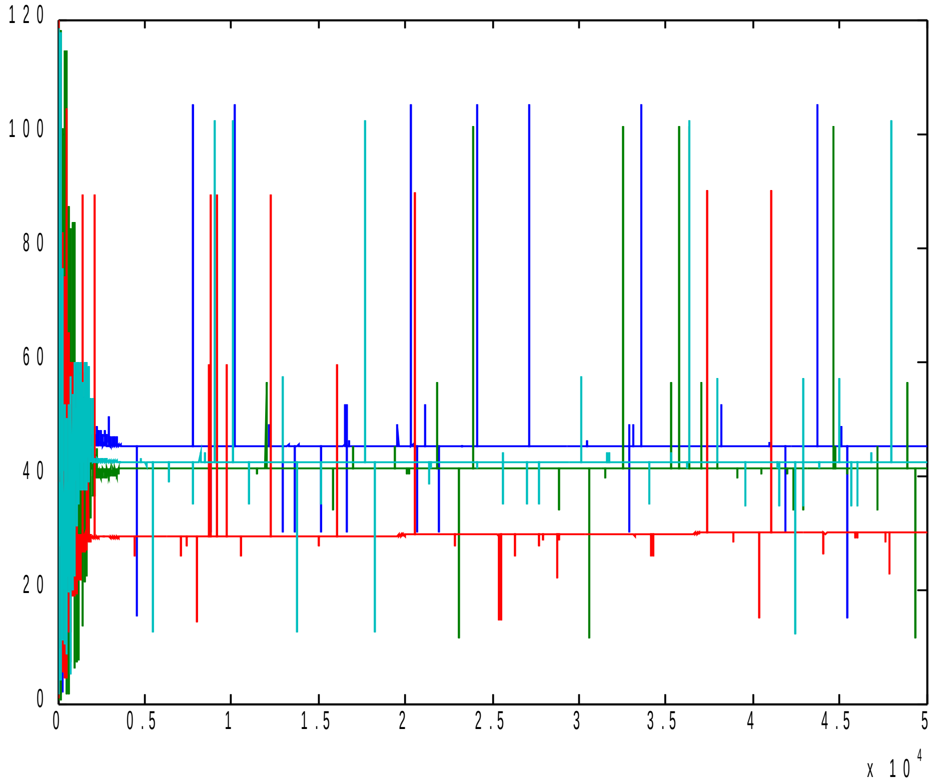
p_{mut} = 0,001

0	0	0	0	0	0	0	0	0,016	0,948	0,019	0,017	0	0	0	0	0	0	0	0	0



p_{mut} = 0,0001

0	0	0	0	0	0	0	0	0	0,437	0,47	0,093	0	0	0	0	0	0	0	0	0

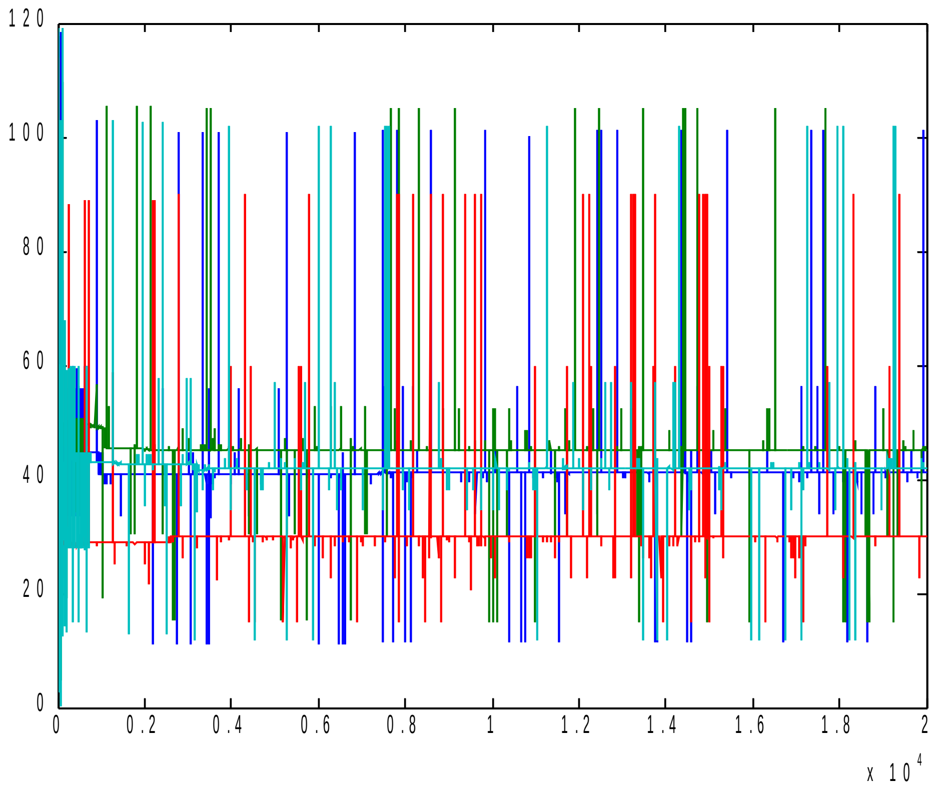


Επιλέγουμε $p_{mut} = 0{,}001$, αφού θεωρούμε ότι για $p_{mut} = 0{,}0001$ ο αλγόριθμος δυσκολεύεται να ξεφύγει από το τοπικό ελάχιστο στο οποίο βλέπουμε ότι βρίσκεται. Όπως και να έχει, οι διαφορές είναι μικρές.

Ο αριθμός των χρωμοσωμάτων στον πληθυσμό εδώ δεν παίζει σημαντικό ρόλο: για τις τιμές των παραμέτρων που προαναφέραμε

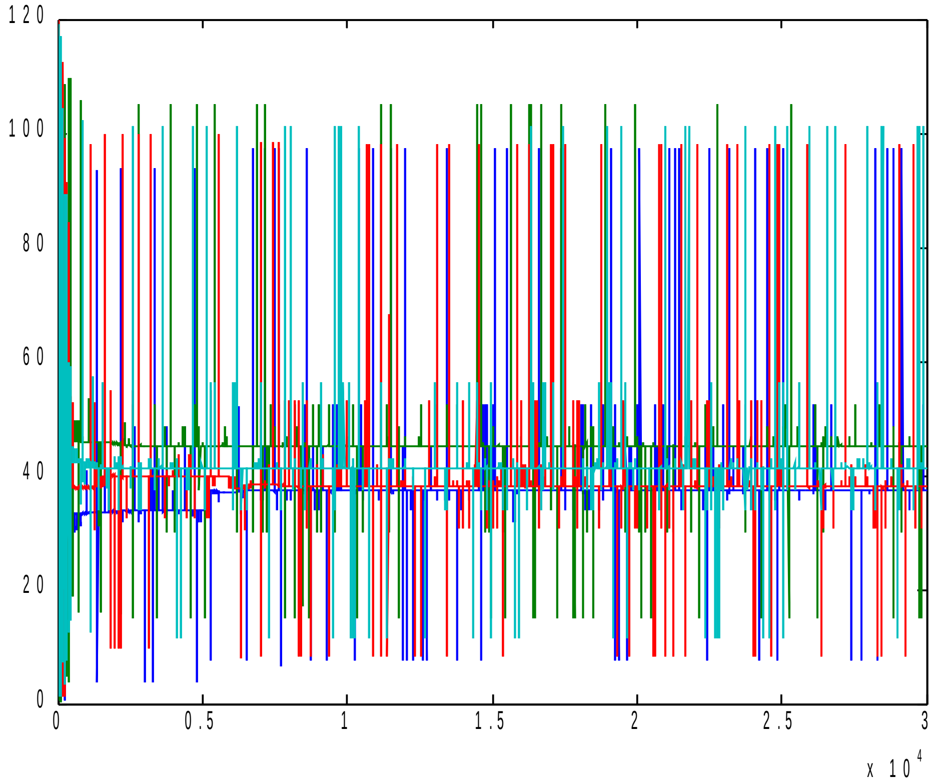
$chrom = 20$

0	0	0	0	0	0	0	0	0,014	0,872	0,084	0,03	0	0	0	0	0	0	0	0	0



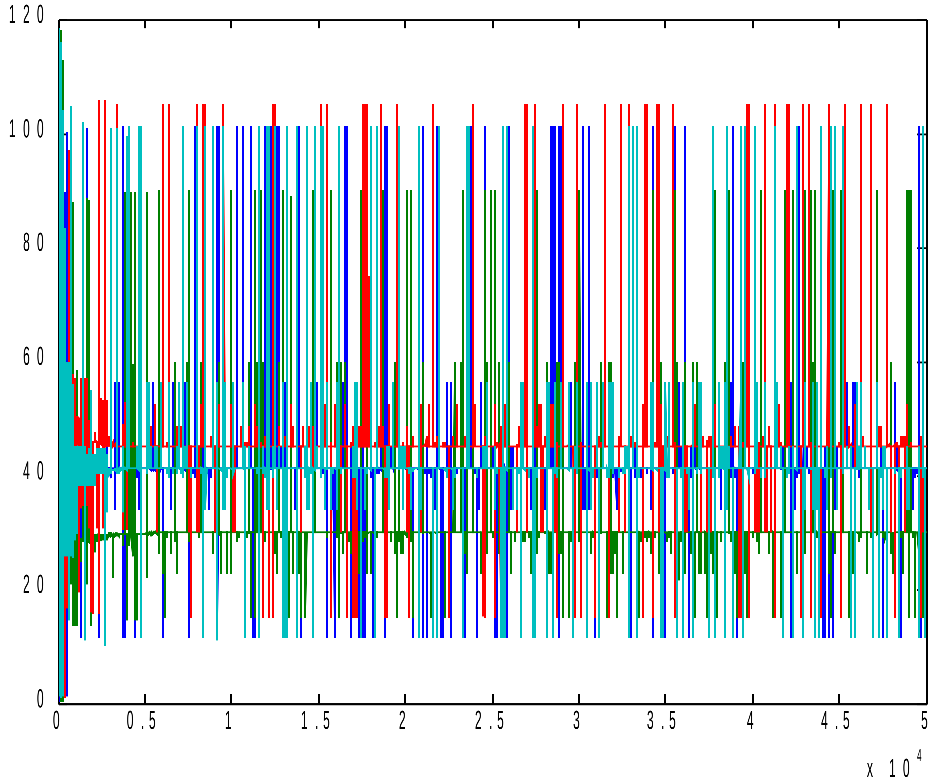
$chrom = 30$

0	0	0	0	0	0	0	0	0	0,66	0,337	0,003	0	0	0	0	0	0	0	0	0



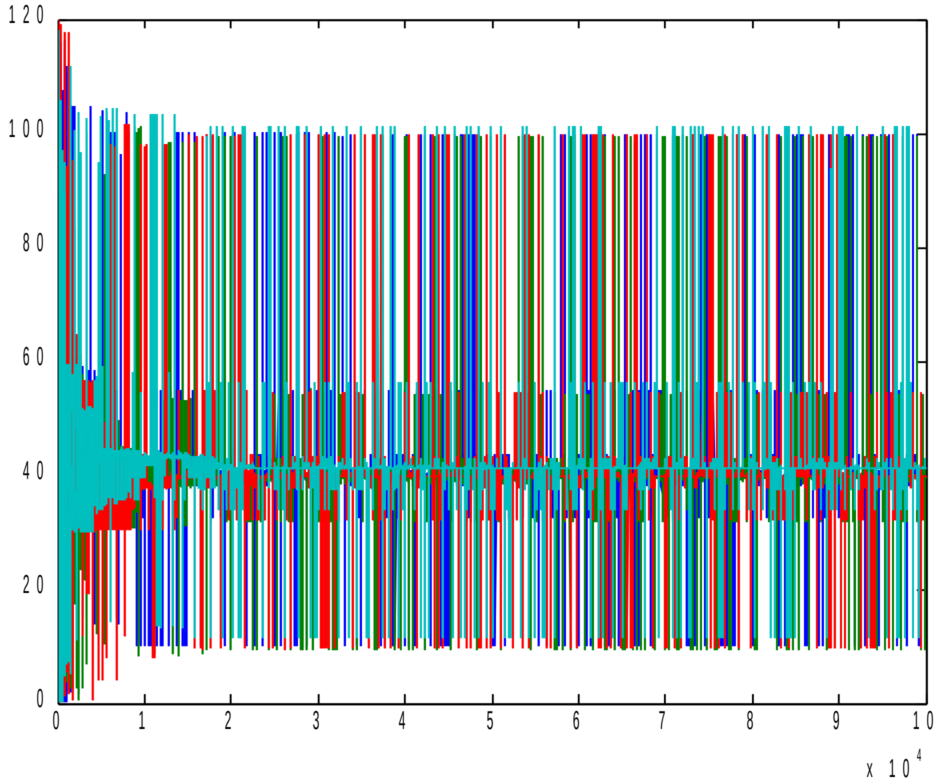
chrom = 50

0	0	0	0	0	0	0	0	0,003	0,961	0,034	0,002	0	0	0	0	0	0	0	0	0

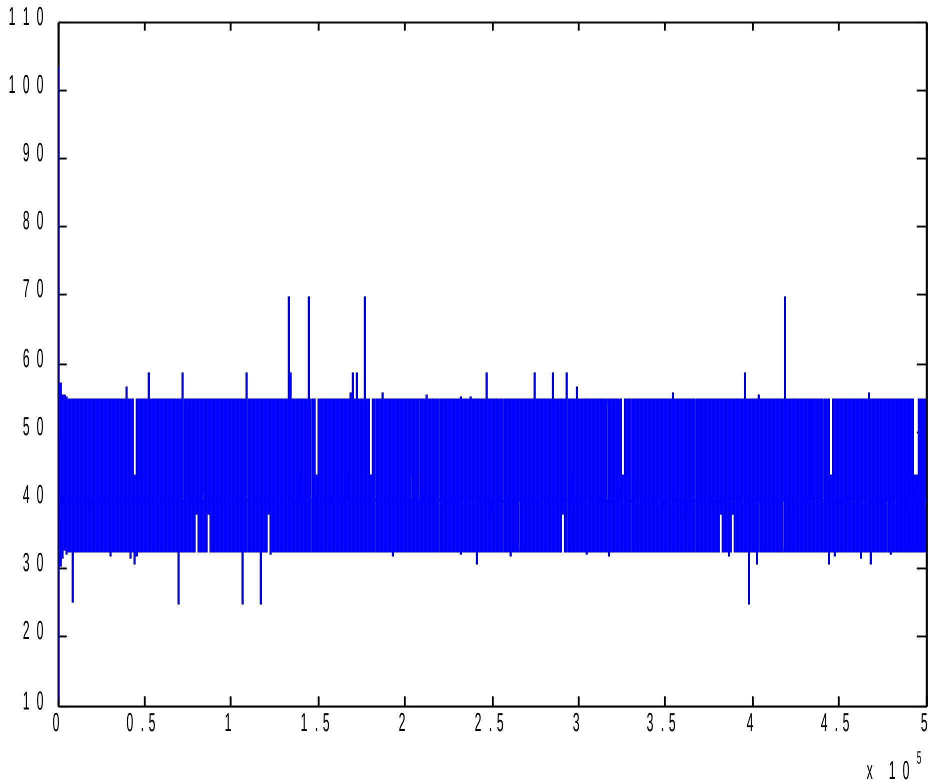


chrom = 100

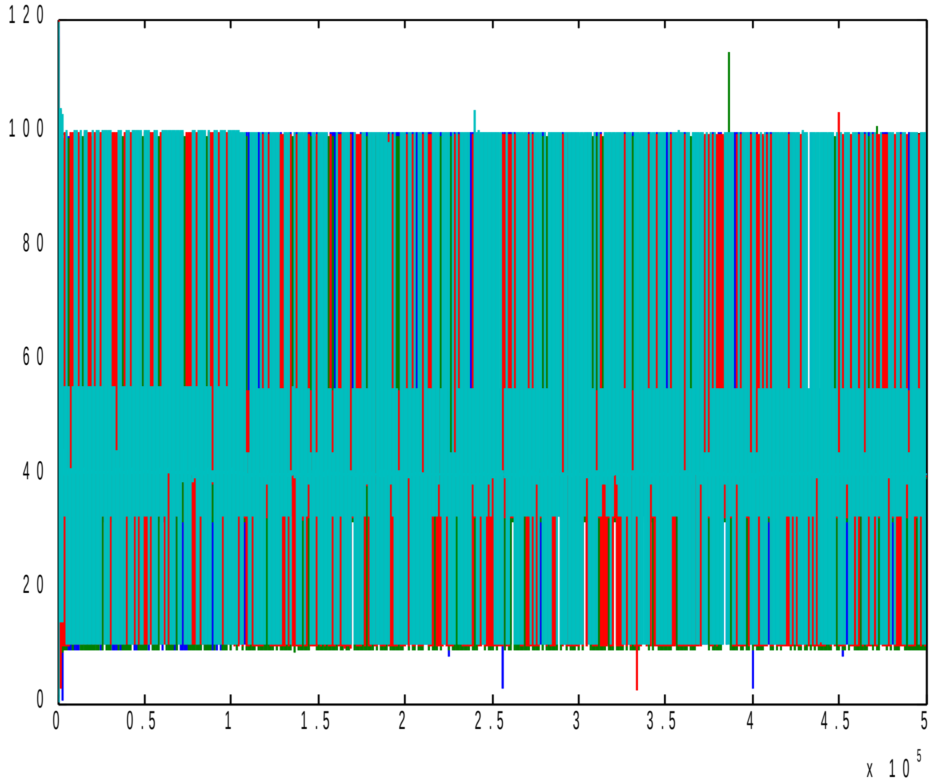
0	0	0	0	0	0	0	0	0,781	0,184	0,035	0	0	0	0	0	0	0	0	0



Τελικά, για $l = 20$, $p_{mut} = 0{,}001$, $p_{cr} = 1$, 50 χρωμοσώματα στον πληθυσμό και 10.000 γενεές έχουμε μέση ποσότητα στην αγορά 39,9285 και τυπική απόκλιση 1,2113



Για τις μεμονωμένες επιχειρήσεις



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,5080	2,3581
2	40,3854	2,4417
3	41,2803	2,4152
4	37,5403	2,4023

Εδώ έχουμε μια αρκετά καλή σύγκλιση των μέσων ποσοτήτων των 3 πρώτων επιχειρήσεων, κοντά στο $q^W = 40$. Ωστόσο σε κανένα από τα 500.000 παίγνια που εκτελέστηκαν, δεν είχαμε $q_i = 40 \forall i$ και τους 4 παίκτες δηλαδή να παίζουν την ποσότητα 40, οπότε θα βρισκόμασταν στο ΝΕ.

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

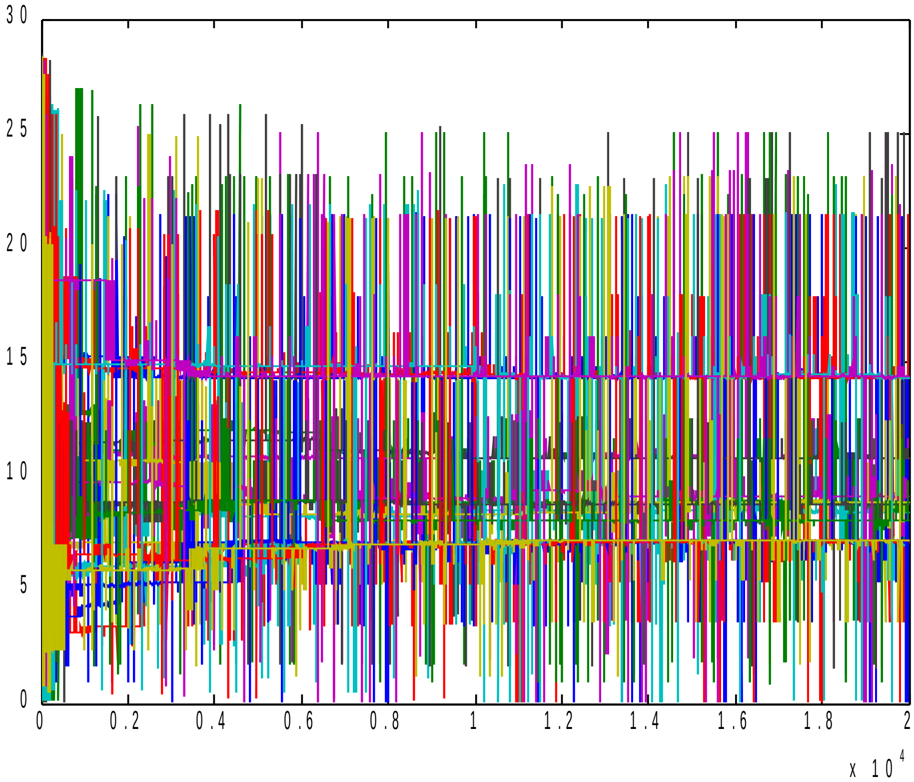
0	0	0	0	0	0	0	0,0039	0,9822	0,0091	0,0048	0	0	0	0	0	0	0	0	0	0

Το διάνυσμα αντικατοπτρίζει την κατάσταση καλύτερα. Βλέπουμε ότι σε καμμία από τις 10.000 γενεές, η μέση απόσταση των χρωμοσωμάτων των πληθυσμών δεν ήταν μικρότερη από 7 bits, από το χρωμόσωμα 0101...01 που αντιστοιχεί (αφού $q_{\max} = 3 * q_w$ πάντα) στην ισορροπία κατά Nash.

Για την περίπτωση των 20 επιχειρήσεων τώρα, οπότε $q^N = 9{,}5238$, για $l = 20$, $p_{cr} = 1$, $p_{mut} = 0{,}001$ και 1.000 γενεές δοκιμάζουμε πρώτα διαφορετικούς πληθυσμούς.

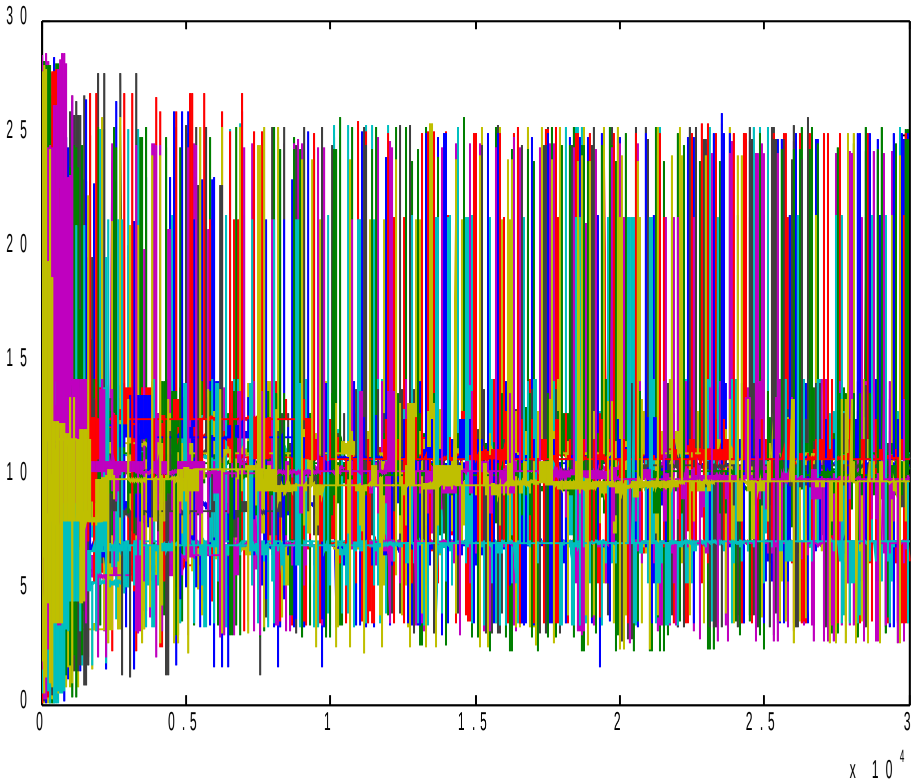
$chrom = 20$

0	0	0	0	0	0	0	0	0	0	0,665	0,335	0	0	0	0	0	0	0	0	0



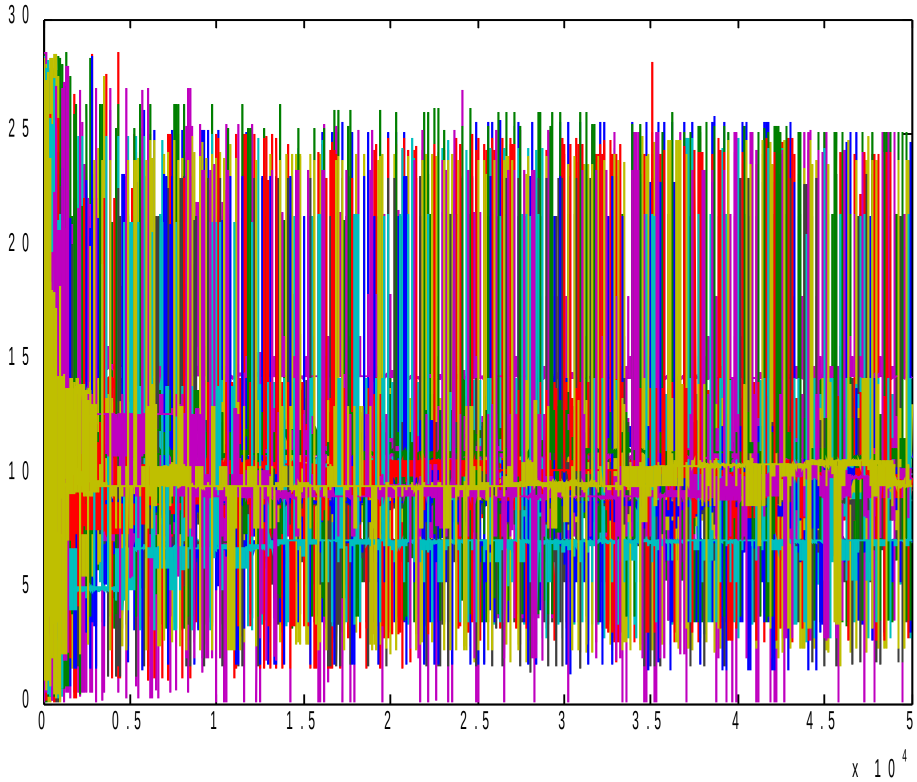
chrom = 30

0	0	0	0	0	0	0	0	0	0	0,407	0,571	0,022	0	0	0	0	0	0	0	0



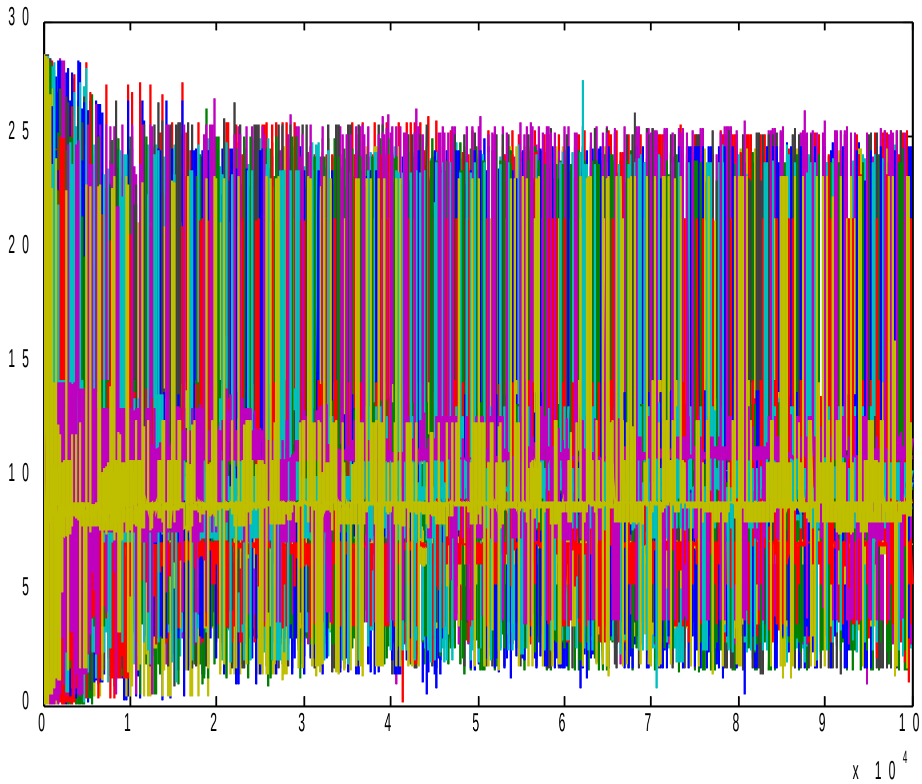
chrom = 50

0	0	0	0	0	0	0	0	0	0,651	0,349	0	0	0	0	0	0	0	0	0	0

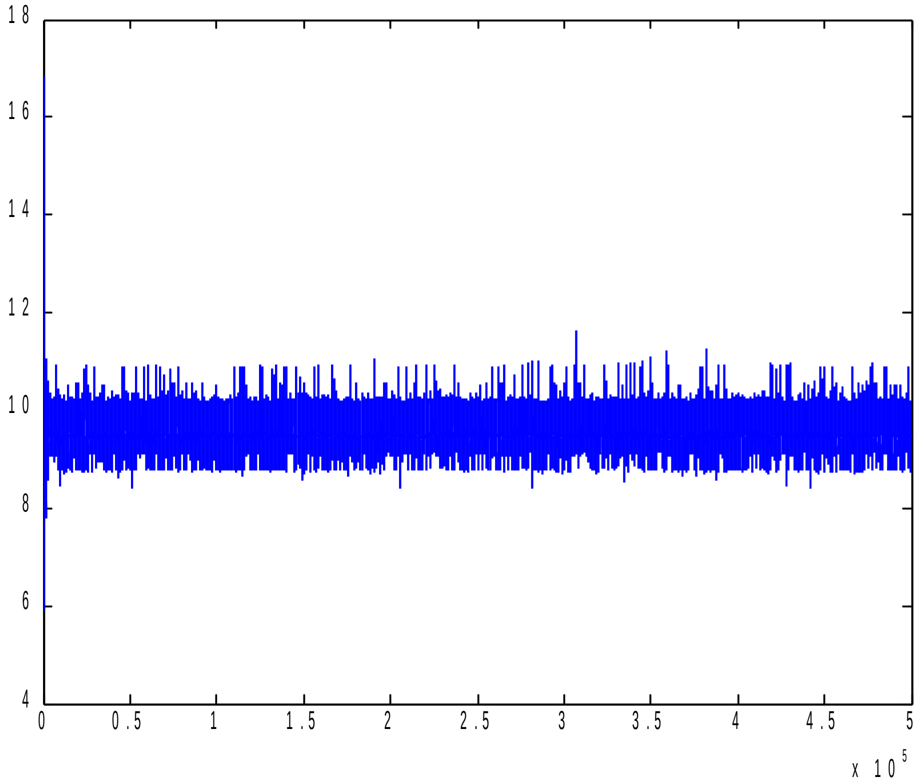


chrom = 100

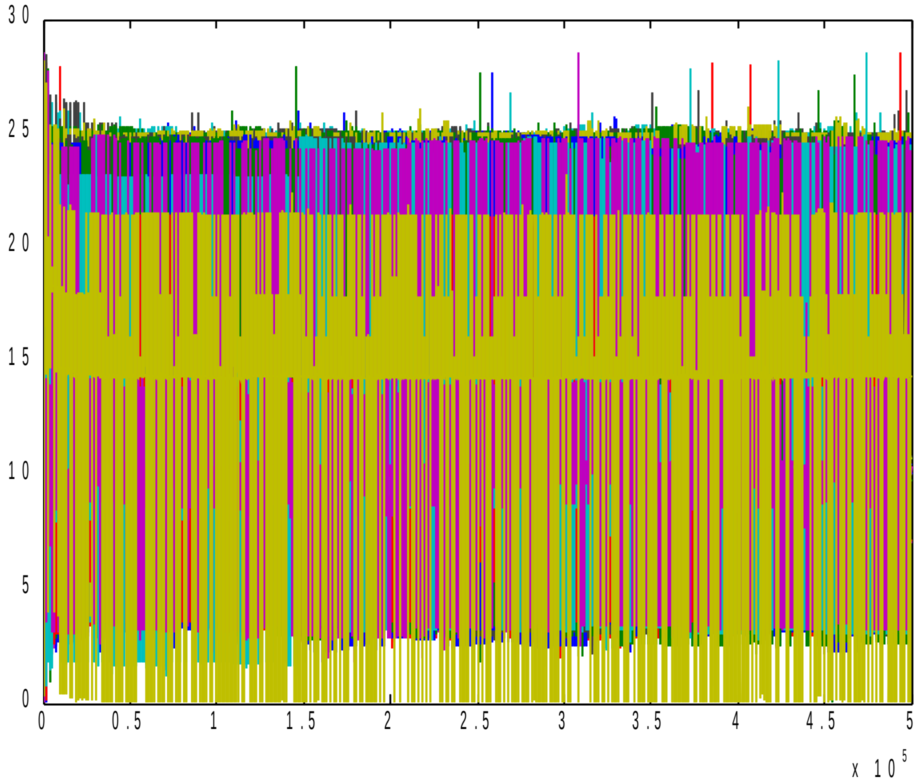
0	0	0	0	0	0	0	0	0	0,269	0,731	0	0	0	0	0	0	0	0	0	0



Επιλέγουμε τα 50 χρωμοσώματα. Έτσι για $p_{mut} = 0{,}001$, $p_{cr} = 1$, 50 χρωμοσώματα στους πληθυσμούς και 10.000 γενεές έχουμε μέση ποσότητα στην αγορά 9,4970 και τυπική απόκλιση 0,1444



Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	10,8288	0,6388
2	7,0947	0,6154
3	10,2057	0,6430
4	10,8813	0,6424
5	7,1261	0,5880
6	7,1344	0,5882
7	10,9166	0,7001
8	10,2031	0,7064
9	10,8485	0,6312
10	7,1344	0,5903
11	7,1326	0,6034
12	7,0856	0,6654
13	10,8842	0,6241
14	10,3887	0,6551
15	10,2595	0,7706
16	10,0809	0,6934
17	7,0950	0,6528
18	9,8799	0,8804
19	10,3597	0,7085
20	14,4003	0,8044

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

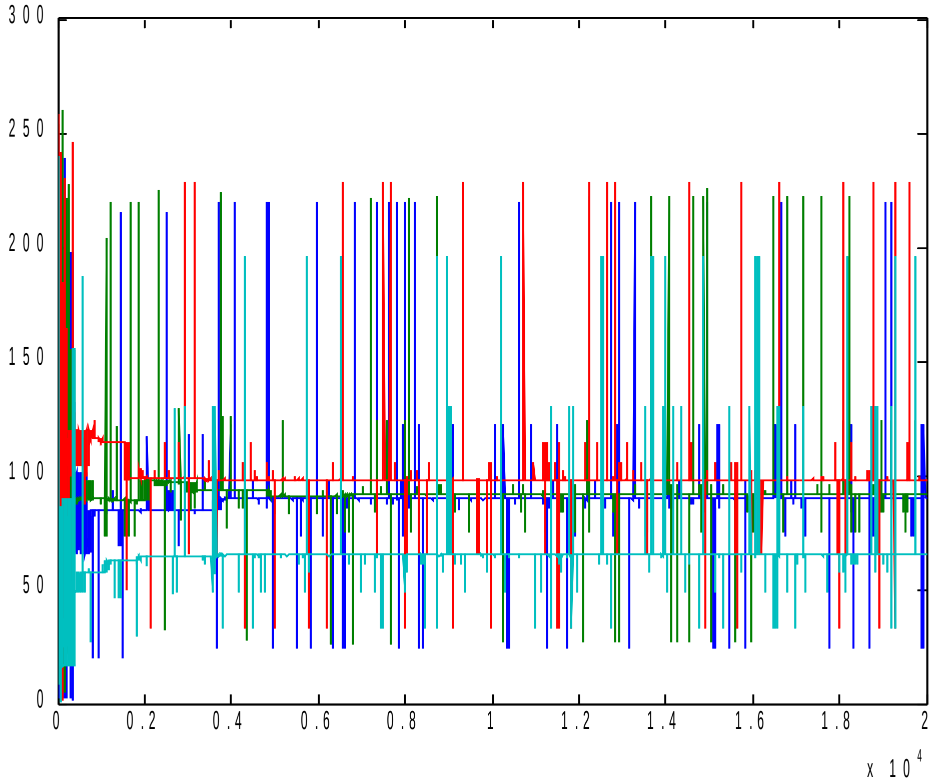
0	0	0	0	0	0	0	0	0	0,0393	0,9004	0,0603	0	0	0	0	0	0	0	0

5.5.8Εφαρμογή του co-evolutionary programming στο πολυωνυμικό υπόδειγμα

Εφαρμόζουμε το co-evolutionary programming στο πολυωνυμικό υπόδειγμα (βλ. ενότητα 3). Για την περίπτωση των 4 επιχειρήσεων και , $p_{mut} = 0{,}001$, $p_{cr} = 1$και αριθμό γενεών 1.000, έχουμε για διαφορετικούς πληθυσμούς:

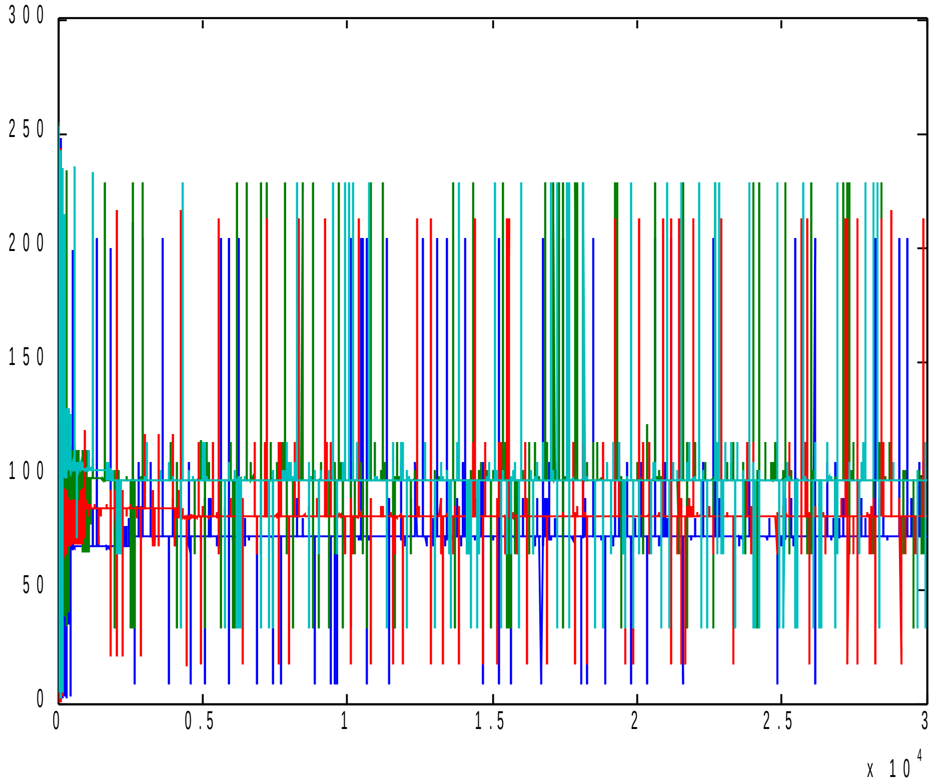
$chrom = 20$

0	0	0	0	0	0	0	0	0,009	0,83	0,161	0	0	0	0	0	0	0	0	0



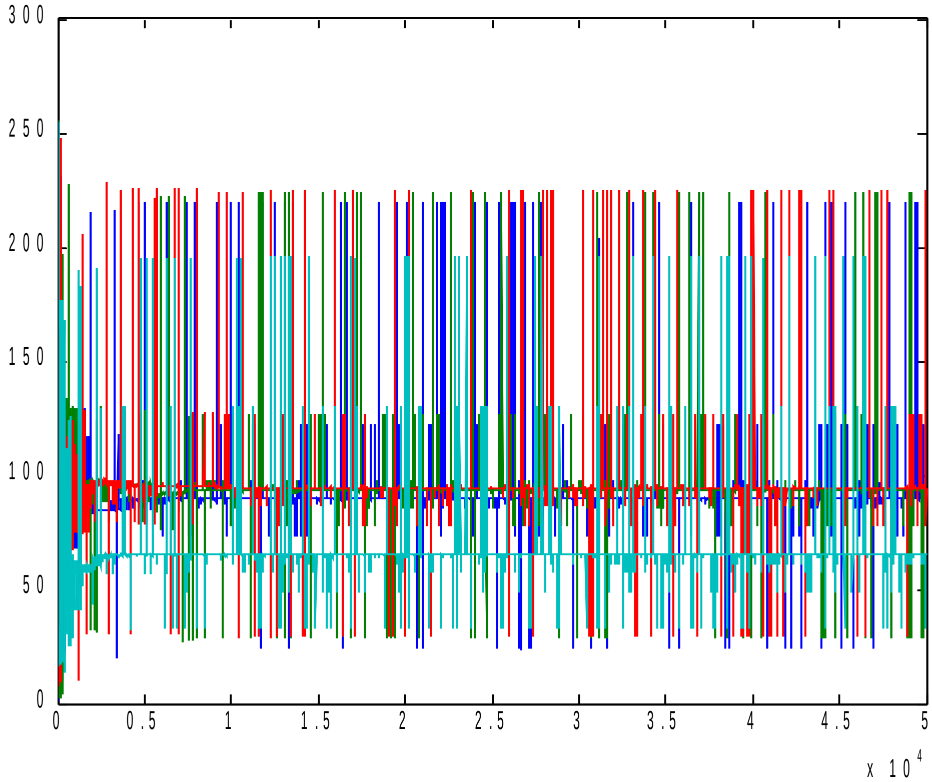
chrom = 30

0	0	0	0	0	0	0	0	0,105	0,126	0,768	0,001	0	0	0	0	0	0	0	0	0



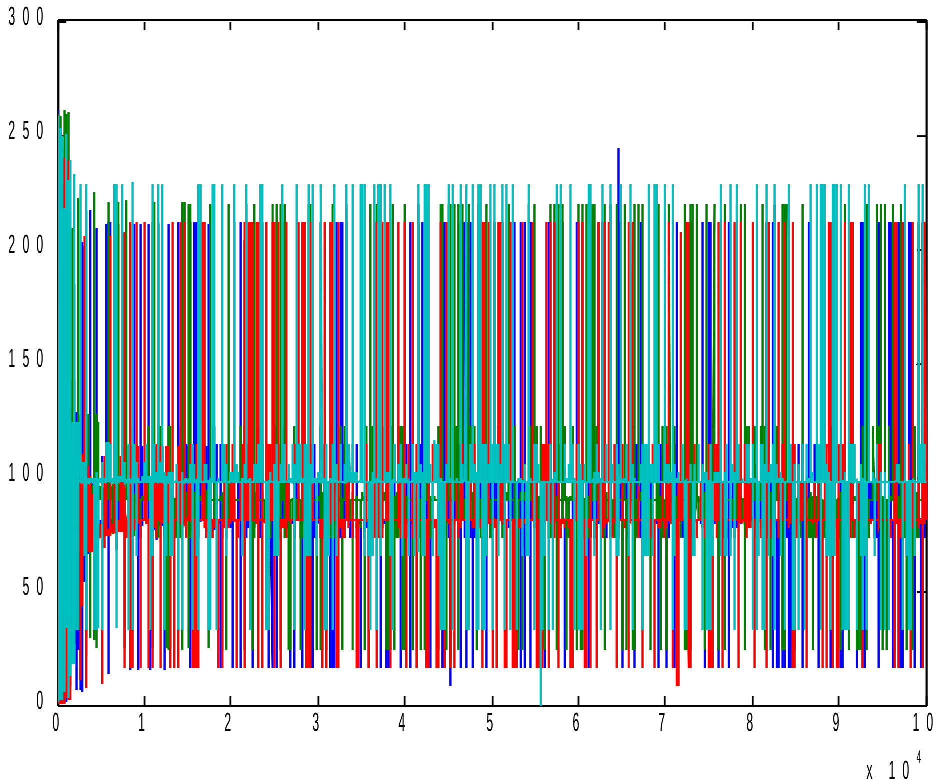
chrom = 50

0	0	0	0	0	0	0	0	0,042	0,927	0,031	0	0	0	0	0	0	0	0	0	0

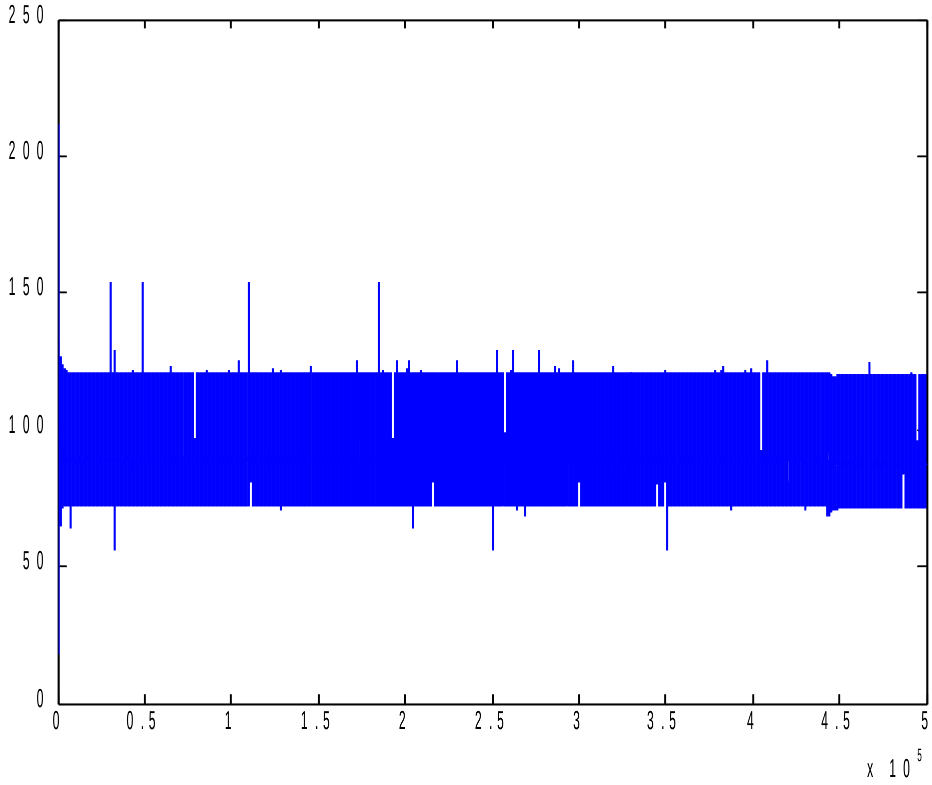


chrom = 100

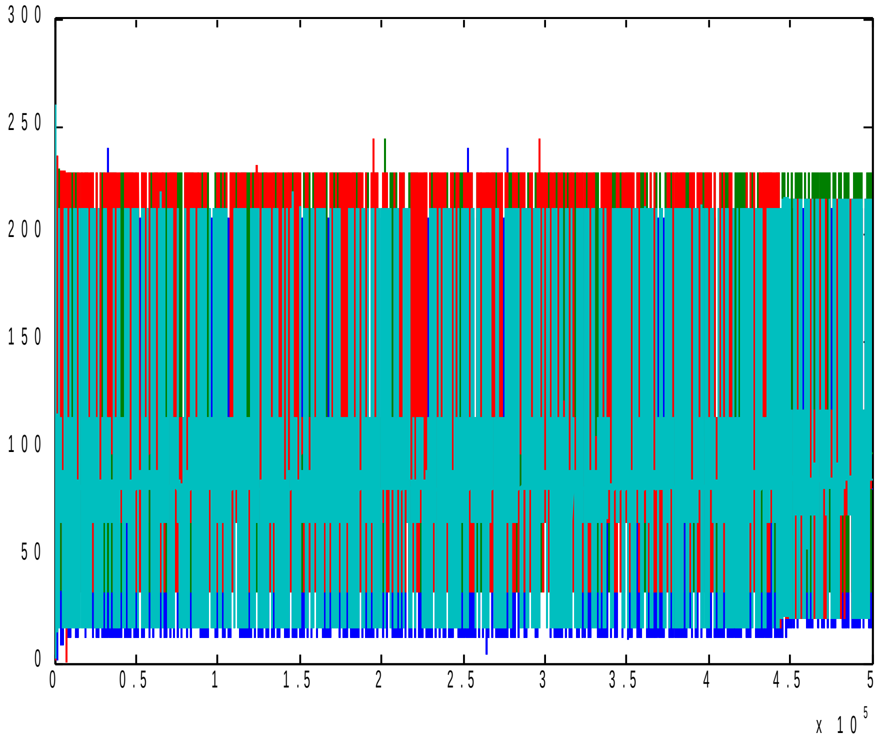
0	0	0	0	0	0	0	0	0	0,165	0,834	0,001	0	0	0	0	0	0	0	0	0



Εμείς θα επιλέξουμε 50 χρωμοσώματα για την τελική εκτέλεση. Έχουμε λοιπόν, για $p_{cr} = 1$, $p_{mut} = 0{,}001$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 10.000 γενεές, μέση ποσότητα στην αγορά 88,5472 και τυπική απόκλιση 2,6838.



Για τις μεμονωμένες επιχειρήσεις έχουμε

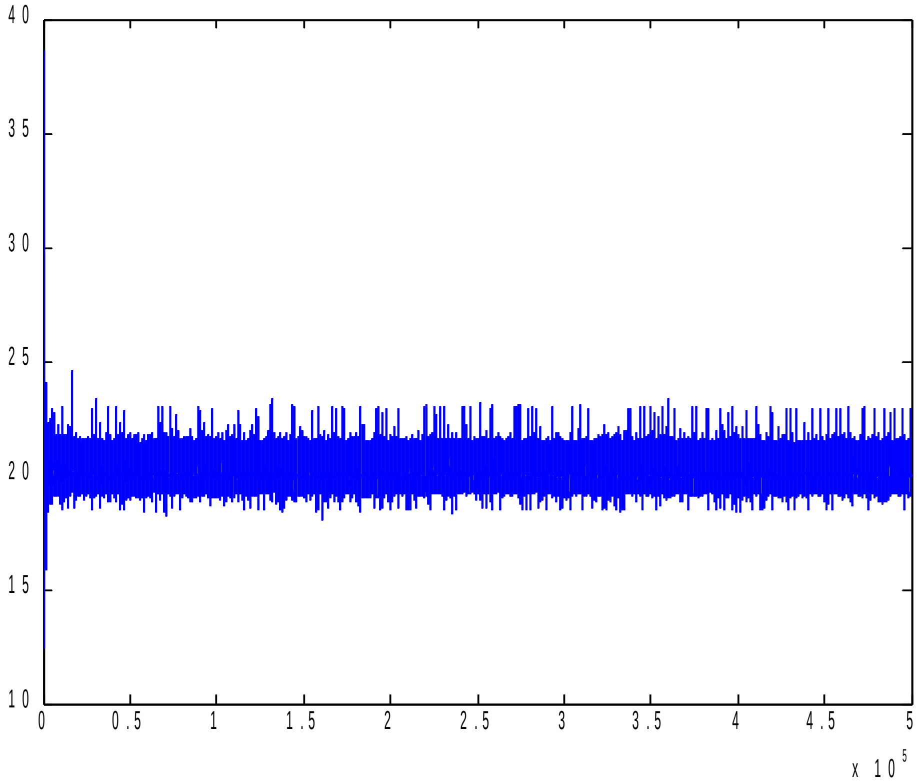


Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	77,6752	7,8282
2	97,8773	7,9961
3	93,9287	7,4739
4	93,9933	7,3539

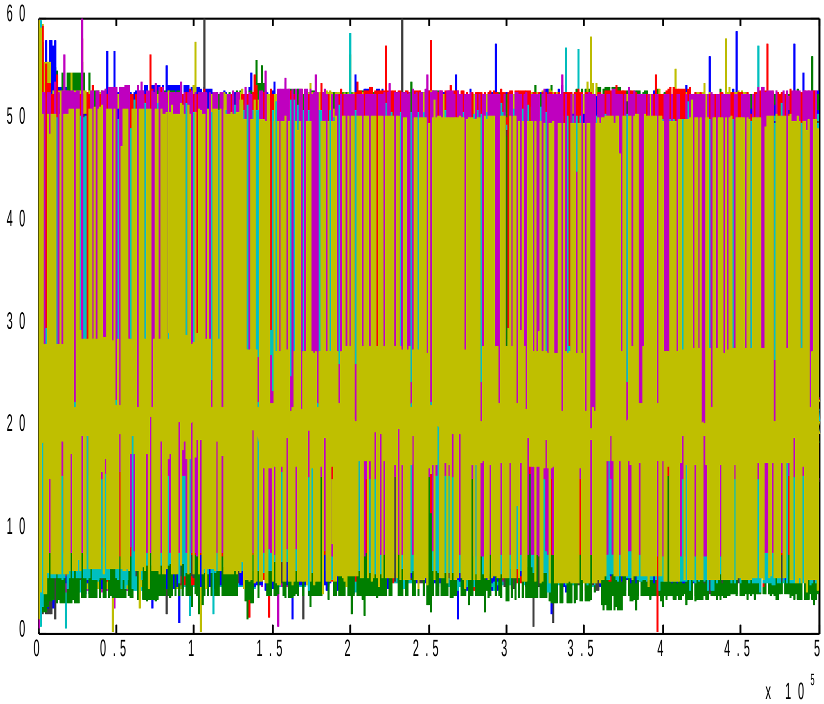
Σε κανένα από τα 500.000 παίγνια της προσομοίωσης δεν είχαμε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0,0025	0,1178	0,867	0,0127	0	0	0	0	0	0	0	0	0	0

Για την περίπτωση των 20 επιχειρήσεων τώρα, με $p_{cr} = 1$, $p_{mut} = 0{,}001$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 10.000 γενεές, έχουμε μέση ποσότητα στην αγορά 20,0092 και τυπική απόκλιση 0,3113



Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	22,8016	1,5220
2	22,8723	1,3621
3	14,8354	1,4708
4	20,1479	1,4965
5	20,0949	1,3198
6	20,3078	1,7863
7	14,9064	1,3603
8	20,4517	1,7354
9	20,3376	2,0025
10	20,1946	1,3377
11	20,0655	1,3565
12	14,9158	1,3384
13	22,7163	1,4119
14	20,2217	1,6037
15	20,2067	1,4574
16	18,4827	1,3296
17	22,7832	1,3638
18	20,6789	1,5809
19	22,7680	1,2506
20	20,3951	1,3627

Σε κανένα από τα 500.000 παίγνια δεν είχαμε ισορροπία κατά Nash, ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

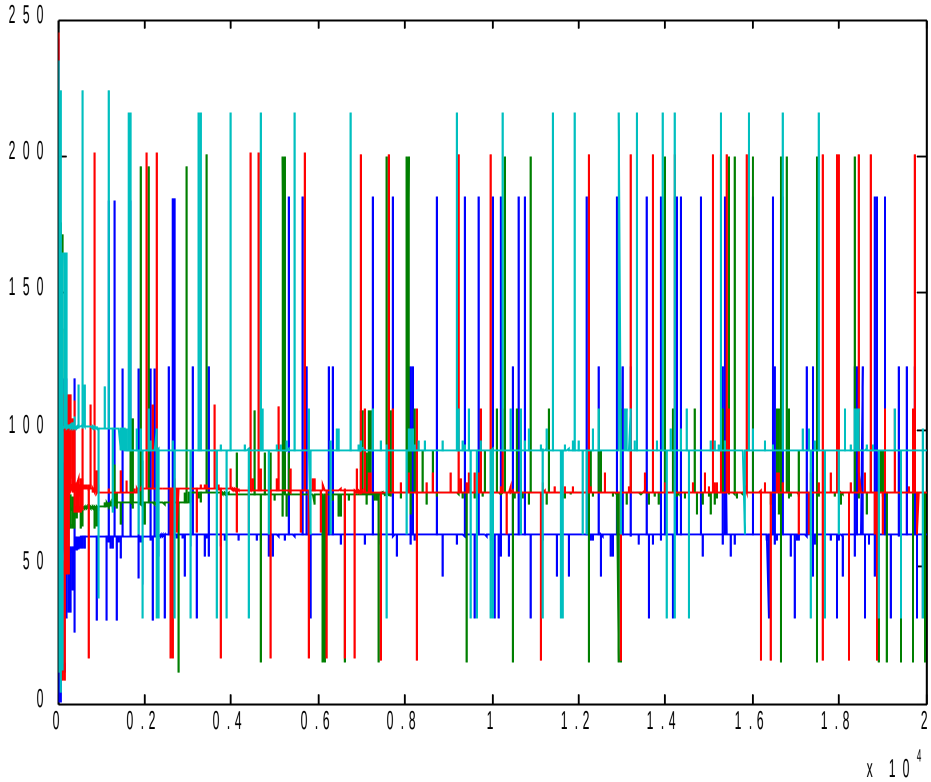
0	0	0	0	0	0	0	0	0,051	0,8821	0,0667	0,0002	0	0	0	0	0	0	0	0	0

5.5.9. Εφαρμογή του co-evolutionary programming στο υπόδειγμα με ρητό εκθέτη

Εφαρμόζουμε τον αλγόριθμο του co-evolutionary programming στο υπόδειγμα με ρητό εκθέτη (βλ. ενότητα 3). Για την περίπτωση των 4 επιχειρήσεων ($q^N = 82{,}2143$) και $p_{mut} = 0{,}01$, $p_{cr} = 1$ και αριθμό γενεών 1.000, έχουμε για διαφορετικούς πληθυσμούς:

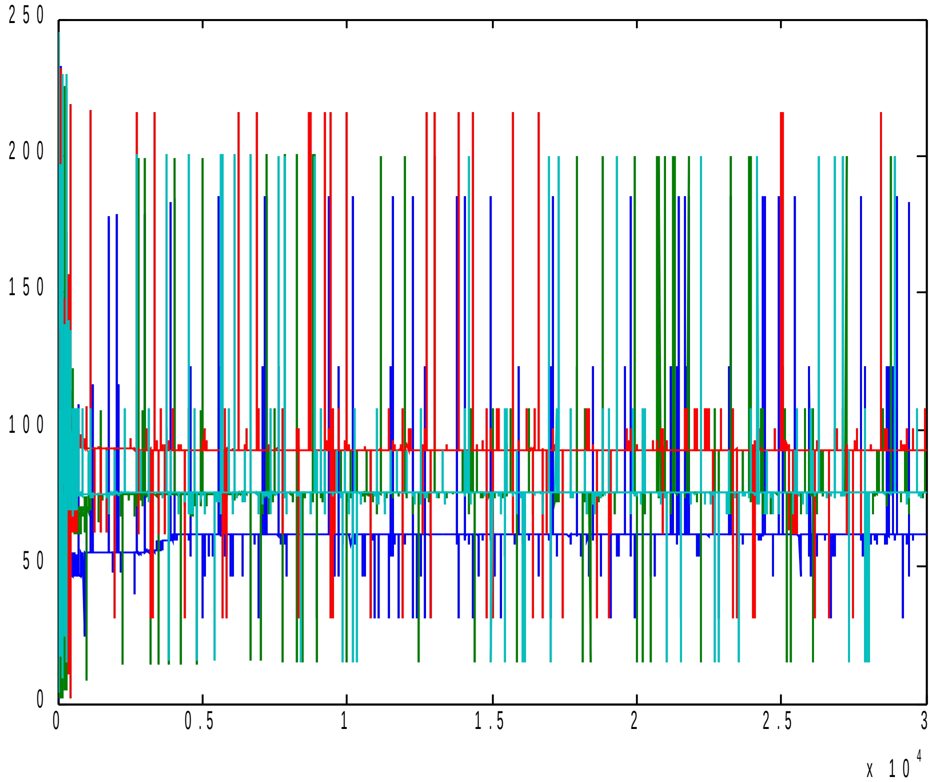
$chrom = 20$

0	0	0	0	0	0	0	0	0	0,532	0,353	0,102	0,013	0	0	0	0	0	0	0	0



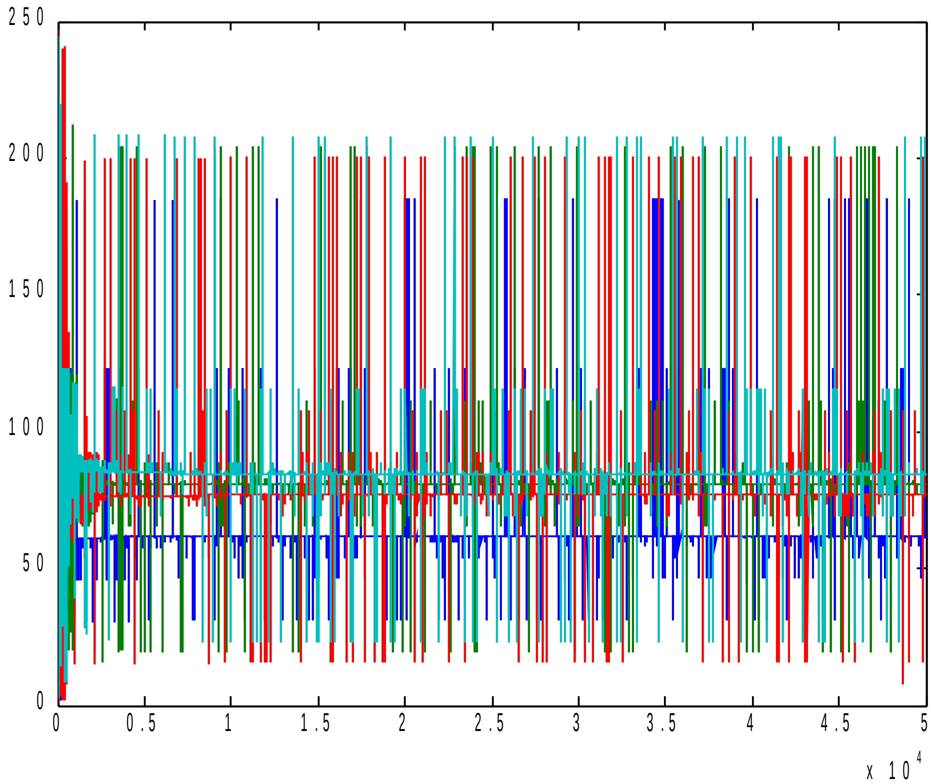
$chrom = 30$

0	0	0	0	0	0	0	0	0	0,014	0,664	0,322	0	0	0	0	0	0	0	0	0



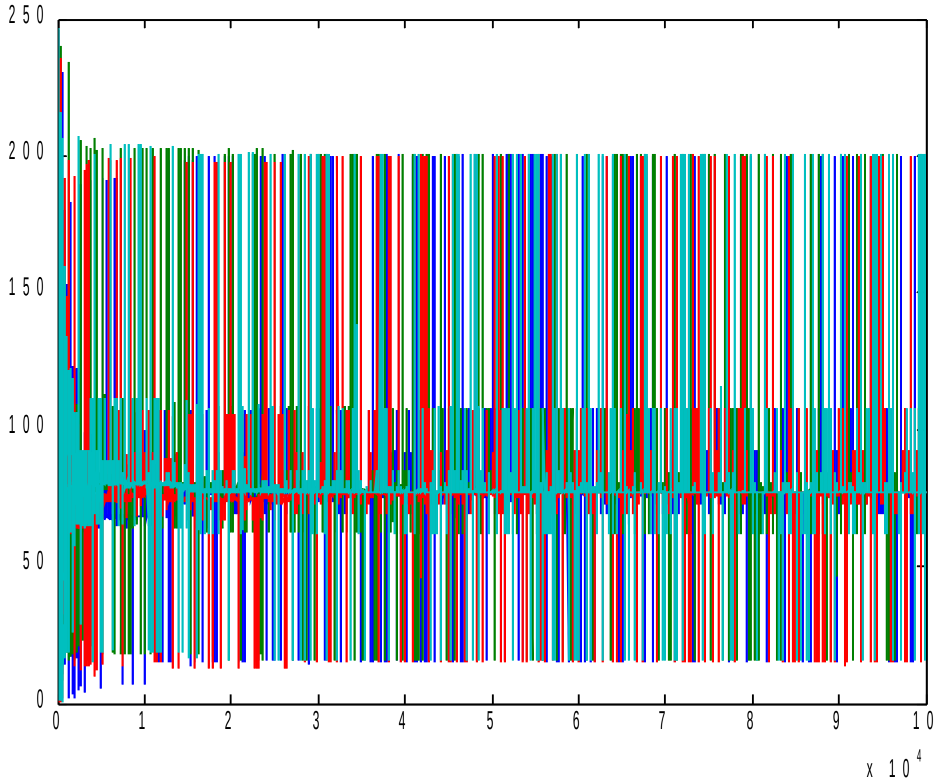
chrom = 50

0	0	0	0	0	0	0	0	0	0,902	0,072	0,026	0	0	0	0	0	0	0	0	0

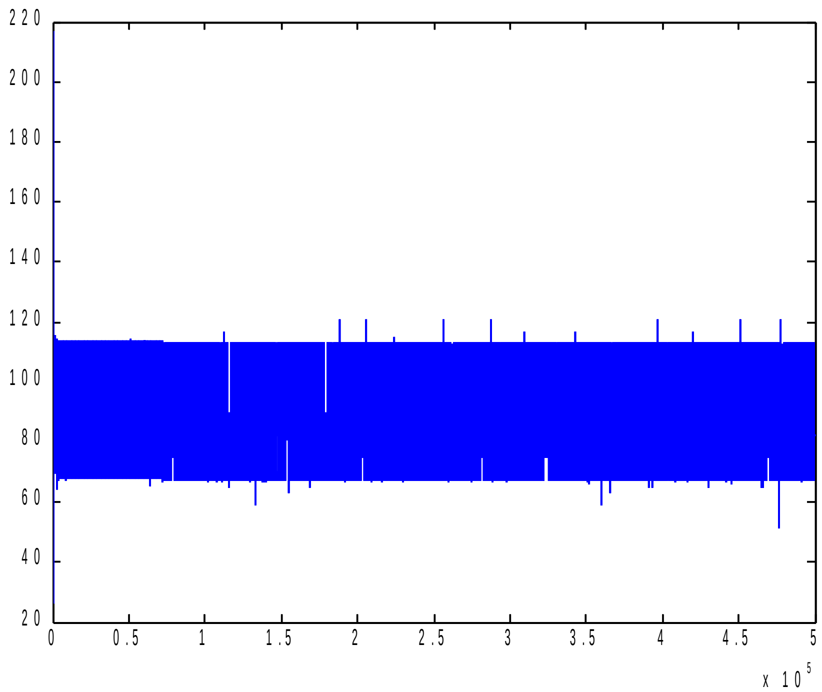


chrom = 100

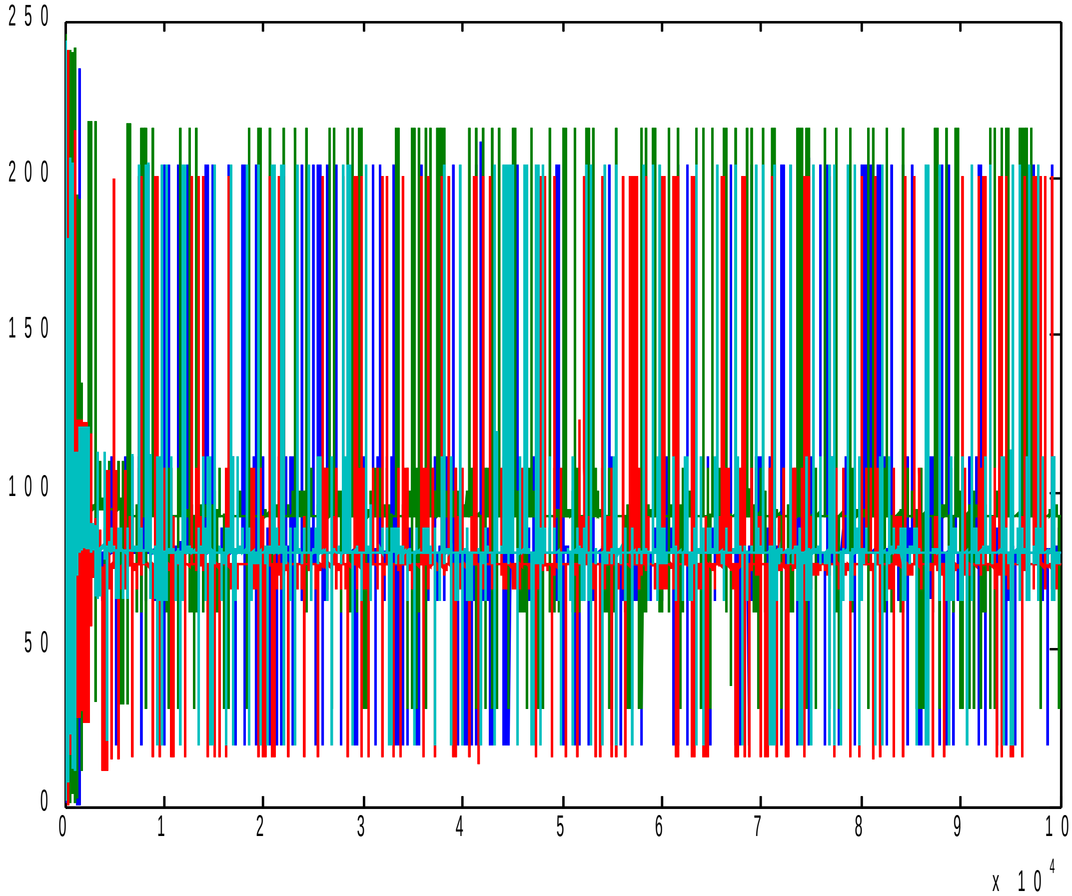
0	0	0	0	0	0	0	0	0,202	0,785	0,013	0	0	0	0	0	0	0	0	0	0



Βλέπουμε κι εδώ ότι, ενώ υπάρχει μια τάση βελτίωσης, καθώς αυξάνεται το μέγεθος του πληθυσμού, οι διαφορές δεν είναι σημαντικές. Και εδώ θα επιλέξουμε 50 χρωμοσώματα για την τελική εκτέλεση. Έχουμε λοιπόν, για $p_{cr} = 1$, $p_{mut} = 0{,}001$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 10.000 γενεές, μέση ποσότητα στην αγορά 82,4325 και τυπική απόκλιση 2,4794.



Για τις μεμονωμένες επιχειρήσεις έχουμε

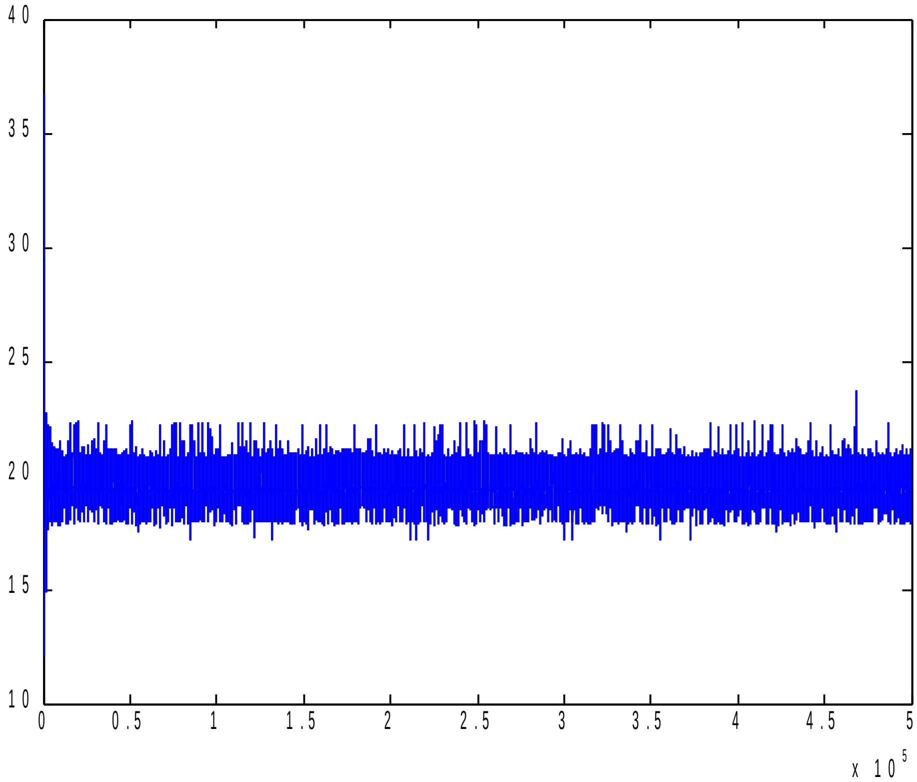


Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	77,1764	5,0346
2	77,1933	4,9813
3	92,5851	4,7529
4	82,7751	4,9192

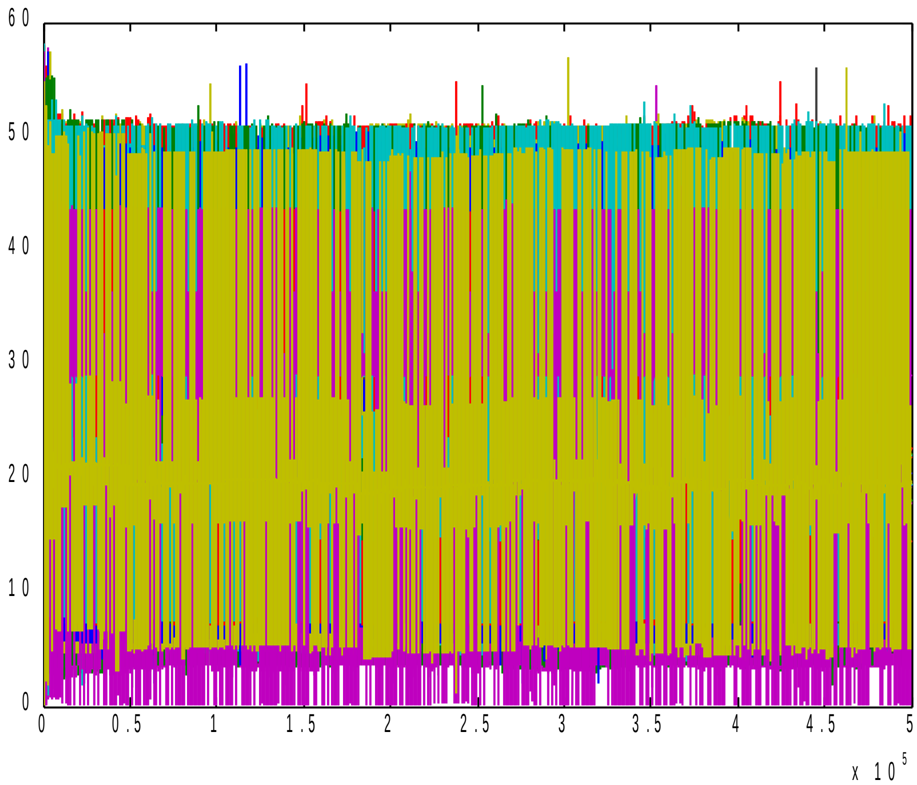
σε κανένα απ΄ τα 500.000 παίγνια δεν είχαμε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0,8559	0,1291	0,0149	0,0001	0	0	0	0	0	0	0	0

Για την περίπτωση των 20 επιχειρήσεων τώρα, με $p_{cr} = 1$, $p_{mut} = 0{,}001$, αριθμό χρωμοσωμάτων στους πληθυσμούς 50, και 10.000 γενεές, έχουμε μέση ποσότητα στην αγορά 19,3841και τυπική απόκλιση 0,3013



Για τις μεμονωμένες επιχειρήσεις έχουμε



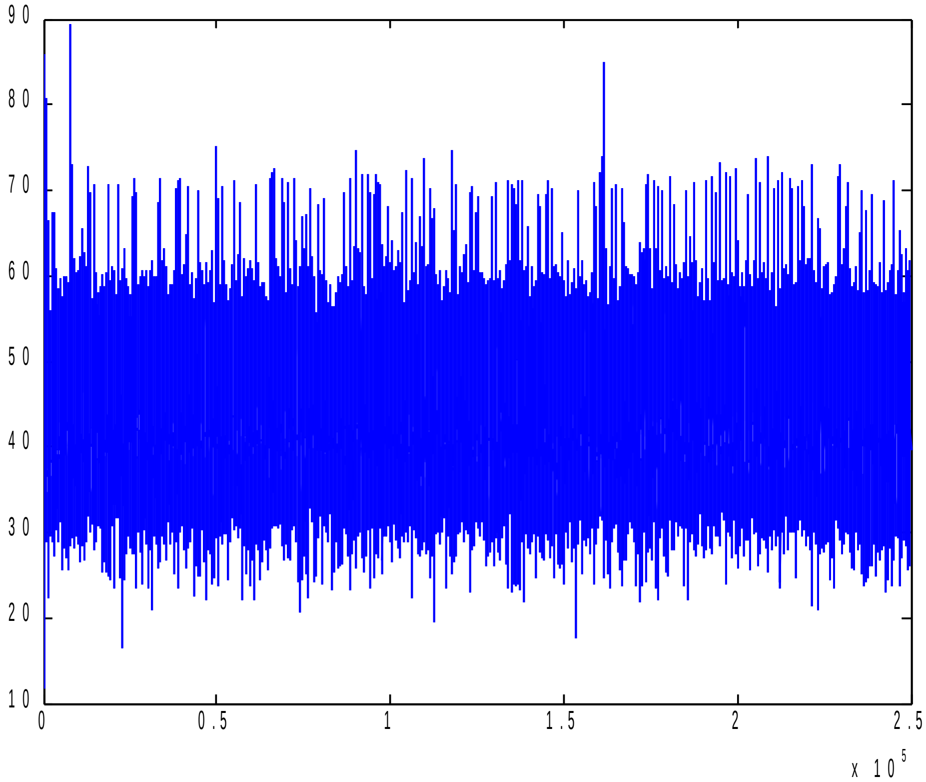
Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	14,4854	1,1795
2	17,9972	1,3021
3	21,8966	1,6421
4	14,5266	1,5816
5	17,9416	1,3326
6	22,0184	1,2974
7	19,6214	1,3477
8	19,6606	1,3203
9	18,0526	1,2644
10	22,0232	1,2417
11	19,6468	1,4052
12	17,9687	1,2573
13	19,7184	1,5547
14	14,4698	1,2233
15	20,3235	1,3431
16	22,0685	1,3137
17	14,4645	1,2455
18	21,9716	1,2617
19	29,1391	1,3701
20	19,6881	1,3713

Σε κανένα από τα 500.000 δεν είχαμε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0	0	0	0,2792	0,7113	0,0095	0	0	0	0	0	0	0	0	0

5.5.10. Εφαρμογή του αλγόριθμου των Alkemade et al. στο γραμμικό υπόδειγμα

Ακολουθώντας του Alkemade et al(2007), θέτουμε $p_{cr} = 1$, $p_{mut} = 0{,}01$. Η μόνη διαφορά είναι δηλαδή ο αριθμός bits των χρωμοσωμάτων (εκείνοι επέλεξαν $l = 6 - 9$, ενώ εμείς $l = 20$). Για την περίπτωση των 4 επιχειρήσεων επιλέγουμε πληθυσμό 100 χρωμοσωμάτων. Η αρχικοποίηση γίνεται τυχαία. Για 10000 γενεές έχουμε μέση ποσότητα 40,8905 και τυπική απόκλιση 4,1137.



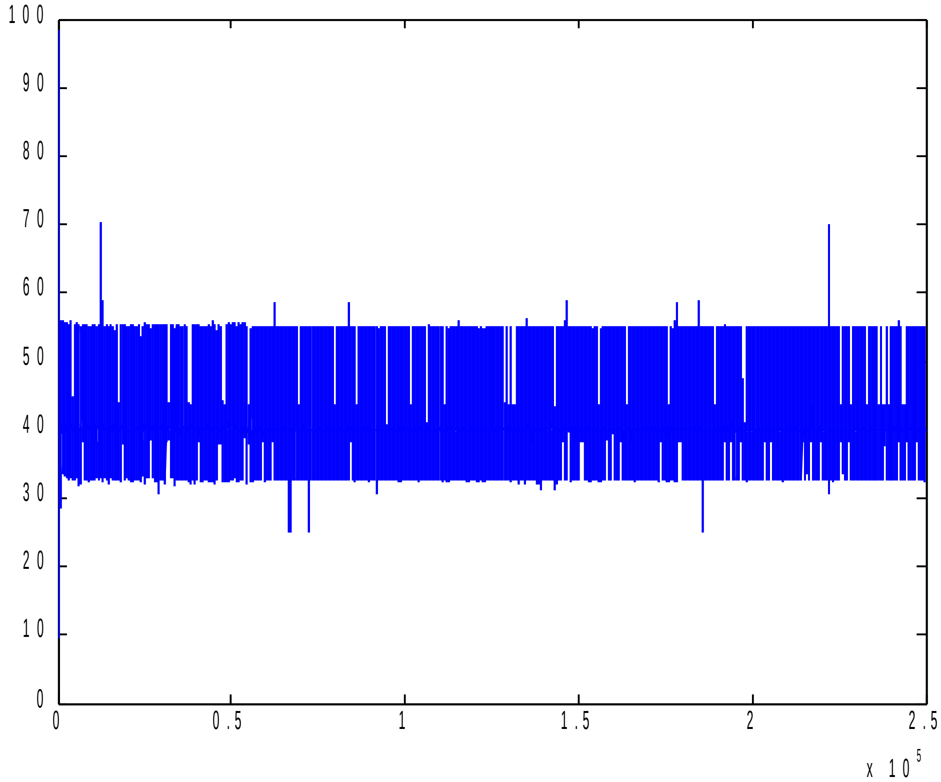
Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,9140	8,1633
2	40,8884	8,0695
3	40,8982	8,1250
4	40,8613	8,0567

Σε κανένα από τα 25.000 παίγνια δεν είχαμε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0,001	0,0179	0,1184	0,2876	0,3429	0,1943	0,0375	0,0004	0	0	0	0	0	0	0	0

Αλλάζοντας την πιθανότητα μετάλλαξης σε $p_{mut} = 0{,}001$, έχουμε την πρώτη επιτυχημένη απόπειρα, όσον αφορά την εύρεση Nash Equilibrium! Η μέση ποσότητα στην αγορά είναι 40,0930 με τυπική απόκλιση 1,2672.



Για τις μεμονωμένες επιχειρήσεις έχουμε:

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,0901	2,3945
2	40,0928	2,4908
3	40,0990	2,5051
4	40,0902	2,4452

Σε 2.732 από τα 25.000 παίγνια που «παίχτηκαν» κατά τη διάρκεια της προσομοίωσης είχαμε ισορροπία κατά Nash, δηλαδή $q_1 = \ldots = q_4 = 40$. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

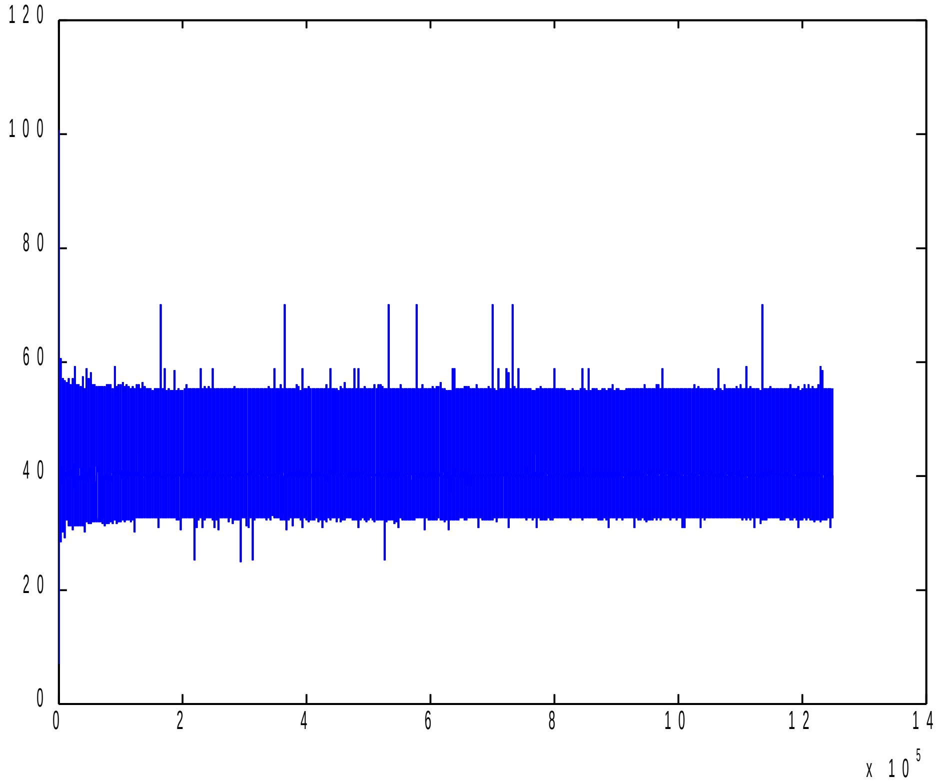
0,0024	0,0133	0,0133	0,0442	0,079	0,2345	0,2084	0,1363	0,118	0,1091	0,041	0,0005
0	0	0	0	0	0	0	0	0			

και, για τις γενεές από τη 1000η και μετά- οπότε αποφεύγουμε τον αρχικό θόρυβο, όπως είπαμε προηγουμένως

0,0025	0,0134	0,0134	0,0443	0,0791	0,2346	0,2085	0,1364	0,1181	0,1092	0,0411
0	0	0	0	0	0	0	0	0	0	

Οι διαφορές δεν είναι σημαντικές. Ο μέσος χρόνος μεταξύ διαδοχικών «επισκέψεων» στην κατάσταση-πληθυσμό με όλα τα χρωμοσώματα ίσα με την ποσότητα Nash είναι $\frac{1}{\pi_0} = 0,0025^{-1} =$ 400 γενεές. Το πρώτο παίγνιο με ισορροπία Nash ήταν το 243.350ο παίγνιο (το 15ο παίγνιο της 608ης γενεάς), ενώ η πρώτη κατάσταση Nash (όλα τα χρωμοσώματα ίσα με το χρωμόσωμα 01...01 που αντιστοιχεί σε $q = 40$) ήταν η 9.890η γενιά. Δεδομένου ότι η γενιά αυτή βρίσκεται κοντά στο τέλος του χρονικού ορίζοντα της προσομοίωσης, είναι σκόπιμο να επαναλάβουμε, με μεγαλύτερο αριθμό βημάτων για καλύτερη εκτίμηση του διανύσματος σταθερών συχνοτήτων.

Για πληθυσμό 500 χρωμοσωμάτων και $p_{mut} = 0{,}001$ έχουμε μέση ποσότητα στην αγορά 40,0744 και τυπική απόκλιση 1,2206.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,0736	2,4032
2	40,0753	2,4211
3	40,0742	2,4039
4	40,0746	2,4116

Σε κανένα από τα 1.250.000 παίγνια δεν είχαμε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0,0386	0,2015	0,2421	0,3099	0,1417	0,0585	0,0077	0	0	0	0	0	0	0	0	0	0

Ο μεγαλύτερος λοιπόν πληθυσμός, δεν έδωσε καλύτερα αποτελέσματα, προφανώς λόγω του ότι ο αριθμός επαναλήψεων δεν είναι αρκετός για σύγκλιση όλων των χρωμοσωμάτων. Βέβαια ο χρόνος υπολογισμού που απαιτήθηκε ήταν ήδη αυξημένος, οπότε δεν κρίνεται σκόπιμο να προσπαθήσουμε με μεγαλύτερο αριθμό γενεών.

Αντ΄ αυτού επιστρέφουμε στην περίπτωση των 100 χρωμοσωμάτων και $p_{mut} = 0{,}001$ και εκτελούμε την προσομοίωση για 100.000 γενεές.

Από τα 2.500.000 παίγνια, είχαμε ισορροπία κατά Nash στα 2.224.900 με το πρώτο να είναι το 48.959^{ο} (το 9^{ο} της 1.958^{ης} γενεάς). Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,10057	0,87991	0,00027	0,00111	0,00934	0,00166	0,00011	0,00141	0,00197	0,00325	0,00034
6,00E-05	0	0	0	0	0	0	0	0	0	

με τον πρώτο πληθυσμό Nash να προκύπτει στην 2.020^{η} γενεά. Το αποτέλεσμα είναι αρκετά διαφορετικό δηλαδή από την περίπτωση των 10.000 γενεών. Στην περίπτωση των 100.000 η εκτίμηση αναμένεται να είναι σαφώς καλύτερη, λόγω του αυξημένου αριθμού παρατηρήσεων. Ο μέσος χρόνος επαφοράς στην κατάσταση Nash εκτιμάται ως $\frac{1}{0{,}10057} = 9{,}9433$, δηλαδή λιγότερος από 10 γενεές. Συμπεραίνουμε ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,1372	0,8628										
0,0986	0,9014										
	0,0370	0,9630									
		0,0090	0,9820	0,0090							
			0,0021	0,9807	0,0171						
				0,1024	0,8916	0,0060					
					0,1818	0,8182					
						0,0071	0,9433	0,0496			
							0,0406	0,8782	0,0812		
0,1372	0,8628										
									0,4412	0,5588	
										0,1667	0,8333

Προφανώς για μια ικανοποιητική εκτίμηση όλων των πιθανοτήτων μετάβασης απαιτείται ένας ιδιαίτερα μεγάλος αριθμός παρατηρήσεων. Ωστόσο οι πιθανότητες μετάβασης δεν είναι απαραίτητες για την αξιολόγηση των αλγορίθμων, ενώ έχουν και περιορισμένη γενικότερη σημασία, αφού δεν μπορούν να χρησιμοποιηθούν για την εκτίμηση των πιθανοτήτωνν μετάβασης P^n, λόγω της απώλειας της Μαρκοβιανής ιδιότητας, στην ομαδοποιημένη διαδικασία.

Παρατηρούμε ότι η πιθανότερη ομαδική κατάσταση είναι εκείνη με τη μέση απόσταση 1 από την κατάσταση Nash, πράγμα λογικό, λόγω της μετάλλαξης. Καμμία συγκεκριμένη κατάσταση της ομαδικής κατάστασης A_1 δεν είχε σημαντική πιθανότητα, με αποτέλεσμα το συμπέρασμα ότι η κατάσταση Nash είναι η κατάσταση σύγκλισης του αλγορίθμου (με την ευρεία έννοια της σύγκλισης, αφού έχουμε αλυσίδα Markov) να είναι ασφαλές.

Όσον αφορά τον χρόνο πρώτης μετάβασης, επιλέγουμε σαν αρχική κατάσταση τη χειρότερη δυνατή, την ακριβώς αντίθετη δηλαδή (με την έννοια των bits) από την κατάσταση Nash. Ο αρχικός πληθυσμός λοιπόν θα αποτελείται από τα χρωμοσώματα $10...10$. Δεδομένου του τρόπου που λειτουργεί ο γενετικός αλγόριθμος, είναι λογικό (βλ. και κεφ.2) να υποθέσουμε ότι ο χρόνος μετάβασης θα είναι ο χειρότερος δυνατός για αυτήν την αρχική κατάσταση.

Εκτελέσαμε 30 (ανεξάρτητες μεταξύ τους εννοείται) προσομοιώσεις και η εκτιμήτρια του αναμενόμενου χρόνου αρχικής μετάβασης που βρήκαμε ήταν 9915 γενεές. Εκτιμάμε λοιπόν ότι, στη χειρότερη περίπτωση, χρειάζονται 9915 γενεές κατά μέσο όρο, για να βρει για πρώτη φορά ο αλγόριθμος την κατάσταση με όλα τα χρωμοσώματα ίσα με την ποσότητα της ισορροπίας κατά Nash.

Στην περίπτωση των 20 επιχειρήσεων τώρα, με $p_{mut} = 0{,}001$ και πληθυσμό 100 χρωμοσωμάτων, έχουμε σε 20.000 γενεές εκτιμηθέν διάνυσμα σταθερών συχνοτήτων,

0	0	0	0	0	0	0,00215	0,10195	0,33335	0,32545	0,19	0,03995	0,00715	0	0	0	0	0	0	0	0

και κανένα παίγνιο με ισορροπία. Η μέση ποσότητα στην αγορά είναι 9,8349 με τυπική απόκλιση 0,2398. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	9,8373	0,9436
2	9,8388	0,9589
3	9,8285	0,9481
4	9,8337	0,9466
5	9,8320	0,9523
6	9,8354	0,9514
7	9,8368	0,9582
8	9,8395	0,9422
9	9,8363	0,9466
10	9,8315	0,9647
11	9,8374	0,9468
12	9,8415	0,9537
13	9,8361	0,9807
14	9,8360	0,9332
15	9,8312	0,9595
16	9,8336	0,9712
17	9,8357	0,9694
18	9,8326	0,9352
19	9,8307	0,9558
20	9,8330	0,9656

Βλέπουμε ότι και πάλι έχουμε σύγκλιση όλων των ποσοτήτων όλων των παικτών κοντά στην μέση ποσότητα, αλλά πιθανότατα ο πληθυσμός είναι πολύ μικρός για να βρούμε την ισορροπία κατά Nash.

Αλλά και για πληθυσμό 400 χρωμοσωμάτων και 20.000 επαναλήψεις (μια υλοποίηση που φτάνει στα όρια της υπολογιστικής ισχύς που έχουμε στη διάθεσή μας), πάλι δε βρέθηκε ισορροπία κατά Nash. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0	0	0,01645	0,20805	0,607	0,1588	0,0097	0	0	0	0	0	0	0	0	0	0

Βλέπουμε ότι η βελτίωση είναι πολύ μικρή.

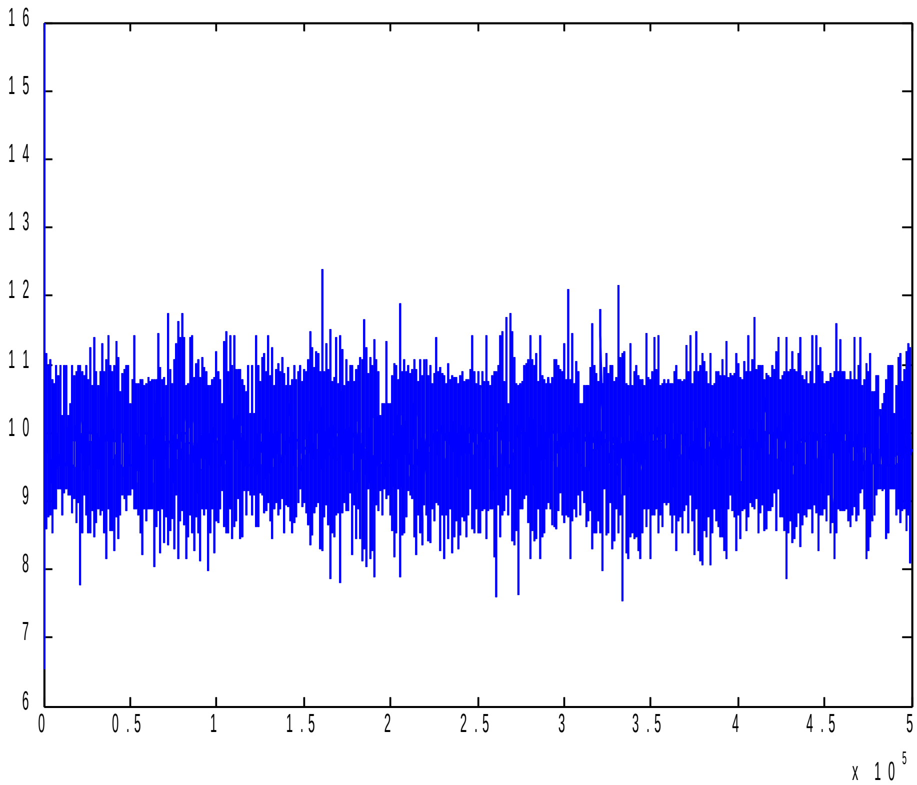
Ούτε μια μείωση της πιθανότητας μετάλλαξης βοηθάει. Για πληθυσμό 100 χρωμοσωμάτων, 10000 επαναλήψεις και $p_{mut} = 0{,}00025$ το εκτιμηθέν διάνυσμα είναι

0	0	0	0	0	0,0163	0,0286	0,0478	0,1729	0,2352	0,354	0,1322	0,013	0	0	0	0	0	0	0	0

ενώ με 200 χρωμοσώματα στον πληθυσμό $p_{mut} = 0{,}00001$ και 20.000 επαναλήψεις έχουμε

0	0	0	0	0	0	0,01	0,0154	0,7292	0,2349	0,0105	0	0	0	0	0	0	0	0	0	0

Τελικά μετά από πολλές απόπειρες που έφτασαν τον υπολογιστή μου στα όριά του, κατέληξα στο συμπέρασμα ότι αναγκαστικά πρέπει να μειώσουμε το μέγεθος του χρωμοσώματος, λόγω ανεπαρκούς υπολογιστικής ισχύς. Πράγματι, για μέγεθος χρωμοσώματος 6 bits, $p_{cr} = 1$, $p_{mut} = 0{,}001$ και 100 χρωμοσώματα στον πληθυσμό, έχουμε, για 100.000 γενεές μέση ποσότητα 9,8176 με τυπική απόκλιση 0,2525.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	9,8167	0,9319

2	9,8171	0,9337
3	9,8193	0,9440
4	9,8201	0,9452
5	9,8163	0,9436
6	9,8179	0,9430
7	9,8208	0,9478
8	9,8202	0,9472
9	9,8169	0,9361
10	9,8161	0,9427
11	9,8166	0,9332
12	9,8179	0,9351
13	9,8165	0,9316
14	9,8170	0,9380
15	9,8157	0,9338
16	9,8156	0,9446
17	9,8167	0,9472
18	9,8191	0,9450
19	9,8182	0,9437
20	9,8168	0,9397

Είχαμε ισορροπία κατά Nash σε 45.585 από τα 50.000 παίγνια που εκτελέστηκαν στην προσομοίωση. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,04737	0,31739	0,43456	0,20068	0	0	0

και, αν ξεκινήσουμε από την 50.000η γενεά,

0,04738	0,3174	0,43457	0,20069	0	0	0

Το πρώτο παίγνιο με ισορροπία κατά Nash ήταν το 7546ο (το 1ο της 1509ης γενιάς) και ο πρώτος πληθυσμός Nash παρουσιάστηκε στην 1729η γενιά.

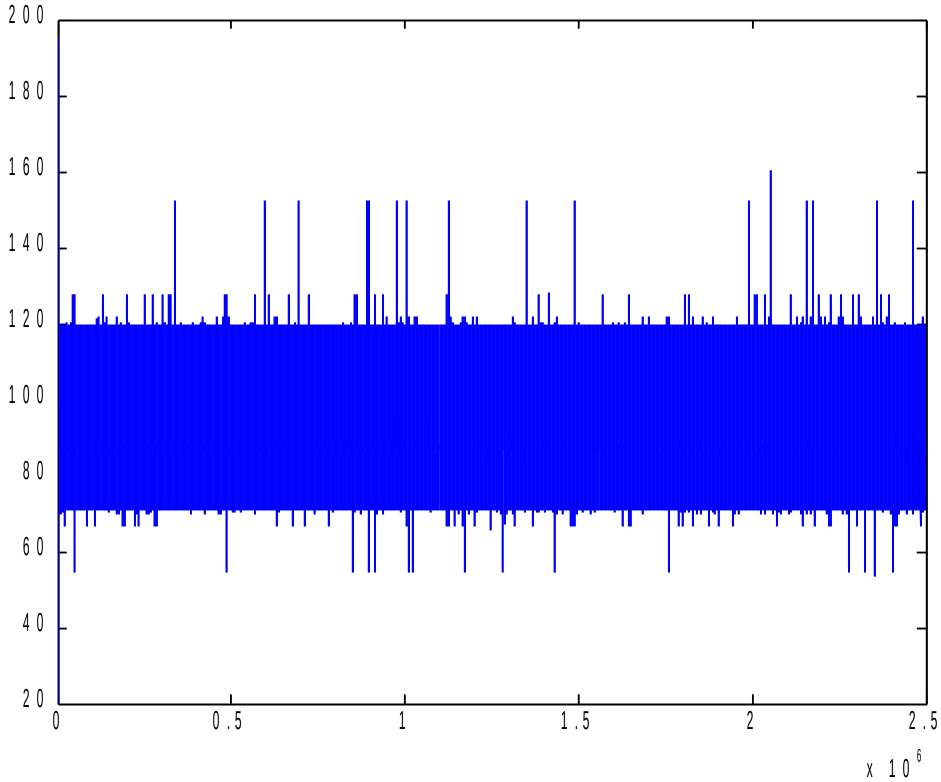
Ο χρόνος επαναφοράς εκτιμάται ως $\frac{1}{0{,}04738} = 21{,}106$. Χρειάζονται δηλαδή περίπου 21 γενεές για να επανέλθουμε στον πληθυσμό με όλα τα χρωμοσώματά του ίσα με την ποσότητα της ισορροπίας κατά Nash. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,5406	0,4594		
0,0686	0,9162	0,0152	
	0,0111	0,9309	0,0579
		0,1254	0,8746

Παρατηρούμε ότι οι πλέον ευσταθείς ομαδικές καταστάσεις είναι εκείνες με μέση απόσταση από τον πληθυσμό Nash, ίση με 1 και 2 bits αντίστοιχα.

5.5.11. Εφαρμογή του αλγόριθμου των Alkemade et al. στο πολυωνυμικό υπόδειγμα

Ξεκινάμε με την περίπτωση 4 επιχειρήσεων. Διατηρούμε τις παραμέτρους που θέσαμε προηγουμένως, στην προσομοίωση για το γραμμικό υπόδειγμα, $p_{cr} = 1$, $p_{mut} = 0{,}001$, $l = 20$, 100 χρωμοσώματα στον πληθυσμό και 100.000 επαναλήψεις. Η μέση ποσότητα στην αγορά ήταν 87,0332 και η τυπική της απόκλιση 2,4475.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	87,0320	4,8787
2	87,0363	4,8923
3	87,0347	4,9030
4	87,0299	4,8697

Σε 2.219.764 από τα 2.500.000 παίγνια που «παίχτηκαν» κατά τη διάρκεια της προσομοίωσης είχαμε ισορροπία κατά Nash, δηλαδή $q_1 = \ldots = q_4 = 40$. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

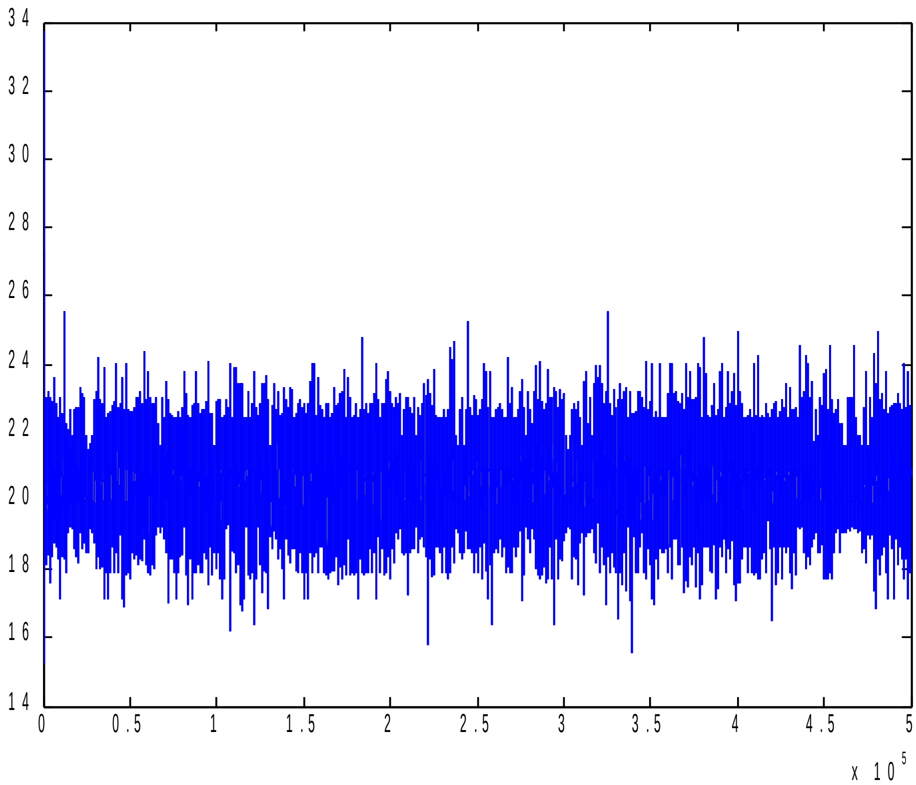
0,10029 0,87742 0,00299 0,00103 0,00148 0,00992 0,0037 0,0017 0,00112 0,00018 0,00014 3,00E-05
0 0 0 0 0 0 0 0 0

Το πρώτο παίγνιο με ισορροπία κατά Nash ήταν το $55881^{ο}$ (το $7^{ο}$ της $2235^{ης}$ γενιάς) και ο πρώτος πληθυσμός Nash παρουσιάστηκε στην $2267^{η}$ γενιά.

Ο χρόνος επαναφοράς εκτιμάται ως $\frac{1}{0{,}1003} = 9{,}97009$. Χρειάζονται δηλαδή περίπου 10 γενιές για να επανέλθουμε στον πληθυσμό με όλα τα χρωμοσώματά του ίσα με την ποσότητα της ισορροπίας κατά Nash. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,1369	0,8631										
0,0987	0,9013										
	0,0067	0,9632	0,0301								
		0,0971	0,9029								
			0,0068	0,9122	0,0811						
				0,0131	0,9748	0,0121					
					0,0351	0,9432	0,0216				
						0,0529	0,9176	0,0294			
							0,0536	0,9286	0,0179		
								0,1667	0,8333		
									0,0714	0,8571	0,0714
										0,3333	0,6667

Για την περίπτωση των 20 επιχειρήσεων έχουμε, για μέγεθος χρωμοσώματος 6 bits, $p_{cr} = 1$, $p_{mut} = 0{,}001$ και 100 χρωμοσώματα στον πληθυσμό και για 100.000 γενεές, μέση ποσότητα στην αγορά 20,5712 με τυπική απόκλιση 0,5379.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση

1	20,574	1,9742
2	20,578	1,9522
3	20,569	1,9693
4	20,572	1,9541
5	20,571	1,9617
6	20,571	1,9691
7	20,57	1,9693
8	20,574	1,989
9	20,576	1,9609
10	20,571	1,9642
11	20,564	1,9845
12	20,573	1,963
13	20,57	1,9629
14	20,569	1,9687
15	20,566	1,9646
16	20,572	1,954
17	20,574	1,9836
18	20,571	1,9402
19	20,569	1,9662
20	20,569	1,9779

Σε 64.315 από τα 500.000 παίγνια που «παίχτηκαν» κατά τη διάρκεια της προσομοίωσης είχαμε ισορροπία κατά Nash, δηλαδή $q_1 = \ldots = q_4 = 40$. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,06757	0,23799	0,43285	0,2609	0,00069	0	0

Αν αγνοήσουμε τον αρχικό θόρυβο και χρησιμοποιήσουμε τις γενιές από την 50.000η και μετά, το διάνυσμα είναι

0,06758	0,238	0,43286	0,26091	0	0	0

Δεν υπάρχουν δηλαδή σημαντικές διαφορές. Το πρώτο παίγνιο με ισορροπία κατά Nash ήταν το 5515ο (το 5ο της 1103ης γενεάς) και ο πρώτος πληθυσμός Nash παρουσιάστηκε στην 2617η γενεά.

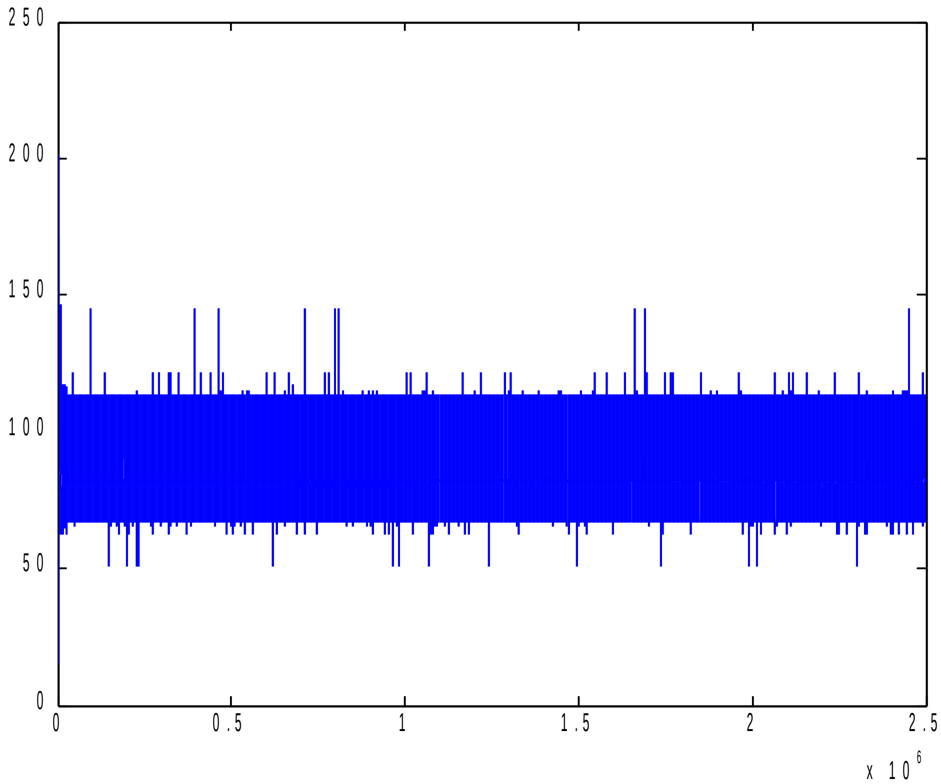
Ο χρόνος επαναφοράς εκτιμάται ως $\frac{1}{0,06758} = 14,7973$. Χρειάζονται δηλαδή περίπου 15 γενεές για να επανέλθουμε στον πληθυσμό με όλα τα χρωμοσώματά του ίσα με την ποσότητα της ισορροπίας κατά Nash.

Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,5551	0,4449			
0,1263	0,8570	0,0167		
	0,0092	0,9200	0,0708	
		0,1174	0,8824	0,0002
			0,0725	0,9275

5.5.12. Εφαρμογή του αλγόριθμου των Alkemade et al. στο υπόδειγμα με ρητό εκθέτη

Ξεκινάμε με την περίπτωση 4 επιχειρήσεων. Οι παράμετροι που θα χρησιμοποιήσουμε είναι οι ίδιες με πρηγουμένως, με μόνη διαφορά των αριθμό δυαδικών ψηφίων στο χρωμόσωμα, $p_{cr}=1$, $p_{mut}=0{,}001$, $l=6$, 100 χρωμοσώματα στον πληθυσμό και 100.000 επαναλήψεις. Η μέση ποσότητα στην αγορά ήταν 82,3050και η τυπική της απόκλιση 2,3611.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	82,3016	4,6934
2	82,3072	4,7298
3	82,3040	4,7106
4	82,3070	4,7208

Σε 2.412.623 από τα 2.500.000 παίγνια που «παίχτηκαν» κατά τη διάρκεια της προσομοίωσης είχαμε ισορροπία κατά Nash, δηλαδή $q_1 = \ldots = q_4 = 40$. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

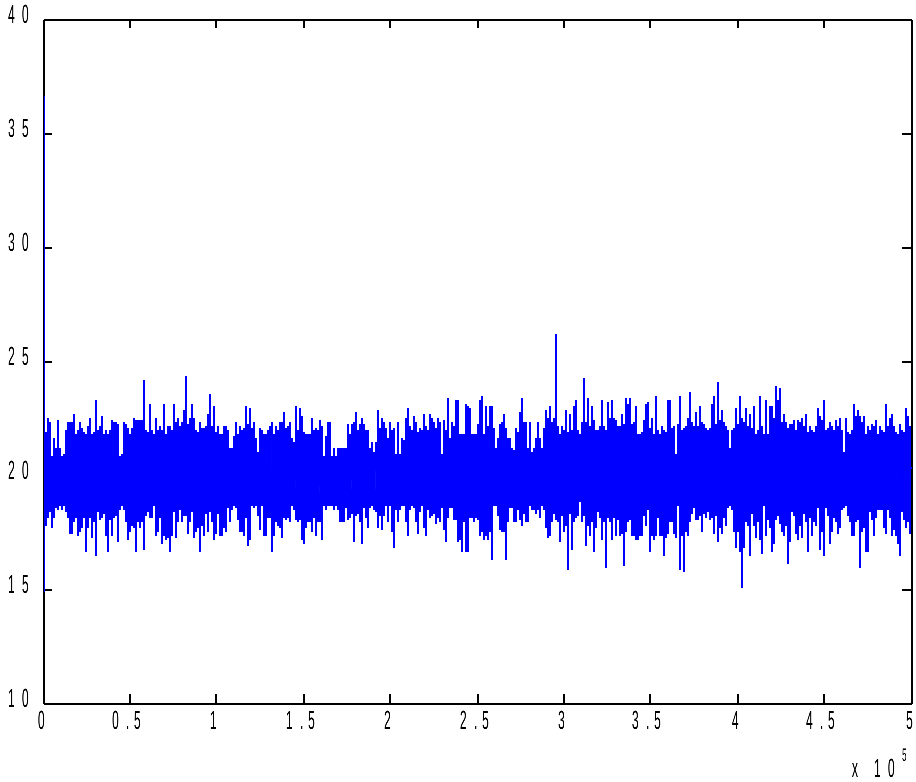
0,52442	0,46626	0,00882	0,00032	0,00018	0	0

Το πρώτο παίγνιο με ισορροπία κατά Nash ήταν το 23049^{ο} και ο πρώτος πληθυσμός Nash παρουσιάστηκε στην 969^{η} γενεά.

Ο χρόνος επαναφοράς εκτιμάται ως $\frac{1}{0{,}52442} = 1{,}9069$. Βλέπουμε ότι ο μικρότερος αριθμός δυαδικών ψηφίων μείωσε σημαντικά τον χρόνο επαναφοράς στην κατάσταση Nash. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,5485	0,4515			
0,5078	0,4921	0,0001		
	0,0057	0,9943		
		0,0313	0,9063	0,0625
			0,1111	0,8889

Για την περίπτωση των 20 επιχειρήσεων έχουμε, για μέγεθος χρωμοσώματος 6 bits, $p_{cr} = 1$, $p_{mut} = 0{,}001$ και 100 χρωμοσώματα στον πληθυσμό και για 100.000 γενεές, μέση ποσότητα στην αγορά 19,9101με τυπική απόκλιση 0,5188.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	19,9113	1,8920
2	19,9115	1,8521
3	19,9126	1,8777
4	19,9110	1,8844
5	19,9139	1,8732
6	19,9065	1,8636
7	19,9106	1,8661
8	19,9085	1,8581
9	19,9066	1,8589
10	19,9116	1,8901
11	19,9117	1,8601
12	19,9093	1,8799
13	19,9111	1,8865
14	19,9088	1,8650
15	19,9040	1,8876
16	19,9146	1,8629
17	19,9052	1,8517
18	19,9101	1,8536
19	19,9105	1,8632
20	19,9125	1,8650

Σε 76.161 από τα 500.000 παίγνια που «παίχτηκαν» κατά τη διάρκεια της προσομοίωσης είχαμε ισορροπία κατά Nash, δηλαδή $q_1 = \ldots = q_4 = 40$. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,07822	0,35078	0,38957	0,18143	0	0	0

Αν αγνοήσουμε τον αρχικό θόρυβο και χρησιμοποιήσουμε τις γενιές από την 50.000η και μετά, το διάνυσμα είναι

0,07823	0,35079	0,38958	0,18144	0	0	0

Δεν υπάρχουν δηλαδή σημαντικές διαφορές. Το πρώτο παίγνιο με ισορροπία κατά Nash ήταν το 4364^{ο} (το 4^{ο} της 872^{ης} γενεάς) και ο πρώτος πληθυσμός Nash παρουσιάστηκε στην 879^{η} γενιά.

Ο χρόνος επαναφοράς εκτιμάται ως $\frac{1}{0{,}07822} = 12{,}7828$. Χρειάζονται δηλαδή περίπου 13 γενεές για να επανέλθουμε στον πληθυσμό με όλα τα χρωμοσώματά του ίσα με την ποσότητα της ισορροπίας κατά Nash.

Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,5523	0,4477	0,0000	0,0000
0,0998	0,8853	0,0149	0,0000
0,0000	0,0134	0,9343	0,0523
0,0000	0,0000	0,1123	0,8876

5.6. Συμπεράσματα

Η εφαρμογή των αλγορίθμων της Arifovic (1994), του Vriend (2000) του co-evolutionary programming (Price 1997) και των Alkemade et al. (2007) στα 3 συμμετρικά υποδείγματα που χρησιμοποιήσαμε (γραμμικό, πολυωνυμικό και με ρητό εκθέτη), και τα οποία είχαν όλα τόσο ισορροπία κατά Nash, όσο και Βαλρασιανή ισορροπία, αποκάλυψε κάποιες εγγενείς αδυναμίες των αλγορίθμων. Ο μοναδικός αλγόριθμος που οδήγησε στην πραγμάτωση παιγνίων με ισορροπία κατά Nash, στην προσέγγιση των ποσοτήτων της ισορροπίας από όλους τους παίκτες και σε υψηλές παρατηρούμενες συχνότητες της «κατάστασης Nash» (της κατάστασης της αλυσίδας Markov που περιγράφει την εξέλιξη των γενεών του αλγορίθμου, στην οποία όλα τα χρωμοσώματα όλων των πληθυσμών είναι ίσα με το χρωμόσωμα που αντιστοιχεί στην ποσότητα της ισορροπίας κατά Nash), ήταν ο αλγόριθμος των Alkemade et al (2007). Σημειώνουμε ότι, δεδομένου ότι ο αλγόριθμος των Alkemade et al. είναι αλγόριθμος κοινωνικής μάθησης (υπάρχει ένας πληθυσμός χρωμοσωμάτων) απαιτείται συμμετρία στις συναρτήσεις κέρδους των παικτών. Δεδομένου τώρα ότι σε ένα παίγνιο Cournot η συνάρτηση κέρδους εξαρτάται από την τιμή, που είναι κοινή για όλους τους παίκτες και την συνάρτηση κόστους, η παραπάνω συνθήκη ισοδυναμεί με συμμετρία στις συναρτήσεις κόστους. Αυτήν την ιδιότητα την είχαν. πράγματι, τα υποδείγματα που εισήχθησαν σε αυτήν την ενότητα. Στην αντίθετη περίπτωση, με διαφορετικές συναρτήσεις κόστους για τον κάθε παίκτη, ο αλγόριθμος των Alkemade et al. (2007) δεν μπορεί να εφαρμοστεί.

Όσον αφορά τώρα τα κριτήρια αξιολόγησης για την αποτελεσματικότητα του αλγορίθμου, όπως υπολογίστηκαν με βάση την εκτίμηση που προέκυψε από τις εκτελέσεις των προσομοιώσεων, είδαμε ότι η κατάσταση Nash ήταν αρκετά ευσταθής καθεαυτή. Σε όλα τα υποδείγματα ο αναμενόμενος χρόνος επαναφοράς ήταν μικρότερος από 10 γενιές στην περίπτωση των 4 παικτών (και 20 ψηφίων στα χρωμοσώματα) και 10-20 γενιές στην περίπτωση των 20 παικτών (και 6 ψηφία σε κάθε χρωμόσωμα). Δεδομένου ότι το κριτήριο αυτό είναι ιδιαίτερα αυστηρό (η κατάσταση Nash χαρακτηρίζεται από έναν πληθυσμό του οποίου *όλα* τα χρωμοσώματα είναι ίσα με την ποσότητα της ισορροπίας κατά Nash) το αποτέλεσμα κρίνεται ιδιαίτερα ικανοποιητικό. Από την άλλη πλευρά ο χρόνος πρώτης μετάβασης στην κατάσταση Nash από την αντίθετή της, όπως εκτιμήθηκε στην απλούστερη των περιπτώσεων (γραμμικό υπόδειγμα με 4 παίκτες) ήταν αυξημένος (περίπου 10.000 γενιές). Τόσο οι εκτιμηθέντες συχνότητες του διανύσματος σταθερών συχνοτήτων των ομαδικών (lumped) καταστάσεων, όσο και οι πιθανότητες μετάβασης που εκτιμήθηκαν με τη διαδικασία των Basawa and Rao (1980), έδειξαν ότι οι πλέον συχνές καταστάσεις της αλυσίδας είναι αφενός η κατάσταση Nash καθεαυτή και αφετέρου οι καταστάσεις, τα χρωμοσώματα των οποίων δεν απέχουν περισσότερο από το χρωμόσωμα που αντιστοιχεί στην ισορροπία κατά Nash από 1 δυαδικό ψηφίο κατά μέσο όρο (η αθροιστική συχνότητα των 2 αυτών ομαδικών καταστάσεων ξεπέρασε το 95% πολλές φορές). Αυτό μας δείχνει ότι πραγματικά, η πορεία τόσο της αγοράς συνολικά, όσο και των μεμονωμένων επιλογών των παικτών, όπως προσδιορίζονται από τον αλγόριθμο μάθησης είναι ιδιαίτερα ευσταθείς γύρω από την επιθυμητή κατάσταση. Μπορούμε να πούμε λοιπόν ότι ο αλγόριθμος των Alkemade et al (2007) αποτελεί πραγματικά έναν πολύ ικανοποιητικό αλγόριθμος μάθησης για το συμμετρικό παίγνιο Cournot, που οδηγεί στην ορθή συμπεριφορά τη διαδικασία μάθησης των πρακτόρων.

Ο αλγόριθμος της Arifovic (1994), οδήγησε σε σύγκλιση της μέσης ποσότητας της αγοράς στην ποσότητα που προβλέπει η ισορροπία Walras. Δεδομένης της αμφιμονοσήμαντης σχέσης μεταξύ της μέσης ποσότητας και της συνολικής, καθώς και της ποσότητας και της τιμής στην αγορά, το ίδιο συμβαίνει και για την ποσότητα και τιμή ισορροπίας στην αγορά. Αυτό που δεν συμβαίνει όμως, είναι η σύγκλιση των μέσων ποσοτήτων που επιλέγουν οι συγκεκριμένοι παίκτες, στην ποσότητα Walras. Πρόκειται για ένα πολύ σημαντικό μειονέκτημα, που δεν έχει καταγραφεί στη βιβλιογραφία μέχρι στιγμής, και το οποίο υποδηλώνει ότι οι πράκτορες που λειτουργούν με το πρότυπο του αλγορίθμου αυτού, δεν μαθαίνουν αποτελεσματικά (αν θεωρήσει κανείς ότι η Βαλρασιανή ισορροπία αποτελεί στόχο προς μάθηση).

Ανάλογα ήταν τα συμπεράσματα που αποκομίσαμε για τους αλγορίθμους του Vriend (2000) και co-evolutionary programming (Price 1997). Εδώ επιτυγχάνεται σύγκλιση στην ισορροπία κατά Nash, αντί της ισορροπίας κατά Walras, στην μέση και συνολική ποσότητα και την τιμή, αλλά οι ποσότητες που επιλέγουν οι μεμονωμένοι πράκτορες, συγκλίνουν σε διαφορετικά επίπεδα, για τον κάθε πράκτορα. Κι εδώ η «κατάσταση Nash» δεν παρατηρήθηκε ποτέ, σε κανένα βήμα των προσομοιώσεων που εκτελέσαμε και το ίδιο συνέβη για τα μεμονωμένα παίγνια : κανένα παίγνιο από όσα εκτελέστηκαν στη διάρκεια των προσομοιώσεων των αλγορίθμων, σε κανένα από τα υποδείγματα, δεν είχαμε ισορροπία κατά Nash. Έχουμε δηλαδή μια κατάσταση ακριβώς ανάλογη με εκείνη του αλγορίθμου της Arifovic, με την μόνη διαφορά ότι η «ισορροπία αναφοράς» είναι η ισορροπία Nash-Cournot, αντί της Βαλρασιανής. Οι επιμέρους διαφορές του αλγορίθμου του Vriend και του co-evolutionary programming είναι μικρές και το ίδιο συμβαίνει και με τα αποτελέσματά τους. Οι διαφορές στις αναμενόμενες ποσότητες (τόσο τις συνολικές όσο και τις επιμέρους) είναι ασήμαντες, ενώ όποιες διαφορές παρατηρήθηκαν στις τυπικές αποκλίσεις αυτών των αναμενόμενων ποσοτήτων, αποδίδονται στις διαφορετικές πιθανότητες μετάλαξης που χρησιμοποιήθηκαν κάθε φορά. Όπως είδαμε και στην ανάλυση του Riechmann, στο κεφάλαιο 2, μικρότερες πιθανότητες μετάλαξης οδηγούν σε μεγαλύτερη «ευστάθεια» του αλγορίθμου, γύρω από την τιμή στην οποία «ισορροπεί» (με την διευρυμένη έννοια της ευστάθειας του Ljapunov πάντα). Η σχέση της παραμέτρου GARate με το μέγεθος του πληθυσμού είναι βέβαια σημαντική στον αλγόριθμο του Vriend, σε αντίθεση με την περίπτωση του co-evolutionary programming, στο οποίο το ζήτημα αυτό ρυθμίζεται, κατά κάποιο τρόπο, αυτόματα, αφού η περίοδος ανανέωσης του πληθυσμού προκύπτει μονοσήμαντα από το πλήθος των χρωμοσωμάτων. Όπως είδαμε (βλ. και Valee και Yildizoglou 2007) το GARate πρέπει να είναι κοντά (αν όχι μεγαλύτερο από) στο 50, με τους πληθυσμούς να είναι επίσης περίπου ίσοι ή και μεγαλύτεροι από τον αριθμό αυτόν. Από αυτήν την σκοπιά, μπορούμε να πούμε ότι το co-evolutionary programming είναι πιο «γρήγορο» στην εκτέλεσή του, αφού λειτουργεί και για μικρότερους πληθυσμούς (ή ισοδύναμα, με μικρότερο GARate, αφού αυτά τα δύο είναι ίσα στον αλγόριθμο αυτό) με δεδομένο πάντα ότι θα κωδικοποιήσουμε κατάλληλα μια διαδικασία που να εγγυάται ότι κάθε χρωμόσωμα θα συμμετέχει σε ένα και μόνο ένα παίγνιο, πριν την ανανέωση του πληθυσμού (βλ. κώδικα Matlab στο παράρτημα Α-3).

Τελικά μπορούμε να πούμε ότι μόνο ο αλγόριθμος των Alkemade et al (2007) έδωσε ικανοποιητικά αποτελέσματα. Αντίθετα με τον αλγόριθμο αυτόν, οι αλγόριθμοι ατομικής μάθησης με πολλούς πληθυσμούς για κάθε παίκτη, δεν οδηγούν στην μάθηση της σωστής συμπεριφοράς, παρόλο που μακροσκοπικά οδηγούν στην «σωστή» ποσότητα και τιμή ισορροπίας. Μπορούμε να πούμε δηλαδή ότι ενώ έχουμε το 3^{ο} από τα ενδεχόμενα που ανέφερε ο Riechmann (βλ. κεφάλαιο 2) με Ljapunov stability στη συμπεριφορά των πρακτόρων και στα μακροοικονομικά μεγέθη, το οποίο χαρακτηρίζει τους περισσότερους γενετικούς αλγορίθμους μάθησης, οι επιλογές των πρακτόρων δεν συγκλίνουν στην επιθυμητή συμπεριφορά της στρατηγικής Nash, αλλά «κολλάνε» κατά κάποιο τρόπο σε «σχεδόν βέλτιστες απαντήσεις» στις συμπεριφορές των συμπαικτών τους που και αυτές επηρρεάζονται από τις συμπεριφορές των υπολοίπων παικτών. Έτσι αυτή η δυναμική διαδικασία μάθησης οδηγείται σε δεσπόζουσες επιλογές για τον κάθε παίκτη που, ενώ αποτελούν «σχεδόν βέλτιστες απαντήσεις» στις επιλογές των υπολοίπων, μπορούν να έχουν πολύ μεγάλη απόκλιση -τελικά, μέσω των αλληλεπιδράσεων αυτών- από την ποσότητα της ισορροπίας κατά Nash.

Το ζήτημα που τελικά ανακύπτει είναι το κατά πόσον θα μπορούσαμε να μετατρέψουμε τους αλγόριθμους αυτούς, που χαρακτηρίζονται από την ύπαρξη πολλών πληθυσμών, ένα για κάθε παίκτη, σε αλγόριθμους κοινωνικής αντί ατομικής μάθησης. Να επιτρέψουμε δηλαδή στην διαδικασία της μίμησης -όπως αυτή εκφράζεται μέσω του τελεστή διασταύρωσης σε έναν γενετικό αλγόριθμο μάθησης- να οδηγήσει στην εξίσωση μέσω της αλληλεπίδρασης, των δυνητικών επιλογών των παικτών, όπως αυτές προσδιορίζονται από τα χρωμοσώματα των αντίστοιχων πληθυσμών τους.

Για να γίνει βέβαια αυτό εφικτό, να έχει δηλαδή νόημα για έναν πράκτορα η μίμηση των επιλογών των συμπαικτών του, θα πρέπει να έχουμε συμμετρία στο παίγνιο, όπως αναλύσαμε παραπάνω. Διαφορετικά μια επιλογή η οποία μπορεί να είναι πολύ καλή για έναν παίκτη, θα μπορούσε να αποβεί ιδιαίτερα ζημιογόνα για κάποιον άλλο. Με δεδομένη όμως την περιορισμένη αποτελεσματικότητα των αλγορίθμων του Vriend και του co-evolutionary programming, στην έκδοσή τους της ατομικής μάθησης, κρίνουμε πως αυτή η θυσία είναι αναγκαία.

6. Μετατροπή των αλγορίθμων ατομικής μάθησης σε κοινωνικής

Στο προηγούμενο κεφάλαιο είδαμε ότι οι αλγόριθμοι ατομικής μάθησης της Arifovic (Arifovic 1994),του Vriend (Vriend 2000) και του co-evolutionary programming (Price 1997), δεν οδηγούν σε αποτελεσματική μάθηση των πρακτόρων-παικτών. Παρόλο που η μέση ποσότητα στην αγορά συγκλίνει στην ποσότητα της ισορροπίας κατά Walras (στον αλγόριθμο της Arifovic) ή κατά Nash (στους αλγόριθμους του Vriend και του co-evolutionary programming), οι μέσες ποσότητες για τις μεμονωμένες επιχειρήσεις δεν συγκλίνουν στην ίδια τιμή. Σε αυτό το κεφάλαιο θα δείξουμε πως μπορούμε να επιλύσουμε αυτό το πρόβλημα, με μετατροπή των αλγορίθμων σε αλγόριθμους κοινωνικής αντί ατομικής μάθησης. Δε θεωρούμε σκόπιμο να ασχοληθούμε περαιτέρω με τον αλγόριθμο της Arifovic, καθότι δεν συγκλίνει στην ισορροπία κατά Nash, όπως θα έπρεπε. Παρόλα αυτά και σε αυτόν επιτυγχάνεται η σύγκλιση όλων των ποσοτήτων στην ισορροπία κατά Walras και μπορεί να τεκμηριωθεί και από άποψης οικονομικής θεωρίας ο τρόπος με τον οποίο επιτυγχάνεται η αλληλεπίδραση. Ωστόσω δεν θα ασχοληθούμε εδώ με αυτό το θέμα.

6.1. Μετατροπή των αλγορίθμων

Η μη σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα ισορροπίας οφείλεται στο γεγονός ότι οι πληθυσμοί με τα χρωμοσώματα- εναλλακτικές των διαφόρων επιχειρήσεων εξελίσσονται αυτόνομα μεταξύ τους. Αν αντί για αυτή τη μέθοδο ατομικής μάθησης, επιτρέψουμε σε κάθε επιχείρηση να λαμβάνει υπόψη τις αποφάσεις των υπολοίπων επιχειρήσεων, καθώς αναθεωρεί τις δικές τις εναλλακτικές, αν με άλλα λόγια ενώσουμε τους πληθυσμούς μετά από κάθε γενεά του αλγορίθμου και εφαρμόσουμε τους τελεστές της διασταύρωσης και της μετάλλαξης στον ενιαίο αυτόν πληθυσμό και στην συνέχεια επαναφέρουμε τα νέα χρωμοσώματα στους πληθυσμούς των επιχειρήσεων, η σύγκλιση επιτυγχάνεται.

Η παραπάνω μέθοδος έχει οικονομική αιτιολόγηση, δεδομένου ότι οι επιχειρήσεις έχουν την ευκαιρία στους αλγόριθμους του Vriend και στο co-evolutionary programming, να παρατηρήσουν όλες τις επιλογές των συμπαικτών τους, καθώς γίνεται το τυχαίο συνταίριασμα μεταξύ των εναλλακτικών των δικών τους και των αντιπάλων τους και εκτελούνται τα παίγνια. Άρα είναι από οικονομικής άποψης δυνατό και όπως θα δούμε και σκόπιμο να τις λάβουν υπόψη τους καθώς αναθεωρούν τα χρωμοσώματα-εναλλακτικές τους για την επόμενη γενεά.

Ο αλγόριθμος του Vriend αναθεωρείται ως εξής (βλ. Παράρτημα Α5 για κώδικα Matlab):

- Επιλέγονται με τυχαίο τρόπο τα χρωμοσώματα-στρατηγικές q_{ij} των πληθυσμών των i επειχειρήσεων.

- Για δεδομένο αριθμό περιόδων εκτέλεσε τον βρόχο
    - Αν η τρέχουσα περίοδος είναι ακέραιο πολλαπλάσιο της παραμέτρου GArate, αντέγραψε όλα τα χρωμοσώματα των πληθυσμών των επιχειρήσεων σε έναν μεγάλο πληθυσμό, εφάρμοσε τους τελεστές του γενετικού αλγορίθμου (διασταύρωσης, μετάλλαξης κλπ.) και αναθεώρησε τους πληθυσμούς των επιχειρήσεων με τα αντίστοιχα χρωμοσώματα της νέας γενιάς.
    - Αν η τρέχουσα περίοδος δεν είναι ακέραιο πολλαπλάσιο της GArate, επέλεξε με κάποιο τρόπο κάποια στρατηγική από τον πληθυσμό της κάθε επιχείρησης, σαν την τρέχουσα στρατηγική της q_{ij}^*. Το κέρδος της στρατηγικής προσδιορίζεται από το αποτέλεσμα του παιγνίου Cournot μεταξύ των τρεχουσών στρατηγικών όλων των επιχειρήσεων $\pi_i\left(q_{ij}^{*t}\right) = \pi_i\left(q_{ij}^*, Q_{-i}^*\right)$. Δηλαδή οι τρέχουσες στρατηγικές για όλες τις επιχειρήσεις χρησιμοποιούνται για να προσδιοριστεί η τρέχουσα τιμή, η οποία οδηγεί στον υπολογισμό του κέρδους της κάθε στρατηγικής ξεχωριστά

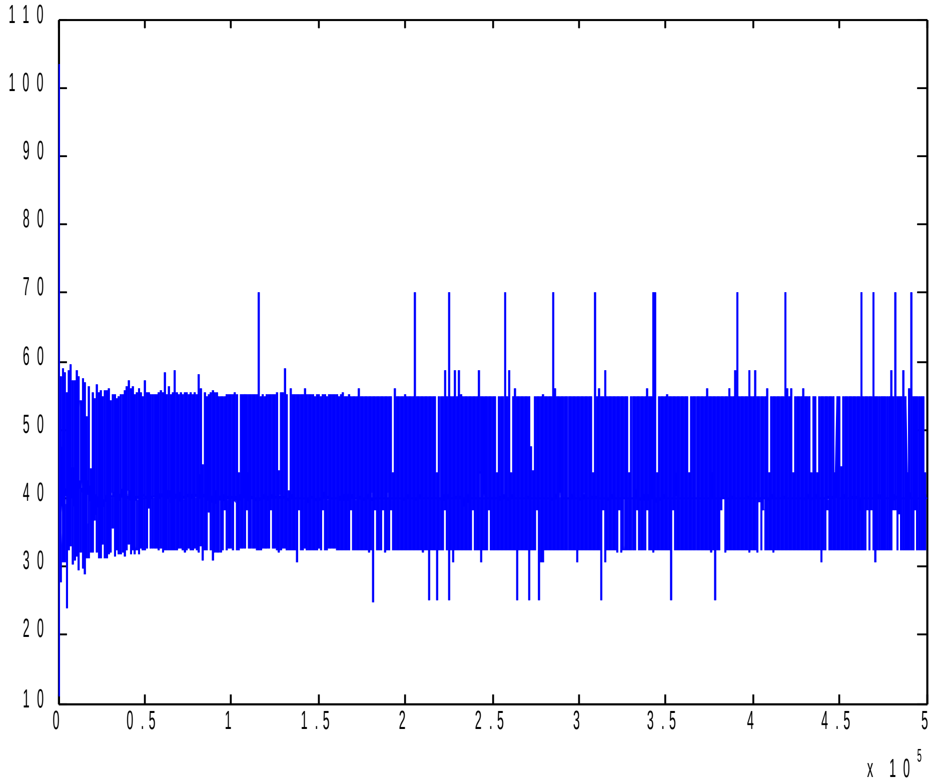
Παρόμοια το co-evolutionary programming αναθεωρείται ως εξής (ο κώδικας Matlab είναι στο παράρτημα Α6):

1. Επέλεξε τις παραμέτρους για τον αριθμό των επιχειρήσεων, τον αριθμό των χρωμοσωμάτων σε κάθε πληθυσμό, την πιθανότητα μετάλλαξης κλπ.
2. Αρχικοποίησε με τυχαίο τρόπο τους i πληθυσμούς (ένας για κάθε επιχείρηση, πάντα) με τυχαίο τρόπο.
3. Επέλεξε τυχαία στρατηγική για κάθε παίκτη, και εκτέλεσε το παίγνιο μεταξύ των επιλεγμένων στρατηγικών. Το κέρδος της κάθε στρατηγικής είναι το κέρδος του παίκτη που την χρησιμοποίησε, στο παίγνιο.
4. Επανέλαβε το βήμα (3) μέχρι να αποδοθεί κέρδος σε όλες τις στρατηγικές
5. Αντέγραψε τα χρωμοσώματα όλων των πληθυσμών σε έναν ενιαίο πληθυσμό, μετεβαλλέ τον με εφαρμογή των τελεστών του γενετικού αλγορίθμου και αντέγραψε τα αντίστοιχα χρωμοσώματα της νέας γενιάς, στους πληθυσμούς των επιχειρήσεων.
6. Επανέλαβε τον βρόχο (3-6) για καθορισμένο αριθμό γενεών.

6.2. Προσομοιώσεις

6.2.1. Γραμμικό υπόδειγμα

Εφαρμόζουμε τον «κοινωνικοποιημένο» αλγόριθμο του Vriend στο υπόδειγμα των 4 επιχειρήσεων. Για 30 χρωμοσώματα των 20 bits στους πληθυσμούς $p_{cr} = 1$, $p_{mut} = 0{,}001$, $GARate = 50$ και 500.000 επαναλήψεις (10.000 γενιές) έχουμε μέση ποσότητα στην αγορά 40,0812 με τυπική απόκλιση 1,4214.



Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,0830	2,8245
2	40,0704	2,7291
3	40,0920	2,8607
4	40,0793	2,6929

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως p_{mut} = 0,1,0,005,0,0001 $GARate$ = 20,100 κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 27.734 από τα 500.000 παίγνια που εκτελέστηκαν. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,261	0,4332	0,2543	0,0515	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

με τον πρώτο πληθυσμό Nash να εμφανίζεται στην 3.749^{η} γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 3,8314 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής.

Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,4663	0,5337		
0,3216	0,6694	0,0088	
	0,0153	0,9803	0,0043
		0,0233	0,9767

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 5742 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης του Vriend με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 40,2697 με τυπική απόκλιση 1,3571. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	39,2516	2,8315
2	37,5449	2,6849
3	39,2165	2,7360
4	45,0440	2,7373

Όπως βλέπουμε ο οι τρεις παίκτες βελτιώθηκαν, όσον αφορά την σύγκλιση στην μέση ποσότητα, αλλά ο τέταρτος παρέμεινε μακράν του ΝΕ. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0 0 0 0 0 0 0 0,0009 0,0306 0,9675 0,001 0 0 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την «κοινωνικοποιημένη» έκδοση του co-evolutionary programming στο ίδιο πρόβλημα με 40 χρωμοσώματα των 20 bits στον κάθε πληθυσμό των 4 επιχειρήσεων, $p_{cr} = 1$, $p_{mut} = 0{,}0005$ και 10.000 γενεές. Η μέση ποσότητα στην αγορά ήταν 40,0241 με τυπική απόκλιση 0,9291. Για τις μεμονωμένες επιχειρήσεις είχαμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	40,0241	1,8169
2	40,0229	1,8163
3	40,0237	1,8662
4	40,0257	1,8113

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως $p_{mut} = 0{,}1,\ 0{,}005,\ 0{,}0001$ $GARate = 20{,}100$ κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 295.250 από τα 400.000 παίγνια που εκτελέστηκαν.

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,143	0,629	0,008	0,085	0,036	0,02	0,049	0,021	0,003	0

8E-04	0	0	0	0	0	0	0	0	0	0

με τον πρώτο πληθυσμό Nash να εμφανίζεται στην 2300η γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 6,9725 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,0333					
0,8898	0,0367				
0,0204	0,9327	0,0429			
	0,1048	0,8667	0,0048		
		0,0571	0,9143		
				0,1250	0,8750

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 3134,67 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 40,0490 με τυπική απόκλιση 0,8915. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	37,5194	1,7297
2	40,1102	1,7069
3	45,0681	1,8079
4	37,4985	1,7741

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων παραμένει

0 0 0 0 0 0 0 0 0,0166 0,9829 0,0005 0 0 0 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του αλγόριθμου του Vriend στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, $p_{mut} = 0{,}0005$, $GArate = 50$ και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, μετά από 1.000.000 επαναλήψεις (20.000 γενεές) η μέση ποσότητα στην αγορά είναι 9,5304 με τυπική απόκλιση 0,1100.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	9,5321	0,4803

2	9,5300	0,4654
3	9,5287	0,4501
4	9,5306	0,4622
5	9,5293	0,4617
6	9,5301	0,4750
7	9,5302	0,4588
8	9,5305	0,4695
9	9,5310	0,4681
10	9,5295	0,4551
11	9,5309	0,4707
12	9,5320	0,4819
13	9,5304	0,4787
14	9,5288	0,4493
15	9,5320	0,4650
16	9,5303	0,4556
17	9,5315	0,4588
18	9,5300	0,4553
19	9,5304	0,4722
20	9,5299	0,4658

ενώ είχαμε ισορροπία κατά Nash στα 889822 από τα 1.000.0000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,1463	0,84815	0,0034	0,0012	0,00095	0	0	0	0

με την πρώτη κατάσταση Nash να εμφανίζεται στην 12000 επανάληψη (240^{η} γενεά). Ο αναμενόμενος χρόνος μεταξύ δυό διαδοχικών μεταβάσεων στην κατάσταση Nash εκτιμάται ως 6,8353 .

Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,1965	0,8035			
0,1386	0,8612	0,0001		
	0,0441	0,9559		
		0,0417	0,9583	
			0,0526	0,9474

Ο αναμενόμενος χρόνος πρώτης «επίσκεψης» στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 137700 επαναλήψεις (2754 γενιές).

Εκτελώντας τον αρχικό αλγόριθμο του Vriend με τις ίδιες παραμέτρους έχουμε μέση ποσότητα στην αγορά 9,5577 με τυπική απόκλιση 0,1064. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	7,0654	0,4316

2	8,9771	0,4452
3	9,0960	0,4308
4	9,1038	0,4713
5	9,8769	0,5092
6	7,0298	0,5122
7	14,3807	0,5081
8	6,9806	0,6265
9	14,3446	0,4963
10	7,0704	0,5102
11	7,0587	0,4462
12	9,8845	0,4522
13	7,0585	0,4899
14	7,0552	0,4431
15	14,3677	0,5163
16	9,0053	0,4566
17	7,0443	0,4865
18	7,0615	0,4387
19	14,3490	0,4817
20	14,3445	0,4550

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0,9992	0,0008	0	0	0

Δεν προέκυψε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του co-evolutionary algorithm στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, $p_{mut} = 0{,}0005$ και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, η μέση ποσότητα στην αγορά μετά από 50.000 γενεές είναι 9,5336 με τυπική απόκλιση 0,1014. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	9,5335	0,4318
2	9,5329	0,4301
3	9,5334	0,4329
4	9,5330	0,4232
5	9,5330	0,4258
6	9,5330	0,4273
7	9,5337	0,4309
8	9,5346	0,4440
9	9,5333	0,4299
10	9,5332	0,4271
11	9,5334	0,4336
12	9,5335	0,4267
13	9,5336	0,4288

14	9,5339	0,4336
15	9,5345	0,4424
16	9,5337	0,4315
17	9,5332	0,4297
18	9,5348	0,4340
19	9,5338	0,4325
20	9,5333	0,4313

ενώ είχαμε ισορροπία κατά Nash στα 856.414 από τα 1.000.000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0,15294	0,79924	0,02258	0,0244	0,00078	6,00E-05	0	0	0

με την πρώτη κατάσταση Nash να προκύπτει στην 2864^{η} γενεά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 6,5385. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,1937	0,8063				
0,1543	0,8454	0,0003			
	0,0106	0,9734	0,0159		
		0,0156	0,9828	0,0016	
			0,0769	0,9231	
				0,3333	0,6667

Ο αναμενόμενος χρόνος πρώτης «επίσκεψης» στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 8321 γενεές.

Εκτελώντας τον αρχικό αλγόριθμο co-evolutionary programming, με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα 9,6625 με τυπική απόκλιση 0,4729. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	9,1021	1,7992
2	7,0998	1,6743
3	7,1033	1,7656
4	9,8540	3,1520
5	7,1093	1,6928
6	8,3561	1,7544
7	8,3340	1,7204
8	15,1159	1,6731
9	8,0457	1,7054
10	7,1022	1,7369
11	7,1091	1,6636
12	7,0981	1,6984
13	15,1191	1,7217
14	7,9665	1,7065
15	8,0077	1,6861
16	7,9754	1,6697
17	7,1177	1,7163
18	7,1106	1,6944
19	30,1584	1,7485
20	8,3655	1,7313

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

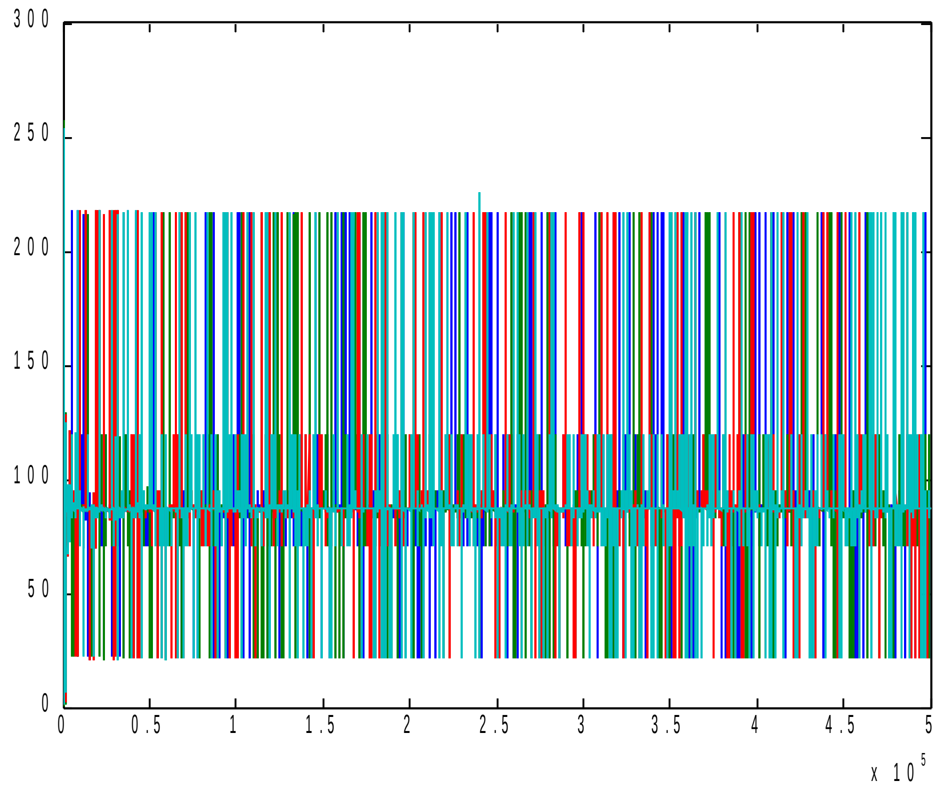
0 0 0 0 1 0 0 0 0

Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

6.2.2. Πολυωνυμικό υπόδειγμα

Εφαρμόζουμε τον κοινωνικοποιημένο αλγόριθμο του Vriend στην περίπτωση των 4 επιχειρήσεων με 40 χρωμοσώματα των 20 bits στον κάθε πληθυσμό

με $p_{cr} = 1$, $p_{mut} = 0{,}00025$, $GARate = 50$ και 500.000 επαναλήψεις. Η μέση ποσότητα στην αγορά ήταν 86,9984 με τυπική απόκλιση 1,7594. Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	86,9991	3,4605
2	86,9905	3,3428
3	86,9994	3,3135
4	87,0046	3,3902

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως p_{mut} = 0,1, 0,005, 0,0001 $GARate$ = 20,100 κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 418.070 από τα 500.000 παίγνια που εκτελέστηκαν. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,6448	0,3286	0,023	0,0036	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

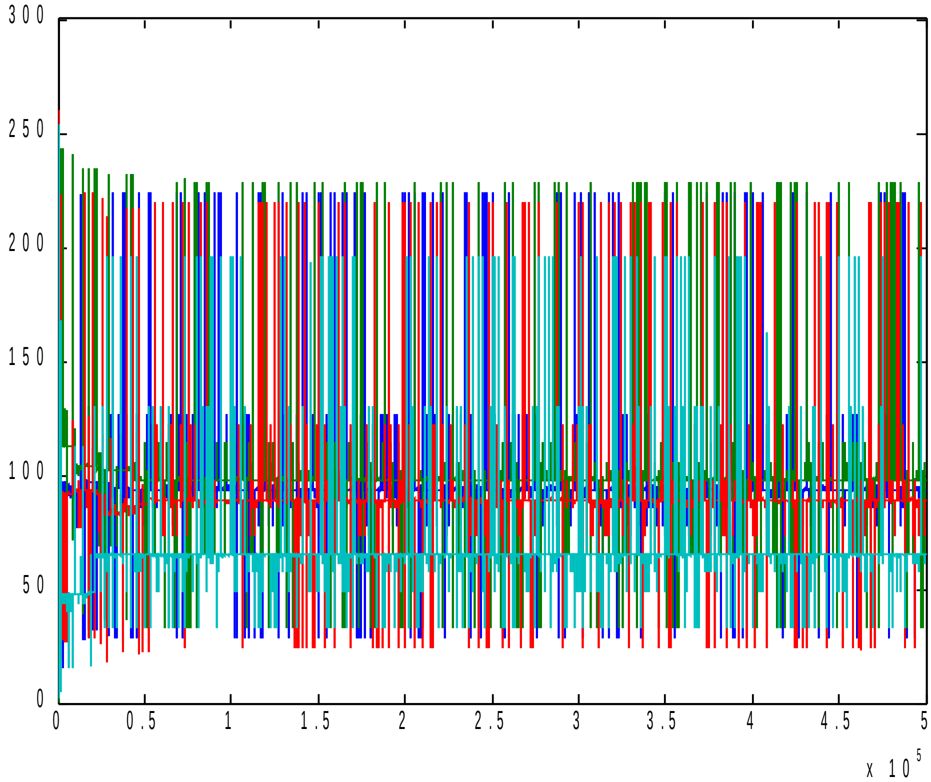
με τον πρώτο πληθυσμό Nash να εμφανίζεται στην $1391^{η}$ γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 1,5509 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής.

Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,7598	0,2402		
0,4714	0,5216	0,0067	
	0,1000	0,8870	0,0130
		0,1111	0,8889

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 6483,5 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης του Vriend με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 86,5465 με τυπική απόκλιση 1,9800. Για τις μεμονωμένες επιχειρήσεις έχουμε



Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	93,7536	3,6042
2	98,4055	4,6625
3	89,4122	3,5453
4	64,6146	5,3246

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων παραμένει

0 0 0 0 0 0 0 0 0,0005 0,9381 0,0172 0,0205 0,0218 0,0019 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του αλγόριθμου του Vriend στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, $p_{mut} = 0{,}0005$, $GArate = 50$ και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, μετά από 1.000.000 επαναλήψεις (20.000 γενεές) η μέση ποσότητα στην αγορά είναι 20,0454 με τυπική απόκλιση 0,3129. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	20,0505	1,3073
2	20,0432	1,2570
3	20,0425	1,2763
4	20,0469	1,3174
5	20,0483	1,3096
6	20,0455	1,2940
7	20,0457	1,2736
8	20,0505	1,3230
9	20,0441	1,2825
10	20,0454	1,2768
11	20,0444	1,2735
12	20,0479	1,2910
13	20,0443	1,2647
14	20,0457	1,2629
15	20,0436	1,2799
16	20,0454	1,2920
17	20,0453	1,2886
18	20,0440	1,2637
19	20,0429	1,2712
20	20,0418	1,2672

ενώ είχαμε ισορροπία κατά Nash στα 676045 από τα 1.000.0000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,10775	0,6581	0,03095	0,1205	0,0742	0,0085	0	0	0

με την πρώτη κατάσταση Nash να εμφανίζεται στην 248800 επανάληψη (4976^{η} γενεά). Ο αναμενόμενος χρόνος μεταξύ δυό διαδοχικών μεταβάσεων στην κατάσταση Nash εκτιμάται ως 9,2807. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,2000	0,8000				
0,1310	0,8684	0,0005			
	0,0129	0,9257	0,0614		
		0,0162	0,9676	0,0162	
			0,0270	0,9616	0,0115
				0,1000	0,9000

Ο αναμενόμενος χρόνος πρώτης «επίσκεψης» στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 189050 επαναλήψεις (3781 γενιές).

Εκτελώντας τον αρχικό αλγόριθμο του Vriend με τις ίδιες παραμέτρους έχουμε μέση ποσότητα στην αγορά 20,0372 με τυπική απόκλιση 0,2331. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	19,8592	1,2679
2	20,5926	1,7733
3	14,8090	0,9332
4	14,8437	0,9679
5	30,1452	0,9098
6	19,7726	0,9524
7	18,7158	1,3609
8	20,7008	1,0810
9	14,7059	1,4341
10	18,6354	0,9993
11	30,1348	0,9719
12	18,5906	1,0235
13	14,8852	1,3212
14	19,5251	0,9298
15	14,8147	1,0370
16	19,4659	1,4168
17	19,7404	1,0759
18	19,7679	0,9556
19	20,7250	0,9734
20	30,3134	1,5047

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0,00665	0,9933	5,00E-05	0	0	0

Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την «κοινωνικοποιημένη» έκδοση του co-evolutionary programming στο ίδιο πρόβλημα με 40 χρωμοσώματα των 20 bits στον κάθε πληθυσμό των 4 επιχειρήσεων, $p_{cr} = 1$, $p_{mut} = 0{,}0005$ και 10.000 γενεές. Η μέση ποσότητα στην αγορά ήταν 87,0058 με τυπική απόκλιση 1,9540. Για τις μεμονωμένες επιχειρήσεις είχαμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	87,0062	3,9258
2	87,0089	3,8091
3	87,0103	3,7425
4	86,9978	3,9229

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως p_{mut} = 0,1, 0,005, 0,0001 $GARate$ = 20,100 κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 241.793 από τα 400.000 παίγνια που εκτελέστηκαν. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,114	0,5199	0,0341	0,0658	0,0659	0,0732	0,1206	0,0035	0,0013	0,001	0,0007
0	0	0	0	0	0	0	0	0	0	0

με τον πρώτο πληθυσμό Nash να εμφανίζεται στην 3700^{η} γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 8,7719 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,2018	0,7982								
0,1750	0,8246	0,0002							
	0,0059	0,9736	0,0205						
		0,0122	0,9529	0,0350					
			0,0364	0,9256	0,0379				
				0,0355	0,9413	0,0232			
					0,0149	0,9826	0,0025		
						0,1143	0,8571	0,0286	
							0,1538	0,8462	
								0,1000	0,9000

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 4067,7 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 86,3391 με τυπική απόκλιση 0,9596. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	97,4890	1,7862
2	74,9728	1,8679
3	82,4704	1,8880
4	90,4242	1,9202

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων παραμένει

0 0 0 0 0 0 0 0 0 0 0,6571 0,3429 0 0 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του co-evolutionary algorithm στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, p_{mut} = 0,0005 και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, η μέση ποσότητα στην αγορά μετά από 50.000 γενεές είναι 20,0234 με τυπική απόκλιση 0,2094.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση

7	20,0228	0,8906
8	20,0223	0,8956
9	20,0223	0,8786
10	20,0229	0,8921
11	20,0248	0,9010
12	20,0234	0,8992
13	20,0239	0,9002
14	20,0235	0,8840
15	20,0238	0,9149
16	20,0222	0,8894
17	20,0235	0,8956
18	20,0240	0,8983
19	20,0232	0,8782
20	20,0247	0,9172

ενώ είχαμε ισορροπία κατά Nash στα 839.356 από τα 1.000.000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0,15084	0,77868	0,05686	0,01222	0,0014	0	0	0	0

με την πρώτη κατάσταση Nash να προκύπτει στην 3777^{η} γενεά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 6,6295. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,2013	0,7987			
0,1547	0,8447	0,0005		
	0,0074	0,9926		
		0,0016	0,9836	0,0147
			0,1429	0,8571

Ο αναμενόμενος χρόνος πρώτης «επίσκεψης» στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 8.297 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο co-evolutionary programming, με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα 20,1135 με τυπική απόκλιση 0,2000.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	0,1227	0,9008
2	17,6630	0,8753
3	18,8022	0,8846
4	30,1194	0,8761
5	30,1251	0,9376
6	14,8341	0,8371

7	14,8147	1,0033
8	17,6500	0,9349
9	22,0752	1,5725
10	30,1616	1,0188
11	18,8537	0,9070
12	16,9406	1,0629
13	30,1507	1,0889
14	14,8014	1,0165
15	14,8274	0,8662
16	14,8273	0,9200
17	14,7766	1,0967
18	14,8343	0,8701
19	18,8849	1,0224
20	17,0052	1,2329

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0,99182	0,00818	0	0	0

Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

6.2.3. Υπόδειγμα με ρητό εκθέτη

Εφαρμόζουμε τον κοινωνικοποιημένο αλγόριθμο του Vriend στην περίπτωση των 4 επιχειρήσεων με 40 χρωμοσώματα των 20 bits στον κάθε πληθυσμό

με $p_{cr} = 1$, $p_{mut} = 0{,}0006$, $GARate = 50$ και 500.000 επαναλήψεις. Η μέση ποσότητα στην αγορά ήταν 82,2976 με τυπική απόκλιση 2,4383. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	82,2918	4,7426
2	82,2956	4,7188
3	82,3147	4,7696
4	82,2882	4,7254

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως $p_{mut} = 0{,}1,\ 0{,}005,\ 0{,}0001$ $GARate = 20{,}100$ κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 40.3910 από τα 500.000 παίγνια που εκτελέστηκαν. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,4162	0,4856	0,0879	0,0103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

με τον πρώτο πληθυσμό Nash να εμφανίζεται στην 1289η γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 2,4027 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,5156	0,4844	
0,4152	0,5842	0,0004

0,0034	0,9875	0,0091
	0,0874	0,9126

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 8171,333 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης του Vriend με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 79.7872με τυπική απόκλιση 2.8020. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	90,2768	5,3025
2	96,6171	5,7789
3	70,5652	12,1758
4	61,6896	5,3746

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων παραμένει

0 0 0 0 0 0 0 0 0,0093 0,0115 0,9715 0,0077 0 0 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την «κοινωνικοποιημένη» έκδοση του co-evolutionary programming στο ίδιο πρόβλημα με 40 χρωμοσώματα των 20 bits στον κάθε πληθυσμό των 4 επιχειρήσεων, $p_{cr} = 1$, $p_{mut} = 0{,}0005$ και 10.000 γενεές. Η μέση ποσότητα στην αγορά ήταν 82,2950 με τυπική απόκλιση 1,9407. Για τις μεμονωμένες επιχειρήσεις είχαμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	82,3005	3,7960
2	82,2903	3,7797
3	82,2961	3,8818
4	82,2931	3,8309

Και για τα υπόλοιπα σετ παραμέτρων που χρησιμοποιήσαμε, όπως $p_{mut} = 0{,}1,\ 0{,}005,\ 0{,}0001$ $GARate = 20{,}100$ κλπ. είχαμε επίσης σύγκλιση των μέσων ποσοτήτων των επιχειρήσεων στην ποσότητα Nash. Για τις συγκεκριμένες τιμές όμως είχαμε και ισορροπία κατά Nash στα 368.454 από τα 400.000 παίγνια που εκτελέστηκαν. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,1715	0,7921	0,0027	0,0026	0,0012	0,0034	0,001	0,002	0,0097	0,0109	0
0	0	0	0	0	0	0	0	0	0	

με τον πρώτο πληθυσμό Nash να εμφανίζεται στην 3742^{η} γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 5,8309 γενιές, που σημαίνει ότι η κατάσταση Nash είναι ιδιαίτερα ευσταθής. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,1983	0,8047		
0,1742	0,8257		
	0,0370	0,9630	
		0,0385	0,9615

0,0833	0,9167					
	0,0294	0,9706				
		0,1000	0,9000			
			0,0500	0,7000	0,2500	
				0,0619	0,9175	0,0206
					0,0275	0,9817

ενώ ο αρχικός χρόνος μετάβασης στην κατάσταση Nash από την αντίθετή της εκτιμάται, μέσω 30 ανεξάρτητων εκτελέσεων του αλγορίθμου, ως 3917,3 γενιές.

Εκτελώντας τον αρχικό αλγόριθμο ατομικής μάθησης με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα στην αγορά 83,4044 με τυπική απόκλιση 0,9507. Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	89,9546	1,8876
2	78,7747	1,8650
3	74,9198	1,8930
4	89,9683	1,8328

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων παραμένει

0 0 0 0 0 0 0 0 0 0 0,0001 0,9835 0,0164 0 0 0 0 0 0 0 0

πολύ μακράν της επιθυμητής κατάστασης. Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του αλγόριθμου του Vriend στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, $p_{mut} = 0,0005$, $GArate = 50$ και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, μετά από 1.000.000 επαναλήψεις (20.000 γενεές) η μέση ποσότητητα στην αγορά είναι 19,3978 με τυπική απόκλιση 0,2348.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	19,4015	0,9856
2	19,3975	0,9557
3	19,3958	0,9456
4	19,3964	0,9436
5	19,3979	0,9678
6	19,3958	0,9906
7	19,3959	0,9487
8	19,4018	0,9987
9	19,3947	0,9765
10	19,3969	0,9688
11	19,3991	0,9771

12	19,3985	1,0148
13	19,3983	0,9690
14	19,3965	0,9602
15	19,4008	0,9683
16	19,3963	0,9558
17	19,3982	0,9952
18	19,3995	0,9724
19	19,3987	0,9641
20	19,3953	0,9693

ενώ είχαμε ισορροπία κατά Nash στα 753404 από τα 1.000.0000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων ήταν

0,12705 0,72965 0,12075 0,01295 0,0096 0 0 0 0

με την πρώτη κατάσταση Nash να εμφανίζεται στην 163250 επανάληψη (3265^{η} γενιά). Ο αναμενόμενος χρόνος μεταξύ δυό διαδοχικών μεταβάσεων στην κατάσταση Nash εκτιμάται ως 7,8709. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,2003	0,7997			
0,1392	0,8586	0,0021		
	0,0128	0,9872		
		0,0039	0,9498	0,0463
			0,0677	0,9323

Ο αναμενόμενος χρόνος πρώτης «επίσκεψης» στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 680.650 επαναλήψεις (13.613 γενιές).

Εκτελώντας τον αρχικό αλγόριθμο του Vriend με τις ίδιες παραμέτρους έχουμε μέση ποσότητα στην αγορά 19,4313 με τυπική απόκλιση 0,2167.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	18,5230	0,9240
2	21,8973	0,9000
3	17,9849	1,0425
4	29,2194	1,0817
5	14,3826	0,9882
6	14,3664	0,8675
7	19,1694	0,9390
8	29,1812	0,9727
9	18,0322	0,9273
10	21,9468	1,0429

11	19,9317	1,6720
12	14,3661	0,9713
13	21,9118	0,9271
14	14,2802	1,1201
15	18,0903	1,1635
16	14,3704	0,8709
17	18,7129	0,9190
18	14,3358	1,1347
19	18,7269	0,9324
20	29,1962	0,9412

ενώ το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0 0 0 0 0,9999 0,0001 0 0 0

Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

Εφαρμόζουμε την κοινωνικοποιημένη έκδοση του co-evolutionary algorithm στο παίγνιο με 20 επιχειρήσεις. Για $p_{cr} = 1$, $p_{mut} = 0{,}0005$ και 20 χρωμοσώματα των 8 bits στον κάθε πληθυσμό, η μέση ποσότητα στην αγορά μετά από 50.000 γενεές είναι 19,4049 με τυπική απόκλιση 0,2092.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	19,4059	0,8924
2	19,4050	0,8800
3	19,4041	0,8666
4	19,4055	0,8597
5	19,4054	0,8779
6	19,4045	0,8786
7	19,4049	0,8778
8	19,4047	0,8850
9	19,4064	0,8795
10	19,4058	0,8924
11	19,4050	0,8649
12	19,4028	0,8616
13	19,4047	0,8778
14	19,4063	0,8847
15	19,4048	0,8701
16	19,4041	0,8637
17	19,4044	0,8679
18	19,4060	0,8769
19	19,4037	0,8363
20	19,4041	0,8628

ενώ είχαμε ισορροπία κατά Nash στα 795.487 από τα 1.000.000 παίγνια. Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0,1425	0,75358	0,09242	0,00838	0,00312	0	0	0	0

με την πρώτη κατάσταση Nash να προκύπτει στην 4.667^{η} γενιά. Ο χρόνος επαναφοράς στην κατάσταση Nash εκτιμάται ως 7,0175. Ο πίνακας των εκτιμητριών των πιθανοτήτων μετάβασης ενός βήματος για τις ομαδικές καταστάσεις που εμφανίστηκαν (ξεκινώντας από την κατάσταση Nash a_0) είναι:

0,2022	0,7978			
0,1509	0,8477	0,0014		
	0,0115	0,9874	0,0011	
		0,0143	0,9785	0,0072
			0,0256	0,9744

Ο αναμενόμενος χρόνος πρώτης μετάβασης στην κατάσταση Nash, με αρχική κατάσταση την αντίθετή της εκτιμάται σε 10452,33 γενεές.

Εκτελώντας τον αρχικό αλγόριθμο co-evolutionary programming, με το ίδιο σετ παραμέτρων έχουμε μέση ποσότητα 19,2605 με τυπική απόκλιση 0,1913.

Για τις μεμονωμένες επιχειρήσεις έχουμε

Επιχείρηση	Μέση ποσότητα	Τυπική απόκλιση
1	0,8613	0,8613
2	0,8290	0,8290
3	0,8295	0,8295
4	0,8212	0,8212
5	0,8271	0,8271
6	0,8454	0,8454
7	0,9127	0,9127
8	0,8148	0,8148
9	0,8049	0,8049
10	0,8026	0,8026
11	0,8494	0,8494
12	0,8687	0,8687
13	0,8355	0,8355
14	0,8546	0,8546
15	0,8788	0,8788
16	0,9079	0,9079
17	0,8984	0,8984
18	0,8767	0,8767
19	0,8643	0,8643
20	0,9832	0,9832

Το εκτιμηθέν διάνυσμα σταθερών συχνοτήτων είναι

0	0	0	0	0,99802	0,00198	0	0	0

Δεν υπήρξε παίγνιο με ισορροπία κατά Nash.

6.3. Συμπεράσματα

Είδαμε λοιπόν ότι οι «κοινωνικοποιημένες» εκδόσεις του αλγόριθμου του Vriend και του co-evolutionary programming οδηγούν, εκτός της σύγκλισης των ποσοτήτων των μεμονωμένων παικτών στην ποσότητα της ισορροπίας κατά Nash, και στην σύγκλιση γενικά του αλγορίθμου κοντά έστω στην κατάσταση με όλα τα χρωμοσώματα ίσα με το χρωμόσωμα που αντιστοιχεί στην ποσότητα αυτή και στην εύρεση συνεπώς της ισορροπίας κατά Nash στα παίγνια Cournot που αναλύσαμε. Πιο συγκεκριμένα, οι μέσες ποσότητες που επιλέγουν οι μεμονωμένοι πράκτορες κατά τη διάρκεια μιας προσομοίωσης, συνέκλιναν στην ποσότητα που αντιστοιχεί στην ισορροπία κατά Nash, με πολύ μεγάλη ακρίβεια, σε όλες τις προσομοιώσεις που εκτελέσαμε. Ο λόγος που συνέβη αυτό είναι φυσικά η κοινωνική αντί της ατομικής μάθησης. Απ’ την στιγμή που η επόμενη γενιά για όλους τους πληθυσμούς, καθορίζεται από την εφαρμογή των τελεστών του γενετικού αλγορίθμου στον συνολικό πληθυσμό που προκύπτει από την ένωση των επιμέρους πληθυσμών, ο γενετικός αλγόριθμος οδηγεί στην ομογενοποίηση του ενιαίου αυτού πληθυσμού και συνεπώς των μεμονωμένων πληθυσμών. Αυτό βεβαίως σημαίνει ότι, εφόσον ο αριθμός των γενεών είναι επαρκής, η αλυσίδα Markov που χαρακτηρίζει την εξέλιξη του συστήματος, θα βρεθεί στην «κατάσταση Nash» και συνεπώς, ο αλγόριθμος θα «ανακαλύψει» το σημείο ισορροπίας. Όπως όμως παρατηρήσαμε, στα συγκεκριμένα σύνολα παραμέτρων που πραγματοποιήθηκε ο συγκεκριμένος στόχος, ο αριθμός επαναλήψεων που απαιτείται είναι ιδιαίτερα αυξημένος (κοντά ή και πάνω από 10.000 γενιές). Ο αριθμός αυτός των επαναλήψεων εξαρτάται βέβαια σε πολύ μεγάλο βαθμό από τις πιθανότητες μετάλαξης, διασταύρωσης και -το σημαντικότερο- από τον αριθμό των επιχειρήσεων και των δυαδικών ψηφίων στα χρωμοσώματα των πληθυσμών των παικτών. Έτσι, για παράδειγμα, δεν μπορέσαμε να βρεθούμε σε ισορροπία κατά Nash στην περίπτωση 20 επιχειρήσεων με 20-μπιτα χρωμοσώματα, με δεδομένη την υπολογιστική που είχαμε στη διάθεσή μας, ενώ από την άλλη, η προσομοίωση του γραμμικού υποδείγματος 4 επιχειρήσεων με 6-μπιτα χρωμοσώματα που πραγματοποιήσαμε συνέκλινε αρκετά γρήγορα (δεν αναλύεται εδώ).

Οι συνθήκες ευστάθειας των αλγορίθμων, φαίνονται από τις αναμενόμενες συχνότητες των ομαδικών (lumped) καταστάσεων και τους αναμενόμενους χρόνους επιστροφής στην κατάσταση Nash. Οι χρόνοι επιστροφής που παρατηρήσαμε στις προσομοιώσεις είναι ιδιαίτερα ικανοποιητικοί (λιγότερο από 10 γενιές). Βέβαια οι χρόνοι αυτοί εξαρτώνται άμεσα από τις παραμέτρους και κυρίως από την πιθανότητα μετάλλαξης. Οι εκτιμήσεις για τις αναμενόμενες συχνότητες των ομαδικών καταστάσεων έδειξαν -στις περιπτώσεις που αναλύσαμε παραπάνω- ότι η πλέον συχνή μεμονωμένη κατάσταση είναι η κατάσταση Nash, ενώ η πλέον συχνή ομαδική κατάσταση ήταν η s_1, η ομαδική κατάσταση δηλαδή που αποτελείται από το σύνολο των μεμονωμένων καταστάσεων στις οποίες η μέση απόσταση των χρωμοσωμάτων όλων των πληθυσμών είναι μικρότερη ή ίση με 1. Πιο συγκεκριμένα, αν λάβουμε υπόψη ότι η αθροιστική συχνότητα των δύο αυτών ομαδικών καταστάσεων s_0 και s_1 είναι βέβαια το άθροισμα των επιμέρους συχνοτήτων, συμπεραίνουμε ότι τα χρωμοσώματα των πληθυσμών των παικτών θα απέχουν -κατά μέσο όρο- από το χρωμόσωμα που αντιστοιχεί στην ισορροπία κατά Nash κατά ένα ή και λιγότερο δυαδικό ψηφίο, στο 90% τουλάχιστον των γενεών που δημιουργούνται κατά τη διάρκεια εκτέλεσης των αλγορίθμων. Για παράδειγμα η αθροιστική συχνότητα των s_0 και s_1 στην προσομοίωση του αλγόριθμου του Vriend που παρουσιάσαμε για το γραμμικό υπόδειγμα 4 παικτών με 20-μπιτα χρωμοσώματα ήταν $n_0 + n_1 = 90.018/100$ γενεές. Ανάλογα ήταν τα αποτελέσματα για τα υπόλοιπα μοντέλα και το «κοινωνικοποιημένο» co-evolutionary programming. Η συμπεριφορά λοιπόν των αλγορίθμων είναι ιδιαίτερα ευσταθής (με την έννοια τουλάχιστον της ευστάθειας Ljapunov [Riechmann 1997]).

Από άποψης οικονομικής θεωρίας το παραπάνω πόρισμα είναι ιδιαίτερα σημαντικό, αφού υποδηλώνει ότι -στα πλαίσια των παραπάνω υποδειγμάτων- η ατομική μάθηση, παρόλο που οδηγεί, όπως είδαμε στο προηγούμενο κεφάλαιο, σε μέση ποσότητα (και άρα ποσότητα και τιμή στην αγορά) ίση με την ποσότητα Nash, δεν οδηγεί στην αποτελεσματική σύγκλιση όλων των παικτών στην παραπάνω ποσότητα. Οι παίκτες δηλαδή δεν μαθαίνουν αποτελεσματικά. Αντίθετα, οι εκδόσεις κοινωνικής μάθησης των παραπάνω αλγορίθμων που προτείναμε εδώ, λύνουν το πρόβλημα. Συνεπώς η ετερογένεια που αναφέραμε στο προηγούμενο κεφάλαιο, που παρατηρείται στην συμπεριφορά των μεμονωμένων παικτών στις αρχικές εκδόσεις ατομικής μάθησης, αίρεται στην «κοινωνικοποιημένη» έκδοση των αλγορίθμων. Εδώ, τόσο τα μακροοικονομικά όσο και τα ατομικά μεγέθη, συγκλίνουν -έστω και κατά προσέγγιση, στη γενική περίπτωση- στην ισορροπία κατά Nash.

Το γεγονός αυτό της ομογενούς σύγκλισης έχει συνέπειες και στο ζήτημα της ισοκατανομής του εισοδήματος, στην τεχνητή οικονομία που περιγράφεται από το παίγνιο Cournot και γενικότερα το υπόδειγμα που μοντελοποιήσαμε, μέσω των γενετικών αλγορίθμων μάθησης. Από την στιγμή που, στην περίπτωση ατομικής μάθησης, οι επιλογές των πρακτόρων διαφέρουν μεταξύ τους, το ίδιο θα συμβαίνει και με το κέρδος τους. Έτσι θα υπάρχει μεγαλύτερη ανισοκατανομή του εισοδήματος στην περίπτωση αυτή (η τυπική απόκλιση και ο συντελεστής Gini θα έχουν υψηλότερη τιμή). Αντίθετα, η μίμηση των επιτυχημένων συμπεριφορών των υπολοίπων πρακτόρων, οδηγεί σε μεγαλύτερη ισονομία στην κατανομή του κέρδους. Η εξίσωση βέβαια δεν μπορεί να είναι πλήρης, αφενός λόγο του ότι θα πρέπει να έχουμε κάποια συγκεκριμένα σύνολα παραμέτρων στον γενετικό αλγόριθμο για να έχουμε σχετικά γρήγορη σύγκλιση στην ισορροπία κατά Nash και αφετέρου γιατί -ακόμα και στην περίπτωση που συγκλίνουμε στην ισορροπία- θα υπάρχουν πάντα τυχαίες διακυμάνσεις, λόγο της φύσης του οποιουδήποτε γενετικού αλγορίθμου.

Όσον αφορά τώρα τη χρήση των παραπάνω αλγορίθμων σαν ευρετικών αλγορίθμων για την επίλυση παιγνίων Cournot, ανακύπτει το ερώτημα αναφορικά με τον τρόπο ανίχνευσης της σωστής ποσότητας σε παίγνια με άγνωστη -λογικά- ποσότητα ισορροπίας. Δεδομένου ότι οι αλγόριθμοι συγκλίνουν σε καταστάσεις με χρωμοσώματα με μικρότερη μέση απόσταση από 1-2 bits από την ποσότητα ισορροπίας, αυτό λογικά σημαίνει ότι το χρωμόσωμα ισορροπίας θα βρίσκεται και μάλιστα με μεγάλη συχνότητα στους πληθυσμούς των τελευταίων γενεών. Αλλά και τα υπόλοιπα χρωμοσώματα των πληθυσμών θα απέχουν κατά έναν ιδιαίτερα μικρό αριθμό bits από το χρωμόσωμα Nash. Ακόμα λοιπόν κι αν οι αλγόριθμοι δεν περνάνε ποτέ στην κατάσταση Nash (για το δεδομένο σύνολο παραμέτρων και τον αριθμό γενεών που χρησιμοποιείται), μπορεί κανείς να απομονώσει εκείνα τα χρωμοσώματα που μπορούν να θεωρηθούν σαν «υποψήφιες ισορροπίες» και -δεδομένης της συμμετρίας του παιγνίου Cournot- να εξετάσει το πρόβλημα βελτιστοποίησης του μεμονωμένου παίκτη, με δεδομένο ότι οι $n-1$ συμπαίκτες του υιοθετούν δεδομένη υποψήφια ποσότητα. Το πρόβλημα είναι πρόβλημα μεγιστοποίησης μίας μεταβλητής και λογικά, εύκολα επιλύσιμο (στην περίπτωση των μοντέλων που χρησιμοποιήσαμε είναι απλό πρόβλημα κυρτού προγραμματισμού μίας μεταβλητής- σε πιο σύνθετες περιπτώσεις ίσως απαιτηθεί η χρήση ενός ευρετικού, ή και γενετικού αλγόριθμου ακόμα). Εφόσον η βέλτιστη ποσότητα συμπίπτει με την υποψήφια ποσότητα που χρησιμοποιήσαμε τότε, εξ΄ ορισμού, έχουμε ισορροπία κατά Nash.

Μια καλύτερη μέθοδος, που είναι η πλέον κατάλληλη στην περίπτωση που ο αλγόριθμος «βρίσκει» την ισορροπία κατά Nash, στην περίπτωση δηλαδή που σε κάποιο από τα παίγνια της προσομοίωσης όλοι οι παίκτες χρησιμοποιούν τη στρατηγική Nash, είναι η «καταγραφή» από τον κώδικα των παιγνίων που πραγματοποιούνται κατά τη διάρκεια της προσομοίωσης και στα οποία όλοι οι παίκτες χρησιμοποιούν την ίδια στρατηγική. Εφόσον το παίγνιο είναι συμμετρικό, όλοι οι παίκτες θα επιλέγουν την ίδια ποσότητα στην ισορροπία. Στη συνέχεια μπορούν να χρησιμοποιηθούν μόνο οι στρατηγικές εκείνες που έχουν χρησιμοποιηθεί από κοινού από όλους τους παίκτες, σαν υποψήφιες ισορροπίες κατά Nash, στη διαδικασία επαλήθευσης που περιγράψαμε παραπάνω.

7. Γενετικός αλγόριθμος διαδοχικών βέλτιστων απαντήσεων

7.1. Εισαγωγή

Σε αυτήν την ενότητα, θα εισάγουμε έναν αλγόριθμο εύρεσης του σημείου ισορροπίας κατά Nash, που βασίζεται σε γενετικούς αλγορίθμους. Σε αντίθεση με τις προηγούμενες περιπτώσεις, στην συγκεκριμένη προσέγγιση οι γενετικοί αλγόριθμοι χρησιμοποιούνται ως εργαλείο βελτιστοποίησης και όχι ως εργαλείο μάθησης. Ο αλγόριθμος αποτελεί μια ειδική περίπτωση των λεγόμενων επαναληπτικών αλγόριθμων αναζήτησης της ισορροπίας κατά Nash (Son and Baldick 2004).

Οι αλγόριθμοι αυτοί βασίζονται στην σύγκλιση των ακολουθιών «βέλτιστων απαντήσεων». Όπως είδαμε στο κεφάλαιο 2, τόσο για την περίπτωση του δυοπωλίου Cournot, όσο και για γενικότερα παίγνια, μπορούμε να εξετάσουμε την ακολουθία «βέλτιστων απαντήσεων» των παικτών και να δούμε αν συγκλίνει στην ισορροπία κατά Nash. Εκεί παραθέσαμε και διάφορες ικανές συνθήκες για την σύγκλιση της ακολουθίας στο σημείο ισορροπίας, καθώς και τους ορισμούς της ασυμπτωτικά σταθερής κατάστασης, της ασθενούς ευστάθειας κλπ.

Στην περίπτωση, τώρα που η διαδικασία αυτή των βέλτιστων απαντήσεων, για ένα γενικό παίγνιο n παικτών συγκλίνει σε ισορροπία κατά Nash σε καθαρές στρατηγικές σε πεπερασμένο αριθμό βημάτων (όπως για παράδειγμα στα παίγνια με ordinal best-reply potential, βλ. Dubey et al 2006), μπορούμε να χρησιμοποιήσουμε επαναληπτικούς αλγορίθμους, για να βρούμε τη βέλτιστη απάντηση του παίκτη σε κάθε βήμα. Τέτοιοι αλγόριθμοι εφαρμόστηκαν για παράδειγμα από τους Hobbs et al. (2000) και Weber et al (1999) σε προβλήματα αγορών ηλεκτρικής ενέργειας. Ένα σημαντικό ζήτημα που ανακύπτει στην πράξη, στην χρησιμοποίηση τέτοιων αλγορίθμων, είναι ο τρόπος με τον οποίο γίνεται η αναζήτηση της βέλτιστης απάντησης. Αν χρησιμοποιηθεί ένας αλγόριθμος τοπικής αναζήτησης, όπως η μέθοδος Newton (Weber et al 1999) ή μια αναζήτηση interior point με penalization (Hobbs et al. 2000), οι Son and Baldick (2004) αποδεικνύουν ότι ο αλγόριθμος μπορεί να «κολλήσει» σε τοπικό ελάχιστο, εφόσον υπάρχει τέτοιο στην συνάρτηση κέρδους κάποιου παίκτη. Δείχνουν μάλιστα, μέσω παιγνίων 2 παικτών, τέτοιες περιπτώσεις.

Αν από την άλλη χρησιμοποιήσουμε έναν κατάλληλο γενετικό αλγόριθμο για την αναζήτηση της βέλτιστης απάντησης, η ακολουθία των καταστάσεων, όπως προκύπτουν από τις λύσεις που προτείνουν οι γενετικοί αλγόριθμοι, θα πρέπει να συγκλίνει -κατά πιθανότητα- στην ισορροπία. Μάλιστα, θα δούμε ότι, λόγω των ιδιοτήτων της ακολουθίας των πιθανών καταστάσεων (εργόδικη αλυσίδα Markov), θα βρεθούμε ούτως ή άλλως σε κατάσταση ισορροπίας κατά Nash, με πίθανότητα που τείνει στο 1, καθώς ο αριθμός των βημάτων τείνει στο άπειρο.

Σημειώνουμε τέλος, ότι το μοντέλο Cournot είναι ένα aggregate game (Dubey et al. 2006), η μοναδική δηλαδή σημαντική παράμετρος στην επιλογή ενός παίκτη -η τιμή- δίνεται από το σύνολο των στρατηγικών των συμπαικτών του. Μπορούμε επίσης να χρησιμοποιήσουμε διαφορετικές υποθέσεις για την αναμενόμενη μεταβολή των αντιδράσεων των συμπαικτών στην ακολουθία διαδοχικών απαντήσεων. Αντί δηλαδή η βέλτιστη απάντηση να προσδιορίζεται με βάση την υπόθεση ότι οι συμπαίκτες θα χρησιμοποιήσουν την ίδια ποσότητα στον επόμενο γύρο (οπότε η αναμενόμενη τιμή που θα χρησιμοποιηθεί στην αναζήτηση της βέλτιστης λύσης της $\pi_i = Pq_i - c_i(q_i)$ θα είναι η $P = P_{t-1}$), μπορούμε να χρησιμοποιήσουμε για παράδειγμα την υπόθεση ότι οι υπόλοιποι παίκτες θα προσφέρουν ποσότητες που θα είναι ίσες με τον μέσο όρο των ποσοτήτων που προσέφεραν στους προηγούμενους γύρους οπότε $q_1^t = r_1\left(\sum_{\tau=0}^{t-1} \frac{q_2^\tau}{t}\right)$ στην περίπτωση του δυοπωλίου. Τέτοιες υποθέσεις μπορούν να μας οδηγήσουν σε σύγκλιση της ακολουθίας βέλτιστων απαντήσεων στην ισορροπία κατά Nash σε ένα ευρύτερο σύνολο παραμέτρων του προβλήματος, από ότι η υπόθεση $q_1 = r(q_2)$ (Fundenberg and Tirole 1991). Απ' την στιγμή μάλιστα που το παίγνιο Cournot είναι «συνολικό παίγνιο» (aggregate game), η μοναδική σημαντική παράμετρος όσον αφορά τον προσδιορισμό της βέλτιστης αντίδρασης ενός παίκτη είναι η αναμενόμενη τιμή P_e και όχι οι στρατηγικές των μεμονωμένων παικτών. Άρα μπορούμε να χρησιμοποιήσουμε διαφορετικές υποθέσεις για τις προσδοκώμενες τιμές (βλ. και Arifovic 1994) και να διερευνήσουμε την σύγκλιση του αλγορίθμου αναζήτησης μας στα διαφορετικά ενδεχόμενα.

7.2. Το μοντέλο

Έστω ότι έχουμε ένα παίγνιο n παικτών, και ξεκινάμε από ένα προφίλ καθαρών στρατηγικών

$$s^{1,1} = \left(s_1^1, \ldots, s_n^1\right)$$

Ξεκινάμε από τον παίκτη 1 και χρησιμοποιούμε έναν γενετικό αλγόριθμο για να βρούμε την -ενδεχομένως- βέλτιστη απάντησή του στην κατάσταση $s_{-1}^{1,1}$ που διαμορφώνεται παραπάνω. Η συνάρτηση προσαρμοστικότητας θα είναι καταταγμένη ανάλογα με το κέρδος του παίκτη 1 με δεδομένες τις στρατηγικές των άλλων παικτών (ordered fitness)

$$s_{-1}^{1,1} = \left(s_2^1, \ldots, s_n^1\right)$$

Επιλέγουμε σαν στρατηγική αντίδρασης για τον παίκτη 1, την καθαρή στρατηγική που δίνεται από το χρωμόσωμα με την καλύτερη συνάρτηση προσαρμοστικότητας, που βρέθηκε σε οποιαδήποτε γενιά του αλγόριθμου (το λεγόμενο "best so far"). Η στρατηγική που προκύπτει για τον παίκτη 1, s_1^2 μας οδηγεί στην κατάσταση

$$s^{1,2} = \left(s_1^2,\ldots,s_n^2\right)$$

Ακολουθούμε την ίδια διαδικασία, για να προσδιορίσουμε την στρατηγική αντίδρασης για τον παίκτη 2 κ.ο.κ.

Τελικά καταλήγουμε σε μια κατάσταση

$$s^{m,m} = \left(s_1^m,\ldots,s_n^m\right)$$

κατά την οποία, οι στρατηγικές που επιλέγονται για τους n παίκτες παραμένουν οι ίδιες, σε σχέση με την κατάσταση $s^{m-1,m-1}$. Αυτό θα μπορούσε να είναι και το κριτήριο τερματισμού του αλγορίθμου μας. Δεδομένου όμως, ότι η όλη διαδικασία αποτελεί μια αλυσίδα Markov, υπάρχει μια -μικρή αλλά θετική- πιθανότητα, το προφίλ αυτό να μην αποτελεί ισορροπία κατά Nash. Γι αυτό ίσως είναι σκόπιμο να χρησιμοποιήσουμε σαν κριτήριο τερματισμού, τη μη μεταβολή του προφίλ στρατηγικών για $k > 1$ αριθμό (συνολικών, για όλους τους παίκτες) επαναλήψεων. Έτσι τερματίζουμε την αναζήτηση όταν

$$s^{m,m} = s^{m-k,m-k} \quad k \geq 1$$

Αυτό μειώνει σημαντικά την πιθανότητα να καταλήξουμε σε ένα προφίλ που δεν είναι ισορροπία κατά Nash, η οποία είναι συνεπώς ιδιαίτερα μικρή.

Ένα ιδιαίτερα σημαντικό ζήτημα ανακύπτει σχετικά με το σύνολο τιμών των χρωμοσωμάτων. Μια αναγκαία συνθήκη για να καταλήξουμε στο ΝΕ είναι, φυσικά, οι ποσότητες όλων των παικτών να είναι μέρος του εφικτού συνόλου τιμών των χρωμοσωμάτων, όπως αυτό καθορίζεται με τη διαδικασία αποκωδικοποίησης που χρησιμοποιήσαμε σε προηγούμενο κεφάλαιο και που θα συνεχίσουμε να χρησιμοποιούμε κι εδώ. Αυτό δεν είναι πάντα δυνατό. Αφενός μπορεί ο διαμερισμός του διαστήματος μεταξύ των $Q_{\min}$ και $Q_{\max}$ να μην είναι αρκετά «πυκνός», ο αριθμός των δυαδικών ψηφίων στα χρωμοσώματα και ο συνεπαγόμενος συντελεστής μετατροπής του χρωμοσώματος σε πραγματικές ποσότητες να μην επαρκούν ώστε να καλυφθεί το διάστημα με την απαιτούμενη ακρίβεια. Μια λύση σε αυτό το πρόβλημα θα μπορούσε να είναι η εισαγωγή ενός εξωτερικού βρόχου, ο οποίος θα αναπροσαρμόζει τις ελάχιστες και μέγιστες τιμές, ουτοσώστε να διευρύνεται η ακρίβεια στις μεταγενέστερες επαναλήψεις του. Αυτός ο βρόχος θα μπορούσε να συνεχίζεται για όσες επαναλήψεις θέλουμε, εφόσον όμως το εύρος των βέλτιστων απαντήσεων -και συνεπώς οι ελάχιστες και μέγιστες τιμές του συνόλου τιμών των χρωμοσωμάτων του κάθε παίκτη- συνεχώς μειώνεται. Αυτό δεν ισχύει πάντα και σίγουρα κάτι τέτοιο δεν είναι εφικτό στις περιπτώσεις που η ακολουθία αποκλίνει. Τέλος σε περιπτώσεις όπως π.χ. που κάποιο q_i^N είναι άρρητος, θα πρέπει να συνεχίσουμε επ΄ άπειρον.

Ένα άλλο σημαντικό ζήτημα είναι να σιγουρευτούμε με όσο το δυνατόν μεγαλύτερη πιθανότητα, ότι πραγματικά οι καθαρές στρατηγικές του προφίλ στο οποίο καταλήγει ο αλγόριθμος, ακόμα κι αν αυτές είναι ευσταθείς, είναι οι βέλτιστες απαντήσεις για καθένα από τους παίκτες, με δεδομένες τις στρατηγικές των συμπαικτών του (οπότε, εξορισμού το προφίλ είναι ισορροπία κατά Nash). Θα μπορούσαμε λοιπόν, είτε να εισάγουμε τις στρατηγικές αυτές στο μοντέλο και να ελέγξουμε αν ικανοποιούνται οι συνθήκες πρώτης και δεύτερης τάξης (στις περιορισμένες περιπτώσεις που είναι εφικτό κάτι τέτοιο), είτε να χρησιμοποιήσουμε κάποιον αλγόριθμο τοπικής αναζήτησης ή κάποιο ευρετικό αλγόριθμο για να ελέγξουμε αν μπορούμε να βρούμε κάποια στρατηγική που να βελτιώνει το κέρδος κάποιου συγκεκριμένου παίκτη, με δεδομένη την επιλογή των συμπαικτών του. Ο ίδιος γενετικός αλγόριθμος που χρησιμοποιούμε για να προσδιορίσουμε τις απαντήσεις των παικτών, θα μπορούσε επίσης να χρησιμοποιηθεί. Βέβαια για να έχει νόημα κάτι τέτοιο -αφού θα έχει ήδη καταλήξει στο συγκεκριμένο προφίλ στρατηγικών- θα πρέπει να αυξήσουμε το μέγεθος του πληθυσμού, ώστε να αυξηθεί έτσι η πιθανότητα εντοπισμού κάποιας στρατηγικής με μεγαλύτερο payoff, που δεν είχε εντοπιστεί προηγουμένως.

Σημειώνουμε τέλος ότι στην περίπτωση που έχουμε συμμετρικό παίγνιο (όπως το παίγνιο Cournot με συμμετρικές συναρτήσεις κόστους) και ταυτόχρονες απαντήσεις, ο αλγόριθμος μπορεί να απλοποιηθεί ακόμα περισσότερο. Έστω ότι βρισκόμαστε σε μία κατάσταση, στην οποία όλοι οι παίκτες ακολουθούν την ίδια στρατηγική

$$s^1 = (s, \ldots, s)$$

Εφόσον η στρατηγική s' είναι η βέλτιστη απάντηση για τον παίκτη i, το ίδιο θα ισχύει, λόγω συμμετρίας και για όλους τους υπόλοιπους παίκτες $j \neq i$. Αν τώρα η ακολουθία αυτών των ταυτόχρονα προσδιορισμένων βέλτιστων απαντήσεων, συγκλίνει στην ισορροπία κατά Nash, μπορούμε να χρησιμοποιήσουμε ένα μονάχα γενετικό αλγόριθμο, που να προσδιορίζει τη βέλτιστη απάντηση s^* σε δεδομένο προφίλ s^i και η οποία να δίνει το προφίλ

$$s^{i+1} = (s^*, \ldots, s^*)$$

για όλους τους παίκτες.

Όπως έχουμε αναφέρει και στο κεφάλαιο 2, ένας γενετικός αλγόριθμος στον οποίο ο τελεστής που εφαρμόζεται τελευταίος είναι ο τελεστής μετάλλαξης, χαρακτηρίζεται από μια στοχαστική διαδικασία, όσον αφορά τις καταστάσεις στις οποίες βρίσκεται ο πληθυσμός των χρωμοσωμάτων του, η οποία αποτελεί μια εργοδική αλυσίδα Markov (regular time-homogeneous ergodic Markov Chain). Αυτό σημαίνει καταρχήν ότι όλες οι καταστάσεις (πληθυσμοί) είναι προσπελάσιμοι από τον γενετικό αλγόριθμο, και ότι -καθώς το $t \to \infty$ - η πιθανότητα να βρεθεί ο αλγόριθμος τουλάχιστον μία φορά σε κάποια δεδομένη κατάσταση, τείνει στο 1. Αυτό βέβαια ισχύει και για τους πληθυσμούς εκείνους που περιλαμβάνουν το χρωμόσωμα-στρατηγική που αποτελεί τη βέλτιστη απάντηση, στο συγκεκριμένο προφίλ στρατηγικών s^{t}_{-i}. Δεδομένου τώρα, ότι το τελικό αποτέλεσμα του αλγόριθμου θα είναι το καλύτερο χρωμόσωμα που βρέθηκε ποτέ, σε όλες τις γενεές που εξετάζονται, έχουμε

$$P\left(s = s^{best}_{-i}\right) \to 1$$

Στην πράξη βέβαια, θέλουμε να βρούμε τη βέλτιστη απάντηση όσο το δυνατόν συντομότερα. Αυτό εξαρτάται αφενός από το συγκεκριμένο πρόβλημα και το συγκεκριμένο σημείο από το οποίο ξεκινήσαμε και αφετέρου από τις παραμέτρους του γενετικού αλγορίθμου (μέγεθος πληθυσμού, πιθανότητα μετάλλαξης κλπ.). Τίθεται και το πρόβλημα της ακρίβειας της λύσης, όπως αναφέραμε προηγουμένως. Στην περίπτωση πάντως ενός κανονικού γενετικού αλγορίθμου (επιλογή γονέων ανάλογη με την προσαρμοστικότητά τους, εφαρμογή τελεστή διασταύρωσης μονού τυχαίου σημείου και τελικά εφαρμογή του τελεστή μετάλλαξης με σταθερή πιθανότητα καθόλη τη διάρκεια του αλγορίθμου) , έχουμε κανονική αλυσίδα Markov και μπορούμε να εκτιμήσουμε τόσο τις συχνότητες εμφάνισης της βέλτιστης απάντησης, όσο και τον αναμενόμενο χρόνο πρώτης «επίσκεψης» στην λύση αυτή, από τις παρατηρήσεις, κατά τη διάρκεια εκτέλεσης του αλγορίθμου, κατά τη μέθοδο του προηγούμενου κεφαλαίου.

Η συνάρτηση προσαρμοστικότητας στον γενετικό αλγόριθμο είναι, στην περίπτωση διατεταγμένης προσαρμοστικότητας, η κατάταξη των στρατηγικών, ανάλογα με το payoff του παίκτη i από την κάθε στρατηγική, με δεδομένες τις στρατηγικές των υπολοίπων παικτών s^{*}_{-i}. Δεδομένου ότι η συνάρτηση προσαρμόστικότητας του γενετικού αλγορίθμου εξαρτάται από τη δεδομένη κατάσταση στην οποία βρίσκεται ο αλγόριθμος αναζήτησης, το ίδιο θα ισχύει και για τις πιθανότητες κατάληξής του σε κάποια συγκεκριμένη λύση -ενδεχόμενη βέλτιστη απάντηση (για τις ακριβείς εκφράσεις για τις πιθανότητες μετάβασης του γενετικού αλγορίθμου βλ. κεφ. 2). Αυτό σημαίνει ότι οι πιθανότητες μετάβασης, για τον αλγόριθμο αναζήτησης διαδοχικών βέλτιστων απαντήσεων, εξαρτώνται από την τρέχουσα κατάσταση στην οποία βρίσκεται. Δεν εξαρτώνται από το χρόνο -η συνάρτηση προσαρμοστικότητας θα είναι η ίδια κάθε φορά που εκτελείται ο γενετικός αλγόριθμος για ένα δεδομένο προφίλ s^{*}_{-i} και δεν εξαρτώνται επίσης και από τις προηγούμενες καταστάσεις που βρισκόταν ο αλγόριθμος στο παρελθόν (μονάχα από τις πιθανότητες κατάληξης του γενετικού αλγορίθμου, για τις διάφορες πιθανές λύσεις, επαναλαμβάνουμε). Τέλος, δεδομένου ότι ένας γενετικός αλγόριθμος με τελευταίο τελεστή αυτόν της μετάλλαξης, όπως ο κανονικός γενετικός αλγόριθμος που χρησιμοποιούμε, εξελίσσεται ως εργόδικη αλυσίδα Markov, μπορούμε να έχουμε, σαν αποτέλεσμα, την οποιαδήποτε στρατηγική (με την πιθανότητα βέβαια να είναι ιδιαίτερα μικρή, σε όλες τις περιπτώσεις, εκτός από την ισορροπία κατά Nash). Έτσι μπορούμε να πούμε τελικά, ότι η στοχαστική διαδικασία του αλγόριθμου διαδοχικών απαντήσεων, συνολικά, είναι εργόδικη (και μάλιστα κανονική στην περίπτωση κανονικού αλγορίθμου) αλυσίδα Markov, χωρίς βέβαια να λαμβάνεται υπόψη το κριτήριο τερματισμού, που θα τερματίσει την στοχαστική διαδικασία του αλγορίθμου, σε συγκεκριμένη χρονική στιγμή και σε συγκεκριμένη κατάσταση (που ελπίζουμε ότι θα είναι το σημείο ισορροπίας κατά Nash).

7.3. Τα υποδείγματα

Θα χρησιμοποιήσουμε το γραμμικό υπόδειγμα που χρησιμοποιήσαμε στα προηγούμενα κεφάλαια, σαν αρχική δοκιμή και το πολυωνυμικό υπόδειγμα και το υπόδειγμα με ρητό εκθέτη με διαφορετικές όμως παραμέτρους στη συνάρτηση κόστους του κάθε παίκτη, τέτοιες ώστε οι συνθήκες των Dubey et al (2006) να παραμένουν εν ισχύ, και συνεπώς να έχουμε σύγκλιση στην αλληλουχία των ταυτόχρονων απαντήσεων και στην συνεχή και στη διακριτή περίπτωση (Amir 1996, Dubey et al 2006).Θα μελετήσουμε την περίπτωση $n = 4$ παικτών με συναρτήσεις κόστους (οι συναρτήσεις ζήτησης δίνονται από το πολυωνυμικό και το εκθετικό υπόδειγμα του προηγούμενου κεφαλαίου) $c_k = kxq_k + y$ (με k τον αριθμό του παίκτη) και παραμέτρους $A = -1$, $B = 7{,}36 \cdot 10^7$, $x = 10$, $y = 10$, στο πολυωνυμικό και $B = 8300$, $x = 100$, $y = 10$, στο υπόδειγμα με ρητό εκθέτη, που συνεπάγονται σύγκλιση της ακολουθίας των διαδοχικών βέλτιστων απαντήσεων (Dubey et al. 2006). Το σημείο ισορροπίας κατά Nash στην περίπτωση του πολυωνυμικού υποδείγματος είναι (Matlab Optimization Toolbox)

q1	86,9400905
q2	86,94006293
q3	86,94003537
q4	86,94000781

Ενώ στην περίπτωση του υποδείγματος με ρητό εκθέτη

q1	86,75703497
q2	83,05809161
q3	79,35914826
q4	75,66020491

Θα χρησιμοποιήσουμε επίσης και το πρώτο υπόδειγμα της Arifovic (1994), στην περίπτωση δυοπωλίου

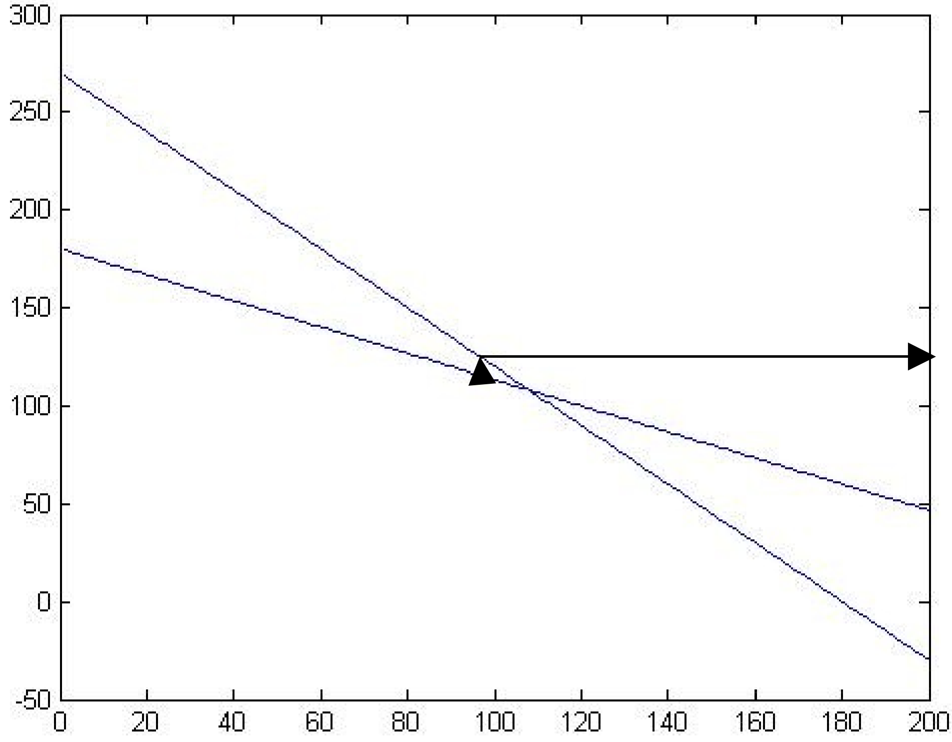
$$P = A - B(q_1 + q_2)$$
$$c = xq + yq^2$$

σαν benchmark της περίπτωσης στην οποία δεν έχουμε σύγκλιση στην αλληλουχία διαδοχικών βέλτιστων απαντήσεων. Για να βρούμε τα σύνολα παραμέτρων για τα οποία η αλληλουχία δεν συγκλίνει, να έχουμε δηλαδή αφενός ισορροπία κατά Nash σε καθαρές στρατηγικές και

$$\frac{\partial^2 u_1}{\partial q_1 \partial q_2}\frac{\partial^2 u_2}{\partial q_1 \partial q_2} > \frac{\partial^2 u_1}{\partial^2 q_1^2}\frac{\partial^2 u_2}{\partial^2 q_2^2}$$

(είναι $u_i = \pi_i$), δουλεύουμε ως εξής:

$\pi_i = \left[A - B\left(q_i + q_j\right)\right]q_i - xq_i - yq_i^2$ και άρα $\frac{\partial u_i}{\partial q_i} = A - B\left(q_i + q_j\right) - Bq_i - x - 2yq_i$, οπότε $\frac{\partial u_i}{\partial q_i} = 0 \Leftrightarrow q_i = \frac{A - x - Bq_j}{2B + 2y}$. Από την εξίσωση αυτή παίρνουμε τόσο τις συναρτήσεις αντίδρασης, $r_i\left(q_j\right) = \frac{A - x - Bq_j}{2B + 2y}$, όσο και την ισορροπία κατά Nash (αφού στην ισορροπία $q_i = q_j$) $q_n = \frac{A - x}{3B + 2y}$. Δεδομένου ότι θέλουμε μέγιστο θα πρέπει $\frac{\partial^2 u_i}{\partial q_i^2} < 0 \Leftrightarrow B + y > 0$. Επίσης $\frac{\partial^2 u_1}{\partial q_1 \partial q_2}\frac{\partial^2 u_2}{\partial q_1 \partial q_2} > \frac{\partial^2 u_1}{\partial^2 q_1^2}\frac{\partial^2 u_2}{\partial^2 q_2^2} \Leftrightarrow B^2 > 4\left(B + y\right)^2$. Αφού $B > 0$ και $B + y > 0$, $B^2 > 4\left(B + y\right)^2 \Leftrightarrow y < \frac{-B}{2}$ και τελικά, $-B < y < \frac{-B}{2}$. Για αυτές τις τιμές των B, y θα έχουμε ΝΕ σε καθαρές στρατηγικές $\left(\frac{\partial^2 u_i}{\partial q_i^2} < 0\right)$ και η ακολουθία των διαδοχικών βέλτιστων απαντήσεων θα αποκλίνει. Για A=1000, B=5, x=100 και $y = \frac{-10}{3}$, $q_n = 108$ και όπως φαίνεται κι απ΄ το διάγραμμα των συναρτήσεων αντίδρασης,



η ακολουθία βέλτιστων απαντήσεων αποκλίνει.

Τέλος θα χρησιμοποιήσουμε και ένα από τα μοντέλα των Son and Baldick (2004). Παρόλο που τα μοντέλα αυτά δεν είναι παίγνια Cournot με τη στενή έννοια (οι παίκτες έχουν διαφορετικές συναρτήσεις ζήτησης), η ύπαρξη local NE traps (Son and Baldick 2004), σημείων δηλαδή που, ενώ δεν είναι ισορροπίες κατά Nash, αποτελούν κοινά τοπικά ελάχιστα των συναρτήσεων κέρδους και όπως προαναφέραμε είναι πιθανό το κόλλημα ενός αλγόριθμου αναζήτησης σε αυτά, και η συμπεριφορά του αλγοριθμού μας κάτω από αυτές τις συνθήκες, αποτελεί σημαντικό κριτήριο για την χρησιμότητα του αλγορίθμου. Οι συναρτήσεις κέρδους των δύο παικτών δίνονται από

$$u_A(x_A, x_B) = 21 + x_A \sin \pi x_A + x_A x_B \sin \pi x_B$$
$$u_B(x_A, x_B) = 21 + x_B \sin \pi x_B + x_A x_B \sin \pi x_A$$

Για $0 \le x \le 7.5$, το NE σε καθαρές στρατηγικές είναι το $(x_A^N, x_B^N) = (6.61, 6.61)$, ενώ υπάρχουν και local NE traps στα σημεία $(6.58, 4.66)$, $(0.25, 0.92)$ και $(6.54, 2.8)$. Οι αλγόριθμοι των Hobbs et al. (2000) και Weber et al. (1999) δεν καταφέρνουν να ξεχωρίσουν το NE από τις local NE traps (Son and Baldick 2004).

7.4. Ο αλγόριθμος

Θα χρησιμοποιήσουμε τον αλγόριθμο που περιγράψαμε παραπάνω, με διαδοχικές απαντήσεις, δεδομένου ότι υπάρχει η θεωρητική θεμελίωση της σύγκλισης της αλληλουχίας τους, στις 2 πρώτες περιπτώσεις (Dubey et al 2006, Amir 1996) . Θα μπορούσε κανείς να χρησιμοποιήσει και ταυτόχρονες βέλτιστες απαντήσεις, ή και βέλτιστες απαντήσεις με διαφορετικές αναμενόμενες ποσότητες για τους συμπαίκτες (βλ. Fundenberg and Tirole 1991). Στην περίπτωση ταυτόχρονων απντήσεων, συμμετρικού παιγνίου και ίσων αρχικών ποσοτήτων για όλους τους παίκτες, ο αλγόριθμος μπορεί να απλοποιηθεί ακόμα περισσότερο (και να γίνει πιο γρήγορος), αφού δεν χρειάζεται να «τρέξει» ο γενετικός αλγόριθμος για κάθε παίκτη ξεχωριστά -η βέλτιστη απάντηση είναι η ίδια για όλους τους παίκτες. Βέβαια το σημαντικότερο στοιχείο παραμένει η σύγκλιση της θεωρητικής ακολουθίας των απαντήσεων στην ισορροπία κατά Nash. Αν η θεωρητική ακολουθία δεν συγκλίνει, ο αλγόριθμος αποκλίνει[12]. Ο αλγόριθμος έχει ως εξής (βλ. παράρτημα A7 για κώδικα Matlab):

1. Επιλογή αρχικών τιμών για τις ελάχιστες και μέγιστες τιμές που μπορούν να πάρουν τα χρωμοσώματα του κάθε παίκτη
2. Επιλογή n τυχαίων αρχικών στρατηγικών για τους παίκτες και εκτέλεση του εξής εσωτερικού βρόχου για όλους τους παίκτες, διαδοχικά
    a. Εύρεση της βέλτιστης απάντησης στο προφίλ στρατηγικών των υπολοίπων, με τη χρήση κανονικού γενετικού αλγορίθμου με διατεταγμένη προσαρμοστικότητα,

[12] Απ΄ την στιγμή που ο αλγόριθμος εξελίσσεται με βάση εργοδική αλυσίδα Markov, θα βρεθεί σε όλες τις δυνατές καταστάσεις του. Ωστόσο αυτό θα οφείλεται σε στοχαστικούς λόγους και ο χρόνος πρώτης μετάβασης στην ισορροπία κατά Nash, αναμένεται να είναι ιδιαίτερα υψηλός, στην περίπτωση που η θεωρητική αλληλουχία δεν συγκλίνει στο NE.

επιλογή γονέων με πιθανότητα ανάλογη της προσαρμοστικότητας, μονό και τυχαία προσδιορισμένο σημείο διαχωρισμού χρωμοσωμάτων και σταθερή πιθανότητα μετάλλαξης. Η καλύτερη στρατηγική που βρίσκει ο γενετικός αλγόριθμος αντικαθιστά την προηγούμενη στρατηγική του παίκτη στο προφίλ τρεχουσών στρατηγικών.

b. Τερματισμός του αλγορίθμου εφόσον το προφίλ στρατηγικών δεν μεταβλήθηκε στις προηγούμενες nT επαναλήψεις (διαφορετικά πήγαινε στο βήμα (a)) ή όταν ο αριθμός των βημάτων της ακολουθίας διαδοχικών βέλτιστων απαντήσεων υπερβεί ένα προκαθορισμένο όριο.

3. Επιλογή νέων τιμών για τις $Q^i_{\min}$, $Q^i_{\max}$, σύμφωνα με τις τιμές των q_i στις τελευταίες K επαναλήψεις και μετάβαση στο βήμα 2, για έναν επίσης προκαθορισμένο αριθμό επαναλήψεων.

7.5. Αποτελέσματα

7.5.1. Γραμμικό υπόδειγμα

Για το γραμμικό υπόδειγμα και για $Q_{\max} = 120$, $pop = 100$, $p_c = 1$, $p_m = 0{,}01$, 1000 γενεές σε κάθε εκτέλεση του γενετικού αλγορίθμου και $T = 5$, είχαμε τις εξής επαναλήψεις για τις 4 επιχειρήσεις:

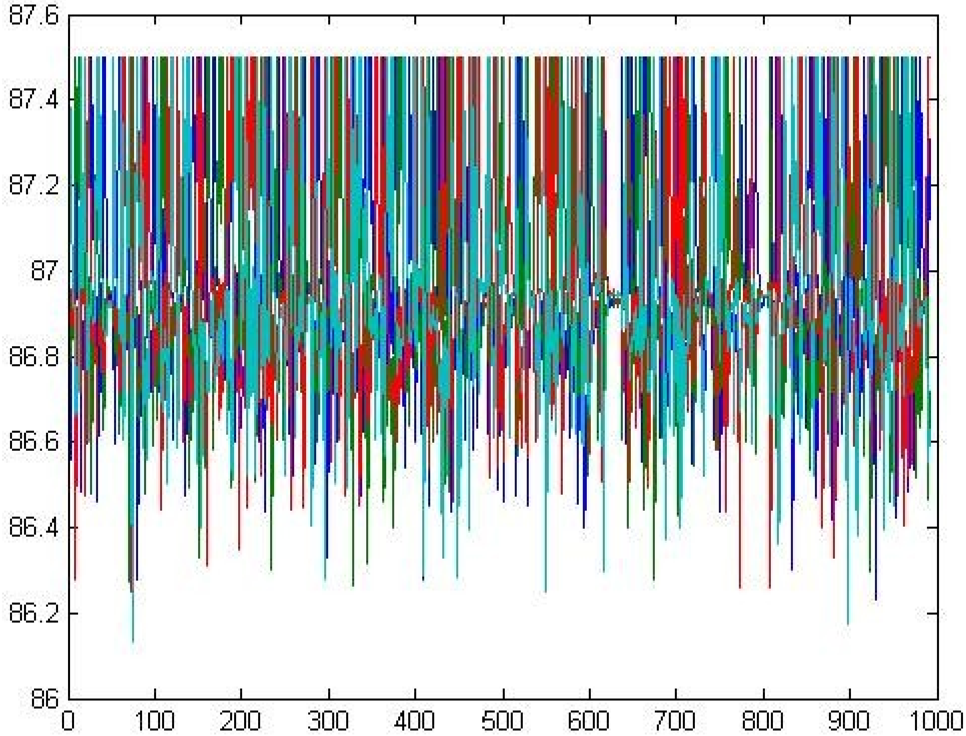
88,554	53,287	40,964	38,243	38,555	39,274	39,747	39,945	40,02	40,026
47,109	50,2	45,586	41,915	40,313	39,844	39,795	39,869	39,946	39,987
28,269	39,233	42,405	42,16	41,146	40,444	40,12	39,995	39,969	39,977
18,047	28,64	35,522	38,841	39,993	40,219	40,195	40,095	40,032	40,005

40,015	40,005	40,001	40	39,999	40	40	40	40	40
40,001	40,004	40,003	40,001	40	40	40	40	40	40
39,989	39,997	40	40	40	40	40	40	40	40
39,997	39,997	39,998	39,999	40	40	40	40	40	40

Ο αλγόριθμος τερμάτισε στην 20η επανάληψη στο NE $q_i^N = 20,\ i = 1,\ldots,4$. Αφήνοντας τον αλγόριθμο να συνεχιστεί μέχρι την 1000η επανάληψη, είχαμε αναμενόμενη συχνότητα για την κατάσταση $q_i^N = 20,\ i = 1,\ldots,4$, ίση με $\hat{\pi}_N = 0{,}984$ και αναμενόμενο χρόνο επαναφοράς $\hat{m}_{NN} = 1{,}01626$.

7.5.2. Πολυωνυμικό υπόδειγμα

Στο πολυωνυμικό υπόδειγμα, για το ίδιο σετ παραμέτρων, παρατηρούμε ότι ο αλγόριθμος δεν συγκλίνει. Οι ποσότητες, μετά από έναν αριθμό βημάτων, κυμαίνονται μεταξύ του 86 και του 87,5.

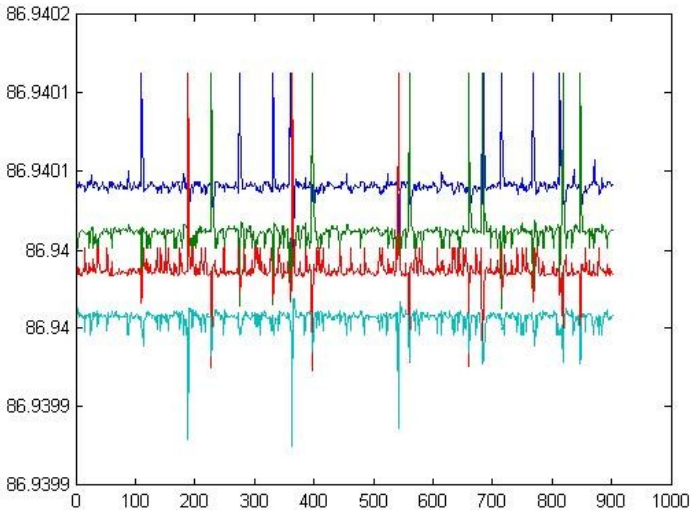


Αν χρησιμοποιήσουμε $Q_{\max} = 87{,}5$ και $Q_{\min} = 86{,}5$ για κάθε q_i^N έχουμε, για τις επαναλήψεις από 100 εώς 1000:

Nash intervals

	q1	q2	q3	q4
min	86,93750	86,93750	86,93750	86,93750
max	86,94310	86,94400	86,94460	86,94370

Θέτωντας για άλλη μια φορά $Q_{\min} = 86{,}9375$ και $Q_{\max} = 86{,}9446$ έχουμε για τις επαναλήψεις από 100 εώς 1000



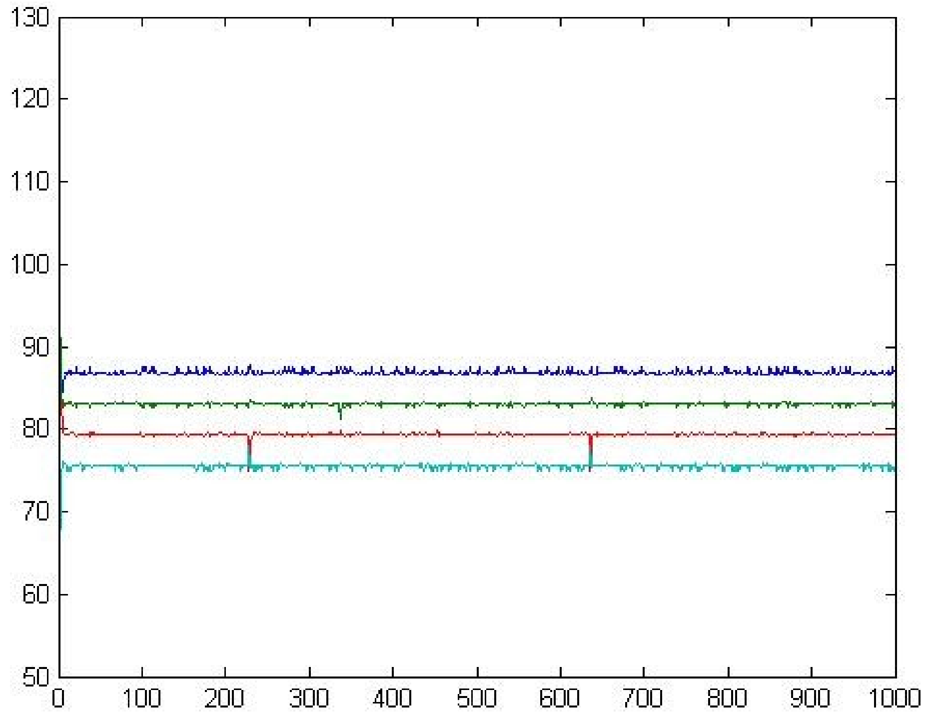
Nash Intervals

	q1	q2	q3	q4
min	86,940046425672900	86,940012996781300	86,939973528073800	86,939924654316600
max	86,940162502539200	86,940162502539200	86,940162502539200	86,940019043368400
mean	86,940091341955200	86,940061909895200	86,940037999838900	86,940006131756100

Μπορούμε να συνεχίσουμε αυτή τη διαδικασία όσες φορές θέλουμε (με διαφορετικά $Q_{\min}$ και $Q_{\max}$ για κάθε παίκτη, απ΄ τη στιγμή που η ακρίβεια γίνεται τόσο μεγάλη, ώστε οι μικρές διαφορές στα q_i^N να γίνουν σημαντικές).

7.5.3. Υπόδειγμα με ρητό εκθέτη

Για $pop = 100$, $p_c = 1$, $p_m = 0{,}01$, $T = 5$, ξεκινάμε με $Q_{\min} = 0$ και $Q_{\max} = 200$ για όλους τους παίκτες.



Για τις παρατηρήσεις από 100 εώς 1000 είχαμε

Nash Intervals

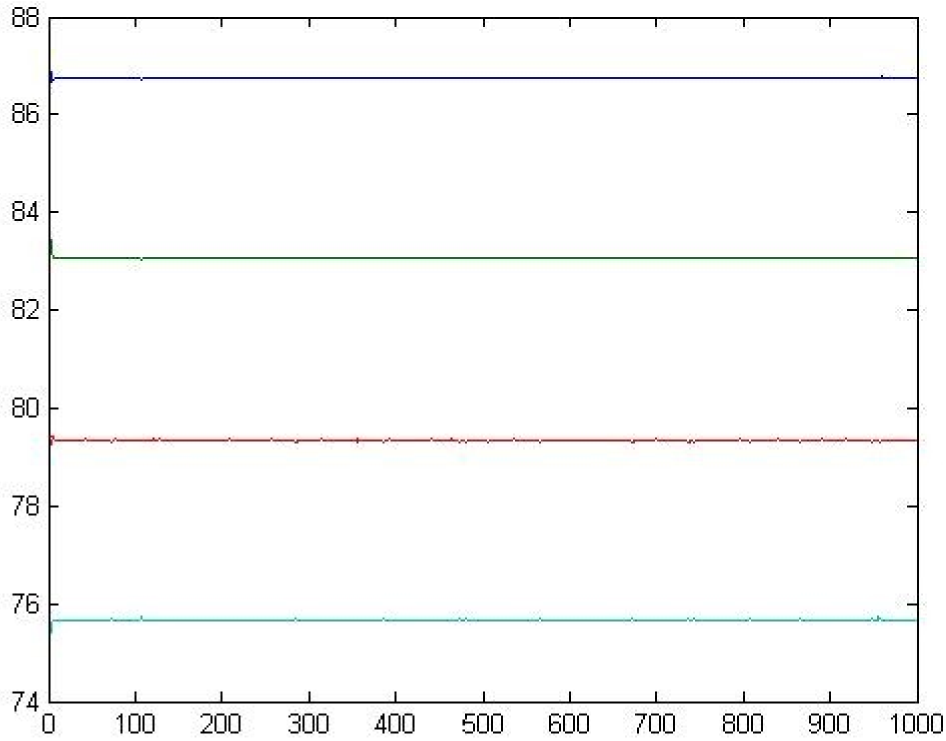
	q1	q2	q3	q4
min	86,620985623346000	81,249886751066900	74,999880790596800	74,999880790596800
max	87,880599861717100	83,759387740505000	79,931716853825400	77,953794435305100
mean	86,871269377449000	83,039159084961000	79,361882675560900	75,557777410666100

Θέτουμε

Value Intervals

	q1	q2	q3	q4
min	86,62	81,24	74,99	74,99
max	87,88	83,76	79,94	77,96

και επαναλαμβάνουμε, για μεγαλύτερη ακρίβεια.



Για τις επαναλήψεις 100...1000 είχαμε

Nash Intervals

	q1	q2	q3	q4
min	86,718436392246600	83,037370488520100	79,321249409913400	75,639685287175500
max	86,777500150203800	83,066017366425900	79,363343823760800	75,732500708103900
mean	86,757826134771000	83,057558000397000	79,357501005526800	75,661182399135300

Θέτουμε τα παραπάνω min και max σαν τις $Q^i_{\min}, Q^i_{\max}$ και συνεχίζουμε, για μεγαλύτερη ακρίβεια.

Για τις επαναλήψεις 100...1000 έχουμε

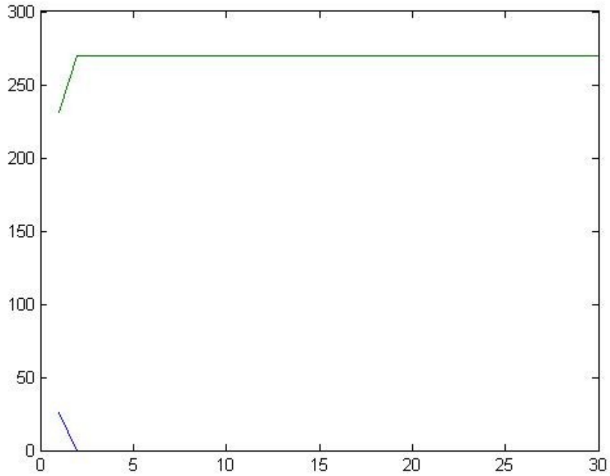
Nash Intervals

	q1	q2	q3	q4
min	86,755675667069100	83,057409410926800	79,358082017011800	75,659988591350600
max	86,757312379383400	83,058247337057300	79,359397508843500	75,662889164536500
mean	86,757064178033700	83,058084838326400	79,359154305916900	75,660173659449200

που μας δίνει το NE με ακρίβεια 4 σημαντικών ψηφίων.

7.5.4. Υπόδειγμα Arifovic

Στο υπόδειγμα της Arifovic, με τις παραμέτρους που εισάγαμε στην προηγούμενη ενότητα, η ακολουθία βέλτιστων απαντήσεων αποκλίνει. Για $pop = 100$, $p_c = 1$, $p_m = 0{,}01$ $T = 5$ και 100 γενιές στην προσομοίωση με $Q^{1,2}_{\min} = 0$, $Q^{1,2}_{\max} = 300$, τα αποτελέσματα ήταν:



Η απόκλιση της ακολουθίας απαντήσεων του αλγορίθμου, είναι προφανής.

7.5.5. Το υπόδειγμα με local NE traps

Για $pop = 100$, $p_c = 1$, $p_m = 0{,}01$ $T = 5$ και 1000 γενιές στην προσομοίωση με $Q_{\min}^{1,2} = 0$, $Q_{\max}^{1,2} = 7{,}5$, εκτελέσαμε 5 προσομοιώσεις με 30 επαναλήψεις η καθεμία. Τα αποτελέσματα ήταν:

Προσ.	**1**		**2**		**3**	
Παίκτης	**1**	**2**	**1**	**2**	**1**	**2**
1	6,613489736	6,611644375	6,564873757	6,615292182	6,552757313	6,621100064
2	6,611944782	6,611794578	6,611458408	6,611844646	6,610857592	6,611901867
3	6,562499106	6,615606895	6,611801731	6,611830341	6,611801731	6,611808883
4	6,611429798	6,611844646	6,611944782	6,611794578	6,611808883	6,611830341
5	6,611801731	6,611808883	6,621100064	6,610857592	6,621100064	6,611029254
6	6,611808883	6,611944782	6,611901867	6,611801731	6,611880409	6,611944782
7	6,611794578	6,611808883	6,621100064	6,610857592	6,611794578	6,611808883
8	6,611944782	6,611794578	6,611901867	6,611944782	6,611808883	6,611808883
9	6,611808883	6,611808883	6,611794578	6,611808883	6,611808883	6,611944782
10	6,611808883	6,611808883	6,611808883	6,611808883	6,621100064	6,610857592
11	6,611808883	6,611830341	6,611944782	6,611801731	6,611944782	6,611801731
12	6,611944782	6,611794578	6,611830341	6,611944782	6,611808883	6,621100064
13	6,611808883	6,611808883	6,611794578	6,611944782	6,610857592	6,611944782
14	6,611808883	6,611808883	6,611794578	6,611808883	6,611794578	6,621100064
15	6,611808883	6,611808883	6,562499106	6,615413776	6,61085044	6,611901867
16	6,611944782	6,611944782	6,611451255	6,611844646	6,611944782	6,611794578
17	6,611801731	6,621100064	6,611801731	6,611808883	6,611808883	6,611808883
18	6,61085044	6,611901867	6,611808883	6,611808883	6,611808883	6,611944782
19	6,611801731	6,611808883	6,611808883	6,562499106	6,611794578	6,611808883
20	6,611808883	6,611944782	6,615413776	6,611451255	6,611944782	6,611794578
21	6,611794578	6,611808883	6,611944782	6,611794578	6,562499106	6,615413776
22	6,611808883	6,611808883	6,611808883	6,611808883	6,611487018	6,611837494
23	6,611808883	6,611944782	6,611808883	6,611808883	6,611808883	6,611808883
24	6,611794578	6,611944782	6,611808883	6,611808883	6,611808883	6,611808883
25	6,611801731	6,611808883	6,611808883	6,611808883	6,621100064	6,61085044
26	6,611944782	6,611801731	6,611808883	6,611808883	6,611901867	6,611801731
27	6,611808883	6,611808883	6,611808883	6,611808883	6,611808883	6,611808883
28	6,611808883	6,611808883	6,611830341	6,611808883	6,611808883	6,611944782
29	6,611808883	6,621100064	6,611808883	6,611944782	6,611794578	6,611808883
30	6,610106573	6,61193763	6,611794578	6,621100064	6,611944782	6,611794578

Προσ.	4		5	
Παίκτης	1	2	1	2
1	0	6,514891639	6,581954081	6,613768686
2	6,617437952	6,611243831	6,611615764	6,611830341
3	6,611866104	6,611801731	6,611808883	6,611808883
4	6,611808883	6,621100064	6,611808883	6,611808883
5	6,610857592	6,611901867	6,611944782	6,611794578
6	6,611801731	6,611808883	6,611944782	6,621100064
7	6,611944782	6,611794578	6,610857592	6,611901867
8	6,611944782	6,611801731	6,562499106	6,621100064
9	6,611944782	6,611944782	6,61085044	6,611901867
10	6,611794578	6,611808883	6,611801731	6,611808883
11	6,611808883	6,562499106	6,611808883	6,611808883
12	6,615606895	6,611429798	6,611830341	6,611808883
13	6,611944782	6,611794578	6,611808883	6,611808883
14	6,611808883	6,611808883	6,611830341	6,611944782
15	6,611808883	6,611830341	6,611794578	6,611944782
16	6,562499106	6,615606895	6,611794578	6,611808883
17	6,611429798	6,611944782	6,611944782	6,611944782
18	6,611794578	6,611808883	6,611794578	6,611808883
19	6,611830341	6,611808883	6,611808883	6,611808883
20	6,611808883	6,611830341	6,611808883	6,611830341
21	6,562499106	6,61537086	6,611808883	6,611944782
22	6,611451255	6,611844646	6,611794578	6,611808883
23	6,611801731	6,611808883	6,611944782	6,611794578
24	6,611808883	6,621100064	6,611808883	6,611808883
25	6,610857592	6,611901867	6,611808883	6,611808883
26	6,611801731	6,611808883	6,611808883	6,611808883
27	6,611808883	6,611944782	6,611944782	6,611794578
28	6,611794578	6,562499106	6,611944782	6,611794578
29	6,615606895	6,611429798	6,611944782	6,611794578
30	6,611944782	6,611794578	6,611944782	6,611794578

Όπως διαπιστώνουμε, ο γενετικός αλγόριθμος δεν έχει το πρόβλημα των αλγορίθμων τοπικής αναζήτησης που, όπως αναφέρουν οι Son and Baldick (2004), «παγιδεύονται» στις local NE traps, και συνεπώς, δίνουν διαφορετικά αποτελέσματα σε κάθε εκτέλεση.

Παρατηρούμε επίσης ότι τα NE και οι local NE traps που αναφέρουν οι ερευνητές (βλ. Σχετικό πίνακα σε προηγούμενη ενότητα), δίνονται με ακρίβεια εκατοστού -στο ίδιο συμπέρασμα καταλήγουμε μετά τη χρήση της συνάρτησης fsolve, του optimization toolbox του Matlab. Συνεπώς, για να υπολογίσουμε με τον αλγόριθμό μας το σημείο ισορροπίας κατά Nash, με όποια επιθυμητή ακρίβεια, θα πρέπει να ακολουθήσουμε τη διαδικασία που ακολουθήσαμε παραπάνω (περισσότερες από μια εξωτερικές επαναλήψεις κλπ.).

7.6. Συμπεράσματα

Όπως είδαμε, ο αλγόριθμος που εισάγαμε στο κεφάλαιο αυτό, αποδίδει ιδιαίτερα ικανοποιητικά, εφόσον η θεωρητική ακολουθία των διαδοχικών βέλτιστων απαντήσεων συγκλίνει στην ισορροπία κατά Nash. Αυτό ισχύει στα πρώτα τρία υποδείγματα από αυτά που μελετήσαμε εδώ, αφού όλα ικανοποιούν τις συνθήκες των Dubey et al. (2006), που συνεπάγονται την σύγκλιση των διαδοχικών βέλτιστων απαντήσεων στην ισορροπία κατά Nash, είτε θεωρήσουμε την συνεχή (οι στρατηγικές επιλογές της ποσότητας μπορούν να πάρουν οποιαδήποτε πραγματική τιμή $q \in [0, q_{\max}]$), είτε τη διακριτή περίπτωση (οι ποσότητες που μπορούν να επιλέξουν οι παίκτες αποτελούν ένα πεπερασμένο υποσύνολο του $[0, q_{\max}]$). Η διακριτή περίπτωση ταιριάζει με ένα «φυσικό» τρόπο στους γενετικούς αλγόριθμους, αφού το σύνολο τιμών ενός χρωμοσώματος είναι πεπερασμένο. Στην συνεχή περίπτωση τίθεται ακόμα και θέμα εφικτότητας της εύρεσης της ακριβούς λύσης. Εφόσον κάποια από τις ποσότητες της ισορροπίας κατά Nash είναι, για παράδειγμα, άρρητος αριθμός, είναι προφανές ότι ένας υπολογιστικός αλγόριθμος δεν μπορεί να βρει με άπειρη ακρίβεια μια λύση. Κρίνεται λοιπόν αναγκαία η εισαγωγή του εξωτερικού βρόχου, ο οποίος επαναπροσδιορίζει το σύνολο τιμών των χρωμοσωμάτων κατά τέτοιο τρόπο, ώστε σε κάθε επανάληψή του να βελτιώνεται η ακρίβεια της λύσης.

Εφόσον η συνθήκη τερματισμού ικανοποιείται τότε (με πιθανότητα που τείνει στο 1, καθώς αυξάνεται ο αριθμός των εσωτερικών επαναλήψεων, όπως προαναφέραμε) η λύση στην οποία καταλήγει ο αλγόριθμος είναι η ισορροπία κατά Nash. Αν από την άλλη, ο αλγόριθμος δεν συγκλίνει στο δεδομένο αριθμό εσωτερικών επαναλήψεων, ο εξωτερικός βρόχος επαναπροσδιορίζει το σύνολο τιμών των χρωμοσωμάτων με τέτοιο τρόπο ώστε να αυξάνεται η δυνατή ακρίβεια των τιμών και -εφόσον η ακολουθία διαδοχικών απαντήσεων συγκλίνει- εγγυάται την αύξηση της ακρίβειας (όπως εκφράζεται από τον αριθμό των σημαντικών δεκαδικών ψηφίων), στον υπολογισμό των ποσοτήτων της ισορροπίας. Ο προσδιορισμός διαφορετικών συνόλων τιμών για τα χρωμοσώματα του κάθε παίκτη είναι προφανώς αναγκαίος στην περίπτωση που έχουμε μη συμμετρικό παίγνιο.

Τα παραπάνω ισχύουν, όπως είδαμε, ακόμα και στην περίπτωση που το παίγνιο έχει "local NE traps" (Son and Baldick 2004). Στο παίγνιο που αναλύσαμε (και το οποίο δεν είναι παίγνιο Cournot με την στενή έννοια) έχουν ήδη δοκιμαστεί αλγόριθμοι διαδοχικών βέλτιστων απαντήσεων που βασίζονται σε τοπική αναζήτηση, οι οποίοι δεν κατάφεραν να ανακαλύψουν την ισορροπία κατά Nash, λόγω της ύπαρξης των τοπικών αυτών ελαχίστων. Οι Son and Baldick (2004) έδειξαν επιπρόσθετα ότι ούτε το co-evolutionary programming που χρησιμοποιήσαμε στα προηγούμενα κεφάλαια ήταν αποτελεσματικό. Ο λόγος που συνέβη αυτό είναι εύκολα κατανοητός, με βάση την ανάλυση του κεφαλαίου 4. Όπως είδαμε οι «ποσότητες ισορροπίας» (οι αναμενόμενες ποσότητες, γύρω από τις οποίες συγκλίνουν οι επιλογές του κάθε παίκτη) είναι διαφορετικές από την ποσότητα που αντιστοιχεί στην στρατηγική Nash του κάθε παίκτη, ακόμα και αν οι μακροοικονομικές μεταβλητές του μοντέλου συγκλίνουν στην ισορροπία. Αυτό υποδηλώνει ότι είναι ιδιαίτερα απίθανη η εύρεση της ισορροπίας κατά Nash με τον αλγόριθμο αυτόν -παρά μόνο ίσως σε σχετικά απλά παίγνια, αφού απαιτείται ο προσδιορισμός *όλως* των στρατηγικών της λύσης και όχι κάποιος μέσος όρος. Το ίδιο βέβαια ισχύει και για τις τροποποιήσεις που κάναμε στον αλγόριθμο του Vriend και στο co-evolutionary programming στο κεφάλαιο 5, αφού απαιτείται συμμετρία στις συναρτήσεις κέρδους για να είναι εφικτή η εφαρμογή τους. Αντίθετα, ο αλγόριθμος που εισάγαμε εδώ μπορεί να προσδιορίσει το σημείο ισορροπίας κατά Nash, ακόμα και σε αυτήν την περίπτωση.

Η μοναδική περίπτωση στην οποία η εύρεση του σημείου ισορροπίας δεν ήταν εφικτή, ήταν στο υπόδειγμα της Arifovic (1994) παραπάνω, με τις παραμέτρους προσδιορισμένες έτσι ώστε η ακολουθία διαδοχικών βέλτιστων απαντήσεων να αποκλίνει. Σε αυτήν την περίπτωση ο αλγόριθμος αποκλίνει, με τις απαντήσεις των δύο παικτών να σταθεροποιούνται στις τιμές 0 (που είναι η ελάχιστη επιτρεπόμενη τιμή) και στην βέλτιστη απάντηση στην τιμή 0 (που είναι περίπου 270). Στην περίπτωση μάλιστα που επιτρέπαμε στις ποσότητες να πάρουν αρνητικές τιμές, οι ποσότητες αναμένεται ότι θα απόκλιναν εντελώς (στο $-\infty$ και $+\infty$ αντίστοιχα). Αυτό βέβαια είναι μια αναμενόμενη αδυναμία του αλγορίθμου, αφού είναι μια «αδυναμία» της θεωρητικής ακολουθίας στην οποία βασίζεται.

Το σημαντικό λοιπόν συμπέρασμα είναι ότι, τόσο σε παίγνια στα οποία δεν υπάρχουν τοπικά ελάχιστα στον συνδιασμό των συναρτήσεων κέρδους, όσο και στα παίγνια με τέτοιες local NE traps, ο αλγόριθμος συγκλίνει στην ισορροπία κατά Nash, με όσο ικανοποιητική ακρίβεια επιθυμούμε και με ικανή προϋπόθεση τη σύγκλιση της θεωρητικής ακολουθίας βέλτιστων απαντήσεων.

8. Συμπεράσματα

Στην μελέτη αυτή αναλύσαμε την εφαρμογή γενετικών αλγορίθμων στο παίγνιο που εισήχθη το 1838 από τον Γάλλο μηχανικό και οικονομολόγο A.A.Cournot (Cournot 1838). Το παίγνιο αυτό περιγράφει μια ολιγοπωλιακή αγορά με ορισμένο αριθμό επιχειρήσεων, οι οποίες προσδιορίζουν την ποσότητα (και μόνο) που θα παράγουν, με σκοπό βέβαια την μεγιστοποίηση του ιδίου τους οφέλους. Η τιμή στην αγορά προσδιορίζεται από την τομή των καμπυλών ζήτησης (εξωγενής) και προσφοράς, η οποία αποτελείται από το σύνολο των ποσοτήτων που προσφέρουν οι επιχειρήσεις. Χρησιμοποιήσαμε διάφορα τέτοια υποδείγματα, N παικτών, τα οποία είχαν πάντα ισορροπία σε καθαρές στρατηγικές -και ως επι των πλείστων οι θεωρητικές ακολουθίες διαδοχικών βέλτιστων απαντήσεων συνέκλιναν στην ισορροπία κατά Nash- για να διερευνήσουμε την σύγκλιση των μεμονωμένων ποσοτήτων που επιλέγουν οι παίκτες, σύμφωνα με κάποιους γενετικούς αλγόριθμους μάθησης (άλλων και δικούς μας), καθώς και την σύγκλιση του γενετικού αλγόριθμου διαδοχικών βέλτιστων απαντήσεων που εισάγαμε στο κεφάλαιο 7. Για να μπορέσουμε να εκτιμήσουμε την απόκλιση των καταστάσεων των αλγορίθμων αυτών, όπως ορίζονται από το σύνολο των πληθυσμών των χρωμοσωμάτων που προσδιορίζουν το σύνολο των εναλλακτικών του κάθε παίκτη σε κάθε επανάληψη του αλγορίθμου, από την κατάσταση - στόχο, στην οποία όλα τα χρωμοσώματα όλων των πληθυσμών όλων των παικτών συμπίπτουν με το χρωμόσωμα που αντιστοιχεί στην ποσότητα της ισορροπίας κατά Nash, εισάγαμε ένα «μέτρο» της μέσης απόστασης των καταστάσεων το οποίο βασίζεται στην έννοια της απόστασης Hamming και στην έννοια των ομαδικών καταστάσεων των αλυσίδων Markov (Kemeny and Snell 1960). Όπως αναλύσαμε, όλοι οι αλγόριθμοι που εισάγαμε χαρακτηρίζονται από εργόδικες αλυσίδες Markov, χωρίς όμως αυτό να ισχύει κατ΄ ανάγκη για την σύνθετη αλυσίδα των ομαδικών καταστάσεων. Αυτό δεν μας εμποδίζει όμως απ΄ το να εκτιμήσουμε, τόσο την αναμενόμενη συχνότητα της κατάστασης - στόχου («κατάστασης Nash»), τον αναμενόμενο χρόνο επαναφοράς στην κατάσταση αυτή, καθώς και τον μέσο χρόνο που απαιτείται για να φτάσουμε στην κατάσταση Nash, με αρχική κατάσταση την ακριβώς αντίθετή της (την κατάσταση στην οποία όλα τα χρωμοσώματα όλων των πληθυσμών είναι αντεστραμμένα). Επίσης ήταν εφικτή (Kemeny and Snell 1960) η εκτίμηση των πιθανοτήτων μετάβασης ενός βήματος της σύνθετης αλυσίδας, χωρίς όμως τη γενίκευση στα n βήματα, λόγω της απώλειας της Μαρκοβιανής ιδιότητας, μετά την εισαγωγή των ομαδικών καταστάσεων.

Ο αλγόριθμος της Arifovic (1994), ο οποίος είχε ήδη αναφερθεί ότι συνέκλινε στην Βαλρασιανή ισορροπία και όχι στην ισορροπία κατά Nash, πράγματι οδήγησε την κατάσταση στην αγορά (τιμή και ποσότητα ισορροπίας, όπως προσδιορίζονται αμφιμονοσήμαντα από την μέση ποσότητα των παικτών) στις τιμές της ισορροπίας Walras, χωρίς να συμβαίνει το ίδιο με τις επιλογές των παικτών καθεαυτές, που συνέκλιναν σε διαφορετικές ποσότητες (μόνο η μέση ποσότητα συνέκλινε στην ποσότητα Walras). Πρόκειται για ένα πολύ σημαντικό μειονέκτημα, που δεν έχει καταγραφεί στη βιβλιογραφία μέχρι στιγμής, και το οποίο υποδηλώνει ότι οι πράκτορες που λειτουργούν με το πρότυπο του αλγορίθμου αυτού, δεν μαθαίνουν αποτελεσματικά (αν θεωρήσει κανείς ότι η Βαλρασιανή ισορροπία αποτελεί στόχο προς μάθηση).

Ανάλογη εικόνα είχαμε και στους αλγορίθμους του Vriend (2000) και co-evolutionary programming (Price 1997). Εδώ είχαμε βέβαια σύγκλιση της μέσης ποσότητας (και της τιμής και ποσότητας στην αγορά, κατά συνέπεια) στην ισορροπία κατά Nash, αλλά επίσης οι αναμενόμενες ποσότητες στις οποίες συγκλίνουν οι επιλογές των μεμονωμένων παικτών, διαφέρουν από την τιμή της ισορροπίας κατά Nash, σε όλες τις προσομοιώσεις που εκτελέσαμε (γραμμικό, πολυωνυμικό και υπόδειγμα με ρητό εκθέτη, για 20 και 4 επιχειρήσεις). Έχουμε δηλαδή μια κατάσταση ακριβώς ανάλογη με εκείνη του αλγορίθμου της Arifovic, με την μόνη διαφορά ότι η «ισορροπία αναφοράς» είναι η ισορροπία Nash-Cournot, αντί της Βαλρασιανής.

Λόγω του ότι ο αλγόριθμος του Vriend είναι παρόμοιος με το co-evolutionary programming, οι διαφορές στην αποτελεσματικότητά τους ήταν επίσης πολύ μικρές. Οι διαφορές στις αναμενόμενες ποσότητες (τόσο τις συνολικές όσο και τις επιμέρους) είναι ασήμαντες, ενώ όποιες διαφορές παρατηρήθηκαν στις τυπικές αποκλίσεις αυτών των αναμενόμενων ποσοτήτων, αποδίδονται στις διαφορετικές πιθανότητες μετάλαξης που χρησιμοποιήθηκαν κάθε φορά. Μικρότερες πιθανότητες μετάλλαξης οδηγούν σε μεγαλύτερη «ευστάθεια» του αλγορίθμου, γύρω από την τιμή στην οποία «ισορροπεί» (με την διευρυμένη έννοια της ευστάθειας του Ljapunov πάντα). Η σχέση της παραμέτρου GArate με το μέγεθος του πληθυσμού είναι βέβαια σημαντική στον αλγόριθμο του Vriend, σε αντίθεση με την περίπτωση του co-evolutionary programming, στο οποίο το ζήτημα αυτό ρυθμίζεται, κατά κάποιο τρόπο, αυτόματα, αφού η περίοδος ανανέωσης του πληθυσμού προκύπτει μονοσήμαντα από το πλήθος των χρωμοσωμάτων. Όπως είδαμε (βλ. και Valee και Yildizoglou 2007) το GArate πρέπει να είναι κοντά (αν όχι μεγαλύτερο από) στο 50, με τους πληθυσμούς να είναι επίσης περίπου ίσοι ή και μεγαλύτεροι από τον αριθμό αυτόν. Από αυτήν την σκοπιά, μπορούμε να πούμε ότι το co-evolutionary programming είναι πιο «γρήγορο» στην εκτέλεσή του, αφού λειτουργεί και για μικρότερους πληθυσμούς (ή ισοδύναμα, με μικρότερο GArate, αφού αυτά τα δύο είναι ίσα στον αλγόριθμο αυτό), με δεδομένο πάντα ότι θα κωδικοποιήσουμε κατάλληλα μια διαδικασία που να εγγυάται ότι κάθε χρωμόσωμα θα συμμετέχει σε ένα και μόνο ένα παίγνιο, πριν την ανανέωση του πληθυσμού (βλ. κώδικα Matlab στο παράρτημα Α-3).

Ο αλγόριθμος των Alkemade et al (2007) έδωσε ικανοποιητικά αποτελέσματα, αλλά εδώ μιλάμε για έναν αλγόριθμο κοινωνικής μάθησης και κοινού πληθυσμού για τους παίκτες.

Το ζήτημα που τελικά ανακύπτει είναι το να μετατρέψουμε τους αλγόριθμους του Vriend και του co-evolutionary programming που είναι αλγόριθμοι ατομικής μάθησης, πολλών πληθυσμών (ένα για κάθε παίκτη), σε αλγόριθμους κοινωνικής μάθησης. Να επιτρέψουμε δηλαδή στην διαδικασία της μίμησης -όπως αυτή εκφράζεται μέσω του τελεστή διασταύρωσης σε έναν γενετικό αλγόριθμο μάθησης- να οδηγήσει στην εξίσωση μέσω της αλληλεπίδρασης, των δυνητικών επιλογών των παικτών, όπως αυτές προσδιορίζονται από τα χρωμοσώματα των αντίστοιχων πληθυσμών τους. Σημειώνουμε ότι για να είναι αυτό εφικτό, να έχει δηλαδή νόημα για έναν πράκτορα η μίμηση των επιλογών των συμπαικτών του, θα πρέπει να έχουμε συμμετρία στο παίγνιο, όπως αναλύσαμε παραπάνω. Διαφορετικά μια επιλογή η οποία μπορεί να είναι πολύ καλή για έναν παίκτη, θα μπορούσε να αποβεί ιδιαίτερα ζημιογόνα για κάποιον άλλο.
Πράγματι οι τροποποιήσεις που κάναμε στο κεφάλαιο 6, έδειξαν ότι οι «κοινωνικοποιημένες» εκδόσεις του αλγόριθμου του Vriend και του co-evolutionary programming που εισάγαμε οδηγούν, εκτός της σύγκλισης των ποσοτήτων των μεμονωμένων παικτών στην ποσότητα της ισορροπίας κατά Nash, και στην σύγκλιση γενικά του αλγορίθμου κοντά έστω στην κατάσταση με όλα τα χρωμοσώματα ίσα με το χρωμόσωμα που αντιστοιχεί στην ποσότητα αυτή. Οδηγούν συνεπώς στην εύρεση της ισορροπίας κατά Nash στα παίγνια Cournot που αναλύσαμε. Σε όλες τις προσομοιώσεις που εκτελέσαμε και σε όλα τα υποδείγματα που εξετάσαμε, είδαμε ότι οι μέσες ποσότητες κατά τη διάρκεια της προσομοίωσης όλων των πρακτόρων συνέκλιναν πολύ κοντά στην ποσότητα της ισορροπίας Nash. Ο λόγος που συνέβη αυτό είναι φυσικά η κοινωνική αντί της ατομικής μάθησης. Από την στιγμή που η επόμενη γενιά των χρωμοσωμάτων των πληθυσμών, καθορίζεται από την εφαρμογή των τελεστών του γενετικού αλγορίθμου στον συνολικό πληθυσμό που προκύπτει από την ένωση των επιμέρους πληθυσμών, ο γενετικός αλγόριθμος οδηγεί στην ομογενοποίηση του ενιαίου αυτού πληθυσμού και συνεπώς και των μεμονωμένων πληθυσμών. Αυτό βεβαίως σημαίνει ότι, εφόσον ο αριθμός των γενεών είναι επαρκής, η αλυσίδα Markov που χαρακτηρίζει την εξέλιξη του συστήματος, θα βρεθεί στην «κατάσταση Nash» (όλα τα χρωμοσώματα όλων των πληθυσμών ίσα με την ποσότητα της ισορροπίας κατά Nash) και συνεπώς, ο αλγόριθμος θα «ανακαλύψει» το σημείο ισορροπίας. Δυστυχώς η συν-εξελικτική φύση των αλγορίθμων έχει ως συνέπεια ότι ο αναμενόμενος αριθμός γενεών που απαιτείται για την πρώτη μετάβαση στην κατάσταση Nash είναι ιδιαίτερα αυξημένος. Έτσι τίθεται ένα θέμα υπολογιστικής ισχύς για μεγάλο αριθμό επιχειρήσεων και δυαδικών ψηφίων στα χρωμοσώματα. Για να είναι εφικτή η επίσκεψη στην κατάσταση Nash σε αυτήν την περίπτωση χρειάζεται μεγάλος αριθμός επαναλήψεων, με την κάθε επανάληψη να απαιτεί μεγάλο χρονικό διάστημα υπολογισμού, λόγω του μεγάλου αριθμού χρωμοσωμάτων. Από

την άλλη είναι και πιθανοκρατικά δυσκολότερο να συμπέσουν όλα τα χρωμοσώματα σε όλους τους πληθυσμούς, με αποτέλεσμα να χρειάζεται ακόμα μεγαλύτερος αριθμός επαναλήψεων, σε σχέση με τις περιπτώσεις με μικρότερο αριθμό πρακτόρων και δυαδικών ψηφίων.Η υπολογιστική ισχύς που είχαμε στη διάθεσή μας, μας επέτρεψε την εκτέλεση αρκετών επαναλήψεων ώστε να διαπιστώσουμε την πραγμάτωση της κατάστασης Nash, στις περιπτώσεις των 4 επιχειρήσεων με 20 δυαδικά ψηφία σε κάθε χρωμόσωμα και των 20 επιχειρήσεων με 8-μπιτα χρωμοσώματα. Ένα θέμα λοιπόν για μελλοντική έρευνα θα μπορούσε να είναι η έμπρακτη απόδειξη της σύγκλισης των αλγορίθμων σε περιπτώσεις με μεγαλύτερο αριθμό επιχειρήσεων και μεγαλύτερη ακρίβεια στην κωδικοποίηση των ποσοτήτων, μέσω της αξιοποίησης μεγαλύτερης υπολογιστικής ισχύος.

Οι συνθήκες ευστάθειας των αλγορίθμων φαίνονται από τις αναμενόμενες συχνότητες των ομαδικών (lumped) καταστάσεων και τους αναμενόμενους χρόνους επιστροφής στην κατάσταση Nash. Οι χρόνοι επιστροφής που παρατηρήσαμε στις προσομοιώσεις είναι ιδιαίτερα ικανοποιητικοί (λιγότερο από 10 γενιές). Βέβαια οι χρόνοι αυτοί συνδέονται άμεσα με τις παραμέτρους και κυρίως από την πιθανότητα μετάλλαξης, με τον αριθμό των δυαδικών ψηφίων στα χρωμοσώματα και τον αριθμό των επιχειρήσεων στο μοντέλο, να έχουν επίσης τον πλέον καταλυτικό ρόλο. Οι εκτιμήσεις για τις αναμενόμενες συχνότητες των ομαδικών καταστάσεων έδειξαν -στις περιπτώσεις που αναλύσαμε παραπάνω- ότι η πλέον συχνή μεμονωμένη κατάσταση είναι η κατάσταση Nash, ενώ η πλέον συχνή ομαδική κατάσταση ήταν η s_1, η ομαδική κατάσταση δηλαδή που αποτελείται από το σύνολο των μεμονωμένων καταστάσεων στις οποίες η μέση απόσταση των χρωμοσωμάτων όλων των πληθυσμών είναι μικρότερη ή ίση με 1. Πιο συγκεκριμένα, αν λάβουμε υπόψη ότι η αθροιστική συχνότητα των δύο αυτών ομαδικών καταστάσεων s_0 και s_1 είναι το άθροισμα των επιμέρους συχνοτήτων, συμπεραίνουμε ότι τα χρωμοσώματα των πληθυσμών των παικτών θα απέχουν -κατά μέσο όρο- από το χρωμόσωμα που αντιστοιχεί στην ισορροπία κατά Nash κατά ένα ή και λιγότερο δυαδικό ψηφίο, με πιθανότητα ίση με το άθροισμα των αναμενόμενων συχνοτήτων των δύο καταστάσεων. Οι προσομοιώσεις μας έδειξαν ότι η αναμενόμενη συχνότητα της σύνθετης αυτής κατάστασης είναι ιδιαίτερα υψηλή (πάνω από 0,9), γεγονός που καταδεικνύει την ευστάθεια του αλγορίθμου. Για παράδειγμα η αθροιστική συχνότητα των s_0 και s_1 στην προσομοίωση του αλγόριθμου του Vriend που παρουσιάσαμε για το γραμμικό υπόδειγμα 4 παικτών με 20-μπιτα χρωμοσώματα ήταν $n_0 + n_1 = 90.018/100$ γενεές. Ανάλογα ήταν τα αποτελέσματα για τα υπόλοιπα μοντέλα και το «κοινωνικοποιημένο» co-evolutionary programming. Οι αλγόριθμοι αυτοί είναι λοιπόν ιδιαίτερα ευσταθείς (με την έννοια τουλάχιστον της ευστάθειας Ljapunov [Riechmann 1997]).

Αξιολογώντας την αποτελεσματικότητα των αλγορίθμων που έχουν χρησιμοποίηθει στο πρόβλημα στο παρελθόν, έχουμε κάθε λόγο να μην μένουμε ικανοποιημένοι. Η σύγκλιση των βασικών παραμέτρων που ορίζουν την κατάσταση της αγοράς, στην ανταγωνιστική, αντί την ισορροπίας κατά Nash, υποδηλώνει την πλήρη αστοχία των υποδειγμάτων μάθησης. Ακόμα και οι αλγόριθμοι που αναλύσαμε στο κεφάλαιο 5, οι οποίοι οδηγούν στην σύγκλιση της τιμής και της ποσότητας στην αγορά στην ισορροπία Nash (Vriend, co-evolutionary programming), μας οδηγούν στο συμπέρασμα ότι η ατομική μάθηση δεν οδηγεί στην αποτελεσματική σύγκλιση όλων των παικτών στην ποσότητα ΝΕ. Οι παίκτες δηλαδή δεν μαθαίνουν αποτελεσματικά. Αντίθετα, οι εκδόσεις κοινωνικής μάθησης των παραπάνω αλγορίθμων που προτείναμε εδώ, λύνουν το πρόβλημα. Συνεπώς η ετερογένεια που αναφέραμε στο προηγούμενο κεφάλαιο, που παρατηρείται στην συμπεριφορά των μεμονωμένων παικτών στις αρχικές εκδόσεις ατομικής μάθησης, αίρεται στην «κοινωνικοποιημένη» έκδοση των αλγορίθμων. Εδώ, τόσο τα μακροοικονομικά όσο και τα ατομικά μεγέθη, συγκλίνουν -έστω και κατά προσέγγιση, στη γενική περίπτωση- στην ισορροπία κατά Nash.

Όσον αφορά την οικονομική ερμηνεία των υποδειγμάτων, η ομογενής αυτή σύγκλιση έχει συνέπειες και στο ζήτημα της ισοκατανομής του εισοδήματος. Στην περίπτωση της ατομικής μάθησης, η διαφοροποίηση στις ποσότητες ισορροπίας των παικτών, σημαίνει προφανώς και αποκλίσεις στα πραγματοποιηθέντα κέρδη τους. Θα υπάρχει λοιπόν μεγαλύτερη ανισοκατανομή του εισοδήματος στην περίπτωση αυτή -η μέτρηση της ανισοκατανομής του εισοδήματος μπορεί να γίνει με τη χρήση του συντελεστή Gini, ο οποίος θα είναι αυξημένος στην περίπτωση της ατομικής μάθησης, σε σχέση με την κοινωνική. Η μίμηση των επιτυχημένων συμπεριφορών των υπολοίπων πρακτόρων, από την άλλη, οδηγεί σε εξίσωση της αναμενόμενης τιμής της κατανομής του κέρδους για κάθε πράκτορα. Η εξίσωση βέβαια των πραγματοποιηθέντων κερδών δεν μπορεί να είναι πλήρης, αφενός λόγο του ότι θα πρέπει να έχουμε κάποια συγκεκριμένα σύνολα παραμέτρων στον γενετικό αλγόριθμο για να έχουμε σχετικά γρήγορη σύγκλιση στην ισορροπία κατά Nash και αφετέρου γιατί -ακόμα και στην περίπτωση που συγκλίνουμε στην ισορροπία- θα υπάρχουν πάντα τυχαίες διακυμάνσεις, λόγο της στοχαστικής διαδικασίας στην οποία οδηγούμαστε με την εφαρμογή του οποιουδήποτε γενετικού αλγορίθμου.

Η ανισοκατανομή αυτή που παρατηρήσαμε στους αρχικούς αλγορίθμους, σημειώνουμε, θα ήταν ακόμα μεγαλύτερη αν συμπεριλαμβάναμε στο υπόδειγμά μας την έννοια της περιουσίας των πρακτόρων, η οποία προφανώς θα συνδεόταν άμεσα με το εισόδημά τους σε κάθε συγκεκριμένο γύρο της προσομοίωσης. Αν μάλιστα δεχόμασταν ότι, η παραγωγική ικανότητα της κάθε επιχείρησης εξαρτάται άμεσα από την περιουσία της, όπως συμβαίνει στις πραγματικές καπιταλιστικές οικονομίες, όπου οι επενδύσεις σε κεφαλαιουχικό εξοπλισμό αυξάνουν σημαντικά την παραγωγική δυνατότητα μιας μονάδας, δεν είναι δύσκολο να φανταστούμε τις μακροπρόθεσμες επιδράσεις αυτής της ανισότητας στην κατανομή εισοδήματος. Η ανισοκατανομή θα είχε μια καθαρά αυξητική τάση, ίσως και με αυξανόμενο ρυθμό, και οι πολλαπλασιαστικές αυτές επιδράσεις θα δημιουργούσαν, μέσω του μηχανισμού της αγοράς πάντα, τεράστιες στρεβλώσεις στο εισόδημα των πρακτόρων. Αυτό είναι ένα σημαντικό ζήτημα προς διερεύνηση, σε μελλοντική έρευνα.

Η μελέτη μας οδηγεί επίσης στο συμπέρασμα ότι οι κοινωνικοί συν-εξελικτικοί γενετικοί αλγόριθμοι που εισάγαμε μπορούν να χρησιμοποιηθούν και σαν ευρετικοί αλγόριθμοι για την επίλυση παιγνίων Cournot σε καθαρές στρατηγικές, εφόσον οι συναρτήσεις κόστους είναι κοινές. Θα πρέπει βέβαια να εισαχθούν κάποιοι μηχανισμοί «ανίχνευσης» της λύσης της ισορροπίας κατά Nash σε καθαρές στρατηγικές (NE). Από την στιγμή που διαπιστώσαμε ότι οι καταστάσεις με μέση απόσταση Hamming από την κατάσταση με όλα τα χρωμοσώματα όλων των πληθυσμών μικρότερη από 1, έχουν ιδιαίτερα αυξημένη συχνότητα στην ισορροπία της αλυσίδας Markov (πολλές φορές η αθροιστική τους συχνότητα υπερβαίνει το 0.9) είναι εύλογο ότι θα υπάρχουν πολλά χρωμοσώματα που θα απεικονίζουν το NE στους πληθυσμούς, μετά από έναν επαρκή αριθμό γενεών. Ακόμα και στην -εξαιρετικά απίθανη- περίπτωση που κάτι τέτοιο δεν συμβαίνει, όλα σχεδόν τα χρωμοσώματα θα έχουν εξαιρετικά μικρή απόσταση Hamming από το χρωμόσωμα της ποσότητας της ισορροπίας κατά Nash. Συνεπώς η εξέταση των πληθυσμών των αλγορίθμων θα μας δώσει μια ξεκάθαρη εικόνα για το ποιό είναι αυτό το χρωμόσωμα. Η επαλήθευση αυτής της «υποψήφιας ποσότητας NE», μπορεί να γίνει με την επίλυση του προβλήματος μεγιστοποίησης ενός παίκτη (και άρα μεταβλητής) που προκύπτει αν θεωρήσουμε ότι όλοι οι παίκτες -πλην του εν λόγω παίκτη- χρησιμοποιούν την ποσότητα αυτή. Εφόσον η ποσότητα που θα προκύψει για τον παίκτη του οποίου την απόφαση προσπαθούμε να βελτιστοποιήσουμε, ισούται με την «υποψήφια ισορροπία» τότε, λόγω συμμετρίας του παιγνίου, η λύση αυτή είναι ποσότητα ισορροπίας κατά Nash σε καθαρές στρατηγικές.

Στην περίπτωση τώρα που ο αλγόριθμος βρίσκει την ισορροπία κατά Nash, στην περίπτωση δηλαδή που υπάρχει παίγνιο κατά τη διάρκεια της εκτέλεσης του αλγορίθμου, στο οποίο όλοι οι παίκτες παίζουν τη στρατηγική NE, μπορούμε να εξετάσουμε ένα πολύ μικρότερο υποσύνολο των στρατηγικών για να βρούμε την «υποψήφια ποσότητα NE». Από την στιγμή που, λόγω συμμετρίας και πάλι, όλοι οι παίκτες θα παίζουν την ίδια στρατηγική στην ισορροπία, θα μπορούσαμε εισάγουμε στον κώδικα εντολές που να καταγράφουν τα παίγνια με αυτό το χαρακτηριστικό -με όλους δηλαδή τους παίκτες να χρησιμοποιούν κοινή ποσότητα. Στη συνέχεια μπορούμε να επιλύσουμε το προαναφερθέν πρόβλημα βελτιστοποίησης μίας μεταβλητής -και το οποίο στις περιπτώσεις των υποδειγμάτων που εξετάσαμε είναι ένα εύκολα επιλύσιμο πρόβλημα κυρτού προγραμματισμού- για αυτές τις υποψήφιες στρατηγικές.

Τελικά, η πλέον αποτελεσματική διαδικασία «ανίχνευσης» του NE είναι η εξέταση πρώτα των ποσοτήτων που χρησιμοποιήθηκαν από κοινού από όλους τους παίκτες, όπως περιγράψαμε στην προηγούμενη παράγραφο και στη συνέχεια, εφόσον δεν βρέθηκε ισορροπία κατά Nash ανάμεσα σε αυτές, εξέταση των πληθυσμών των τελευταίων γενεών των αλγορίθμων για χρωμοσώματα με μεγάλη συχνότητα. Στην περίπτωση δεν έχουμε ούτε τότε αποτέλεσμα, οπότε ο αριθμός των γενεών που χρησιμοποιήθηκαν δεν επαρκεί για σύγκλιση κοντά στην «κατάσταση Nash», θα πρέπει να επαναλάβουμε με μεγαλύτερο αριθμό γενεών ή με μικρότερη πιθανότητα μετάλλαξης.

Τελικά, όπως είδαμε, η παραπάνω μεθοδολογία της κοινωνικής μάθησης, μπορεί να εφαρμοστεί στην περίπτωση που οι συναρτήσεις κέρδους είναι συμμετρικές. Στο κεφάλαιο 7, επανήλθαμε στην μεθοδολογία των κλασικών γενετικών αλγορίθμων βελτιστοποίησης (αντί των συν-εξελικτικών γενετικών αλγορίθμων μάθησης) για να αντιμετωπίσουμε την περίπτωση των μη συμμετρικών παιγνίων. Βασιζόμενοι στην σύγκλιση της θεωρητικής ακολουθίας των διαδοχικών βέλτιστων απαντήσεων που αναλύσαμε διεξοδικά στο κεφάλαιο 2, χρησιμοποιήσαμε έναν αλγόριθμο ο οποίος, αφού προσδιορίσει κάποιες αρκετά γενικές αρχικές τιμές για το εύρος των δυνατών ποσοτήτων που μπορούν να κωδικοποιήσουν τα χρωμοσώματα του πληθυσμού του κάθε παίκτη και μια τυχαία αρχική κατάσταση στην αγορά, προσδιορίζει διαδοχικά την βέλτιστη απάντηση του καθενός από τους παίκτες, με δεδομένες τις επιλογές των υπολοίπων, ανανεώνοντας την κατάσταση στην αγορά (τις ποσότητες δηλαδή που επιλέγουν οι υπόλοιποι παίκτες) κάθε φορά. Οι προσομοιώσεις που πραγματοποιήσαμε έδειξαν ότι ο αλγόριθμος αποδίδει ιδιαίτερα ικανοποιητικά, καλύτερα από κάθε άλλον επαναληπτικό αλγόριθμο -αφού αποφεύγει το πρόβλημα των "local NE traps"- εφόσον η θεωρητική ακολουθία διαδοχικών βέλτιστων απαντήσεων συγκλίνει σε ισορροπία κατά Nash. Τα 4 από τα 5 υποδείγματα που μελετήσαμε ικανοποιούσαν αυτήν την προϋπόθεση. Ο αλγόριθμος βρήκε την ισορροπία κατά Nash με την όποια επιθυμητή ακρίβεια, προσδιορίζοντας κατάλληλα τον αριθμό των επαναλήψεων του εξωτερικού βρόχου. Στα υποδείγματα αυτά, η ικανοποίηση των συνθηκών των Dubey et al. (2006), εγγυάται την σύγκλιση της ακολουθίας των διαδοχικών βέλτιστων απαντήσεων στο NE, τόσο στην συνεχή (οι ποσότητες είναι πραγματικοί αριθμοί, μεταξύ του $q_{\min}$ και του $q_{\max}$), όσο και στη διακριτή περίπτωση (οι ποσότητες που μπορούν να επιλέξουν οι παίκτες αποτελούν ένα πεπερασμένο υποσύνολο του $[q_{\min}, q_{\max}]$). Στην δεύτερη μάλιστα περίπτωση, η ισορροπία κατά Nash θα βρίσκεται εξ΄ ορισμού στο σύνολο των δυνατών τιμών των χρωμοσωμάτων, οπότε ο εξωτερικός βρόχος παύει να είναι απαραίτητος. Αντίθετα στην συνεχή περίπτωση και εφόσον η NE δεν βρίσκεται στις εφικτές τιμές των χρωμοσωμάτων (κάτι που θα ισχύει πάντα αν για παράδειγμα, η ποσότητα ισορροπίας είναι άρρητος αριθμός), ο εξωτερικός βρόχος οδηγεί σε μια συνεχή βελτίωση της ακρίβειας της λύσης. Αυτό γίνεται, όπως είπαμε, λαμβάνοντας σαν ελάχιστη και μέγιστη τιμή για το σύνολο τιμών των χρωμοσωμάτων ($q_{\min}$ και $q_{\max}$) τις ελάχιστες και μέγιστες τιμές στα τελευταία K (με K=100 στην εφαρμογή μας) βήματα του εσωτερικού βρόχου στην προηγούμενη επανάληψη. Δηλαδή αν για παράδειγμα οι ποσότητες που αντιστοιχεί ένα χρωμόσωμα δεδομένου παίκτη ήταν από 0 εώς 10 και μετά την εφαρμογή του αλγορίθμου προσδιορισμού των διαδοχικών βέλτιστων απαντήσεων (του εσωτερικού βρόχου δηλαδή), για τις επαναλήψεις από την 100^{η} και μετά, η ελάχιστη ποσότητα που προσδιορίστηκε για τον παίκτη αυτόν ήταν 4 και η μέγιστη 5, τότε στην επόμενη επανάληψη του εξωτερικού βρόχου, η ελάχιστη και μέγιστη εφικτή τιμή που θα μπορούν να πάρουν τα χρωμοσώματα του παίκτη αυτού, θα είναι 4 και 5 αντίστοιχα. Η μέθοδος αυτή μας επιτρέπει να υπολογίζουμε την ισορροπία κατά Nash με οποιαδήποτε επιθυμητή ακρίβεια, με δεδομένη βέβαια την σύγκλιση της ακολουθίας διαδοχικών βέλτιστων απαντήσεων στην ισορροπία. Ο προσδιορισμός διαφορετικών συνόλων τιμών για τα χρωμοσώματα του κάθε παίκτη είναι προφανώς αναγκαίος στην περίπτωση που έχουμε μη συμμετρικό παίγνιο.

Αν από την άλλη η συνθήκη τερματισμού ικανοποιείται (ο αλγόριθμος τερματίζει εφόσον δεν υπάρχει μεταβολή στις προσδιοριζόμενες ποσότητες για κάποιον αριθμό εσωτερικών επαναλήψεων), δείξαμε ότι με πιθανότητα που τείνει στο 1 καθώς ο αριθμός των εσωτερικών επαναλήψεων τείνει στο άπειρο, η λύση στην οποία καταλήγει ο αλγόριθμος είναι η ισορροπία κατά Nash.

Τα παραπάνω ισχύουν, όπως είδαμε, ακόμα και στην περίπτωση που το παίγνιο έχει “local NE traps” (Son and Baldick 2004). Στο παίγνιο με “local NE traps” που αναλύσαμε (και το οποίο δεν είναι παίγνιο Cournot με την στενή έννοια) έχουν ήδη δοκιμαστεί αλγόριθμοι διαδοχικών βέλτιστων απαντήσεων που βασίζονται σε τοπική αναζήτηση, οι οποίοι δεν κατάφεραν να ανακαλύψουν την ισορροπία κατά Nash, λόγω της ύπαρξης των τοπικών αυτών ελαχίστων. Οι Son and Baldick (2004) έδειξαν επιπρόσθετα ότι ούτε το co-evolutionary programming που χρησιμοποιήσαμε στα προηγούμενα κεφάλαια ήταν αποτελεσματικό. Ο λόγος που συνέβη αυτό είναι εύκολα κατανοητός, με βάση την ανάλυση του κεφαλαίου 5. Όπως είδαμε στο κεφάλαιο αυτό, οι ποσότητες ισορροπίας για κάθε μεμονωμένο παίκτη διαφέρουν μεταξύ τους, στον αλγόριθμο ατομικής μάθησης του co-evolutionary programming. Συνεπώς, παρόλο που έχουμε σύγκλιση της μέσης και της συνολικής ποσότητας στην ποσότητα του NE, η ισορροπία κατά Nash δεν παρατηρείται ποτέ στο παίγνιο -αφού οι παίκτες χρησιμοποιούν διαφορετικές στρατηγικές από εκείνη που αντιστοιχεί στο NE. Συνεπώς ο αλγόριθμος του co-evolutionary programming, όπως έχει μέχρι στιγμής παρουσιαστεί στην βιβλιογραφία, είναι ακατάλληλος για την επίλυση παιγνίων Cournot (την εύρεση δηλαδή της ισορροπίας κατά Nash). Αλλά και οι εκδόσεις κοινωνικής μάθησης που εισάγαμε στο κεφάλαιο 6, προϋποθέτουν την ύπαρξη κοινών συναρτήσεων κόστους στο παίγνιο Cournot ή γενικότερα την συμμετρία του παιγνίου, για να εφαρμοστούν. Έτσι κανείς από τους συν-εξελικτικούς γενετικούς αλγόριθμους που έχουν προταθεί μέχρι στιγμής, δεν μπορεί να επιλύσει το παίγνιο των Son and Baldick (2004).

Η μοναδική περίπτωση στην οποία η εύρεση του σημείου ισορροπίας δεν ήταν εφικτή από τον επαναληπτικό γενετικό μας αλγόριθμο, ήταν στο υπόδειγμα της Arifovic (1994) παραπάνω, με τις παραμέτρους προσδιορισμένες έτσι ώστε η ακολουθία διαδοχικών βέλτιστων απαντήσεων να αποκλίνει. Σε αυτήν την περίπτωση ο αλγόριθμος αποκλίνει, με τις απαντήσεις των δύο παικτών να σταθεροποιούνται στις τιμές 0 (που είναι η ελάχιστη επιτρεπόμενη τιμή) και στην βέλτιστη απάντηση στην τιμή 0 (που είναι περίπου 270). Σε περίπτωση μάλιστα που επιτρέπαμε στις ποσότητες να πάρουν αρνητικές τιμές, οι ποσότητες αναμένεται ότι θα απόκλιναν εντελώς (στο $-\infty$ και $+\infty$ αντίστοιχα). Αυτό βέβαια είναι μια αναμενόμενη αδυναμία του αλγορίθμου, αφού είναι μια «αδυναμία» της θεωρητικής ακολουθίας διαδοχικών βέλτιστων απαντήσεων, στην οποία βασίζεται, ως επαναληπτικός αλγόριθμος.

Το σημαντικό λοιπόν συμπέρασμα είναι ότι, τόσο σε παίγνια στα οποία δεν υπάρχουν τοπικά ελάχιστα στον συνδιασμό των συναρτήσεων κέρδους, όσο και στα παίγνια με τέτοιες local NE traps, ο αλγόριθμος συγκλίνει στην ισορροπία κατά Nash, με όσο ικανοποιητική ακρίβεια επιθυμούμε και με ικανή προϋπόθεση τη σύγκλιση της θεωρητικής ακολουθίας βέλτιστων απαντήσεων, σε αντίθεση με τους άλλους επαναληπτικούς αλγόριθμους που έχουν προταθεί μέχρι στιγμής.

Συνοψίζοντας, στην μελέτη αυτή εισάγαμε 3 διαφορετικούς πρωτότυπους αλγορίθμους: δύο συν-εξελικτικούς γενετικούς αλγορίθμους κοινωνικής μάθησης πολλών πληθυσμών για συμμετρικά παίγνια Cournot και έναν ευρετικό αλγόριθμο που βασίζεται στους γενετικούς αλγόριθμους βελτιστοποίησης για μη συμμετρικά παίγνια με συγκλίνουσα ακολουθία διαδοχικών βέλτιστων απαντήσεων. Η αποτελεσματικότητα των αλγορίθμων αυτών υπερβαίνει την αποτελεσματικότητα των ήδη προταθέντων γενετικών αλγορίθμων για το πρόβλημα της σύγκλισης των αποφάσεων των παικτών στην ισορροπία κατά Nash του παιγνίου Cournot. Εκτός αυτού εισάγαμε ένα τρόπο «μέτρησης» της απόκλισης των καταστάσεων στις οποίες βρίσκεται το σύστημα κάθε στιγμή, που βασίζεται στην έννοια των ομαδικών καταστάσεων (Kemeny and Snell 1960), της αλυσίδας Markov που περιγράφει την στοχαστική εξέλιξη των αλγορίθμων (και των γενετικών αλγορίθμων μάθησης γενικότερα), από την κατάσταση στην οποία όλα τα χρωμοσώματα όλων των πληθυσμών είναι ίσα με το χρωμόσωμα που αντιστοιχεί στην ισορροπία κατά Nash (κατάσταση Nash). Επίσης διερευνήσαμε τις ιδιότητες σύγκλισης των ποσοτήτων των μεμονωμένων πρακτόρων -κάτι που δεν είχε γίνει μέχρι στιγμής- και μάλιστα δεν αρκεστήκαμε στην απλή παράθεση της αναμενόμενης τιμής και της διακύμανσης που δεν έχουν προβλεπτική ικανότητα στο συγκεκριμένο πρόβλημα, λόγω της μη ανεξαρτησίας των διαδοχικών παρατηρήσεων των αλυσίδων Markov, αλλά προχωρήσαμε στην εκτίμηση σημαντικών παραμέτρων των εργόδικων αλυσίδων που χαρακτηρίζουν τους αλγόριθμους που μελετήσαμε, όπως τις αναμενόμενες συχνότητες της κατάστασης Nash και των ομαδικών καταστάσεων που βρίσκονται γύρω από αυτήν, τους αναμενόμενους χρόνους επαναφοράς και πρώτης μετάβασης από δεδομένη αρχική κατάσταση και λοιπά. Τέλος, αφού παρατηρήσαμε ότι, για κατάλληλα σύνολα παραμέτρων, σε ένα μεγάλο μέρος των παίγνιων που εκτελούνται κατά τη διάρκεια της προσομοίωσης οι παίκτες χρησιμοποιούν στρατηγικές Nash και τα παίγνια βρίσκονται σε ισορροπία, δείξαμε τον τρόπο με τον οποίο οι συγκεκριμένοι αλγόριθμοι μπορούν να χρησιμοποιηθούν για την εύρεση της ισορροπίας (να χρησιμοποιηθούν δηλαδή σαν ευρετικοί αλγόριθμοι, πέρα από την ικανότητά τους της προσομοιώσης μοντέλων τεχνητών πρακτόρων).

Όσον αφορά τον ευρετικό αλγόριθμο βελτιστοποίησης που εισάγαμε για τα παίγνια με συγκλίνουσα ακολουθία διαδοχικών βέλτιστων απαντήσεων, αφού δείξαμε ότι η ικανοποίηση της συνθήκης τερματισμού συνεπάγεται την εύρεση με πιθανότητα που τείνει στο 1 της ισορροπίας, δείξαμε ότι ο αλγόριθμος βρίσκει την ισορροπία (αν οι διαδοχικές απαντήσεις συγκλίνουν) και μάλιστα μπορεί να αντιμετωπίσει με επιτυχία το πρόβλημα των “local NE traps” (Son and Baldick 2004) ενώ, με την εισαγωγή του εξωτερικού βρόχου πετύχαμε τον υπολογισμό της ισορροπίας κατά Nash με όποια επιθυμητή ακρίβεια.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Alander J.T. (1992): “On Optimal Population Size of Genetic Algorithms”. Proceedings of CompEuro 92, pp. 65-70. IEEE Computer Society Press.

Alkemade, F., La Poutre, H. and Amman H. (2007): “On Social Learning and Robust Evolutionary Algorithm Design in the Cournot Oligopoly Game”. Computational Intelligence 23, pp.162-175.

Alos-Ferrer, C. and Ania, A. (2005). “The Evolutionary Stability of Perfectly Competitive Behaviour”. Economic Theory 26, pp. 497-516.

Amir, R. (1996): “Cournot Oligopoly and the Theory of Supermodular Games”. Games and Economics Behaviour 15, pp. 132-148.

Arifovic, J. (1994): “Genetic Algorithm Learning and the Cobweb Model”. Journal of Economic Dynamics and Control 18, pp. 3-28.

Arifovic, J. and Maaschek M. (2006): “Revisiting Individual Evolutionary Learning in the Cobweb Model – An Illustration of the Virtual Spite-Effect”. Computational Economics 28, pp. 333- 354.

Arthur, W.B., Durlauf, S.N., and Lane, D.A., eds. (1997): “The Economy as an Evolving Complex System II”. In Proceedings Vol. XXVII, SFI Studies in the Sciences of Complexity. Addison-Wesley MA.

Axelrod, Robert (1984) : The Evolution of Cooperation. ISBN 0-465-02121-2.

Barr, J. and Saraceno, F. (2002): “A Computational Theory of the Firm”. Journal of Economic Behavior and Organization 49, pp. 345-361.

Barr, J. and Saraceno, F. (2005): “Cournot Competition, Organization and Learning”. Journal of Economic Dynamics and Control 29, pp. 277-295.

Basawa IV, Rao P (1980): Statistical Inference for Stochastic Processes. Academic Press, London.

Batten, D. (2000): Discovering Artificial Economics: How Agents Learn and Economies Evolve. Boulder Co: Westview Press.

Cournot, A.A.(1838): Recherches sur les principes mathématiques de la théorie des richesses (Researches into the Mathematical Principles of the Theory of Wealth). (1897, Engl. trans. by N.T. Bacon).

Dawid, H. (1996): Adaptive Learning by Genetic Algorithms. Springer, Berlin.

Dawid, H. and Kopel, M. (1998): “On Economic Applications of the Genetic Algorithm: A Model of the Cobweb Type”. Journal of Evolutionary Economics 8, pp. 297-315.

Dawid, H. (1999): Adaptive Learning by Genetic Algorithms, 2nd ed. Springer, Berlin.

Day, D., and Chen, P. (1993): Nonlinear Dynamics and Evolutionary Economics. Oxford University Press.

Debreu, D.(1952): “A Social Equilibrium Existence Theorem”. Proceedings of the National Academy of Sciences 38, pp. 886-893.

Dindos M., Mezzetti C. (2003): “Better Reply Dynamics and Global Convergence to Nash Equilibrium in Aggregative Games”. Working Paper, University of North Carolina.

Dubey, P., Haimanko, O. and Zapechelnyuk, A. (2006): “Strategic Complements and Substitutes and Potential Games”. Games and Economic Behaviour 54, pp. 77-94.

Duffy,John (2004): “Agent-Based Models and Human Subjects Experiments”. University of Pittsburg Working Papers, April 2004.

Epstein, J.M., and Axtell, R. (1996): Growing Artificial Societies: Social Science from the Bottom Up. MIT Press, Cambridge MA.

Franke, R. (1998): “Co-evolution and Stable Adjustments in the Cobweb Model”. Journal of Evolutionary Economics 8, pp. 383-406.

Friedman, D. (1998): “On Economic Applications of Evolutionary Game Theory”. Journal of Evolutionary Economics 8: 15-43.

Foster, D. and Young, P. (1990): “Stochastic Evolution Game Dynamics”. Theoretical Population Biology 38.

Fundenberg,D. and Tirole, J. (1991): Game Theory. The MIT Press, Cambridge MA.

Glicksberg I.L.(1952): “A Further Generalization of the Kakutani Fixed Point Theorem with Application to Nash Equilibrium Points”. Proceedings of the National Academy of Sciences 38, pp. 170-174.

Goldberg, D.E. (1988): Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA.

Goodman, R. (1988): Introduction to Stochastic Models. Benjamin-Cummings, CA.

Jensen, M. (2007): “Stability of Pure Strategy Nash Equilibrium in Best-Reply Potential Games”.

Hamilton, W. (1970): “Selfish and Spiteful Behaviour in an Evolutionary Model”. Nature 228, pp. 1218-1255.

Hayek,F.A. (1948): Individualism and Economic Order. University of Chicago Press.

Hirsch, M. and Smale, S. (1974): Differential Equations, Dynamical Systems and Linear Algebra. Academic Press.

Hobbs B.F., Metzler C.B. and Pang J.S.(2000): “Strategic gaming analysis for electric power systems: An MPEC approach”. IEEE Transactions on Power Systems 15, pp. 638-645.

Holland, J.H. (1975): Adaptation in Natural and Artificial Systems. University of Michigan Press.

Holland, J. (1992): Adaptation in Natural and Artificial Systems. MIT Press, Cambridge MA.

Isaacson, D.L., and Madsen, R.W. (1976): Markov Chains, Theory and Applications. Wiley, NY.

Kemeny J.G., Snell J.L. (1960): Finite Markov Chains. D.Van Nostrand Company, Inc. Princeton.

Khan, M. Ali (1985): “On Extensions of the Cournot-Nash Theorem”. Advances in Equilibrium Theory (C.D. Aliprantis et al. eds.), Springer Verlag.

Khan, M.Ali, Παπαγεωργίου, N. (1987): “On Cournot-Nash Equilibria in Generalized Qualitative Games with an Atomless Measure Space of Agents”. Proceedings of the American Mathematics Society, Vol. 100, No 3 (July 1987), pp. 505-510.

Kim, T., Prikry, K., και Γιανέλλης, N. (1985): “Equilibria in Abstract Economies with a Measure Space and with an Infinite Dimensional Strategy Space”. Univ. of Minessota Disc. Paper No 1128, March 1985.

Krugman, P. (1996): The Self-Organizing Economy. Blackwell Publishers, Cambridge MA.

Leontief, W. (1934): “Verzogerte Angebotsanpassung und Partielles Gleichgewicht”. Zeitschrift fur Nationalokonomie 5: pp. 670-676.

Marimon, R. et al. (1993): “Adaptive Learning, Evolutionary Dynamics and Equilibrium Selection in Games”. European Economic Review 37: 603-611.

McManus, M. (1962): “Number and Size in Cournot Equilibrium”. Yorkshire Bulletin of Economic Social Research 14, pp. 14-22.

McManus, M. (1964): “Equilibrium, Number and Size in Cournot Oligopoly”. Yorkshire Bulletin of Economic Social Research 16, pp. 68-75.

Milgrom, P. and Roberts, J. (1990): “Rationalizability Learning and Equilibrium in Games with Strategic Complementarities”. Econometrica 58: pp. 1255-1277.

Miller, J. (1996): “The Co-evolution of Automata in the Repeated Prisoner’s Dilemma”. Journal of Economic Behavior and Organization 29, pp. 87-112.

Mitchell, N. (1996): An Introduction to Genetic Algorithms. The MIT Press, MA.

Muth, J.F. (1961): “Rational Expectations and the Theory of Price Movements”. Econometrica 29, pp. 315-335.

Nash, J. (1950): “The Bargaining Problem”. Econometrica 18, pp. 155-162.

Nash, J. (1950): "Equilibrium points in n-person games". Proceedings of the National Academy of Sciences 36(1), pp. 48-49

Nash, J. (1950): “Non-cooperative Games”. The Annals of Mathematics 54(2), pp. 286-295.

Nix, A.E., and Vose, M.D. (1992): “Modeling Genetic Algorithms with Markov Chains”. Annals of Mathematics and Artificial Intelligence 5: 79-88.

Novshek (1985): “On the Existence of Nash Equilibrium”. Rev. Economic Studies 52, pp. 85-98.

Olsen, M. (1965): The Logic of Collective Action. Harvard University Press.

Porter,R., Nudelman, E., and Stoham, Y. (2004): “Simple Search Methods for Finding Nash Equilibrium”. Proceedings of AAAI-04, pp. 664-669.

Price T.C (1997): “Using Co-Evolutionary Programming to Simulate Strategic Behavior in Markets”. Journal of Evolutionary Economics, Vol. 7, pp. 219-254.Nash, J. (1950): “The Bargaining Problem”. Econometrica 18, pp. 155-162.

Rhode, P. and Stageman, M. (1995): “Evolution through Imitation (with Applications in Duopoly)”. Working Paper E95-43, Virginia Polytechnic Institute and State University, Department of Economics.

Riechmann, T. (1997): “A Genetic Algorithm Model of Natural Monopoly”. Mimeo.

Riechmann, T. (1999): “Learning and Behavioral Stability”. Journal of Evolutionary Economics 9: 225-242.

Riechmann, T. (2001): “Genetic Algorithm Learning and Evolutionary Games”. Journal of Economic Dynamics & Control 25: 1019-1037.

Roberts, J. and Sonnenschein, H. (1976): “On the Existence of Cournot Equilibrium without Concave Profit Functions”. Journal of Economic Theory 13, pp. 112-117.

Sandholm, T., Gilpin, A., and Conitzer, V. (2005): “Mixed-Integer Programming Methods for Finding Nash Equilibria”. Proceedings of AAAI-05, pp. 495-501.

Sargent, T. (1993): “Bounded Rationality in Macroeconomics”. The Arne Ryde Memorial Lectures. Clarendon Press, Oxford UK.

Schaffer, M. (1988): “Evolutionary Stable Strategies for a Finite Population and a Variable Contest Size”. Journal of Theoretical Biology 132: 469-478.

Schelling, T.C. (1978): Micromotives and macrobehavior. W.W. Norton and Co.

Smith, A. (1937) : An Inquiry into the Nature and Causes of the Wealth of Nations. New York: Cannan Edition, American Modern Library Series.

Smith, John Maynard and Price, George (1973) : “The Logic of Animal Conflict”. Nature 246, pp 15-18.

Smith, John Maynard (1982) : Evolution and the Theory of Games. ISBN 0-521-28884-3

Socha J., Norton P.(1994): Το βιβλίο της Assembly για PC. Peter Norton Press, MA.

Son Y.S and Baldick R. (2004): “Hybrid Coevolutionary Programming for Nash Equilibrium Search in Games With Local Optima”. IEEE Transactions on Evolutionary Computation 8, pp. 305-315.

Sugden, R.(1986): The Economics of Rights, Cooperation and Welfare. Blackwell Publishing.

Thomas B. (1984) : “Evolutionary Stability : States and Strategies”. Theoretical Population Biology 26, pp 49-67.

Thomas B. (1985) : “On Evolutionary Stable Sets”. Journal of Mathematical Biology 22, pp 105-115.

Topkis D.M (1998): Supermodularity and Complementarity. Princeton University Press, Princeton.

Valee, T. And Yildizoglou, M. (2007): “Convergence In Cournot Oligopoly with Social and Individual Learning”. Groupe de Recherche en Economie Theorique et Appliquee (GREtha) Working Paper no 2007-07 (June 2007).

Vega-Redondo, F. (1997): “The Evolution of Walrasian Behaviour”. Econometrica 65, pp. 375-384.

Vriend, N.J. (2000): “An Illustration of the Essential Difference between Individual and Social Learning, and it’s Consequences in Computational Analyses”. Journal of Economic Dynamics and Control 24, pp. 1-19.

Vives, X. (1990): “Nash Equilibrium with Strategic Complementarities”. Journal of Mathematical Economics 19, pp. 305-321.

Vives, X. (2000): Oligopoly Pricing: Old Ideas and New Tools. The MIT Press, Cambridge MA.

Weber D.J. and Overbye T.J. (1999): “A two-level optimization problem for analysis of market bidding strategies” in Proc. Power Eng. Soc. Summer Meeting 1999, vol. 2, 1999, pp. 846–851.

Weibull, J. (2006): Evolutionary Stability and the Replicator Dynamics, Lecture Notes, Stockholm School of Economics.

Wikipedia Community (2007): “Compact Space”. Wikipedia 7/10/2007. http://en.wikipedia.org/wiki/Compact_Space

Wikipedia Community (2007): “Cournot’s Competition”. Wikipedia 5/10/2007. http://en.wikipedia.org/wiki/Cournot_competition

Wikipedia Community (2007): “Evolutionary Stable Strategy”. Wikipedia (30/10/2007). http://en.wikipedia.org/wiki/Evolutionarily_stable_strategy

Wilson, C. (2005): “Concave and Quasi-Concave Functions”, ver. 31.0006. Prof. Wilson’s homepage, NY University, 12/10/2007.
http://www.wilsonc.econ.nyu.edu/UMath/Handouts/ums06h23convexsetsandfunctions.pdf

Young, H.P. (1998): Individual Strategy and Social Structure. Princeton University Press.

Γιανέλλης, N. and Prabhakar, N. (1983): “Existence of maximal Elements and Equilibria in Linear Topological Spaces”. Journal of Mathematical Economics 12, pp. 233-245.

Παράρτημα – Κώδικας Matlab

1. Arifovic 1994, multi-population algorithm

```
numberIterations=10000;   %number of generations
numberFirms=20;         %number of firms
populationSize= 50; %number of chromosomes in population
chromosomeSize=20;

%Markov chain variables
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;


firmQuantityHistory=zeros(numberIterations,numberFirms);
populationMatrix=zeros(numberFirms,populationSize,chromosomeSize);
firmProfitHistory=zeros(numberIterations,numberFirms);

%%%%%initializing variables and parameters

x_cost=56;
y_cost=0;
%Cost = xq+0.5yq^2

a_demand=256;
b_demand=1;
%price=a-b(q_1+q_2+...+q_n)


%genetic algorithm
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.01; %probability that a childs'bit undergoes mutation


%solutions' space
maxQuantity=30;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);
    %quantities should be such, that price will remain positive
%target should be 1/3 of maxQ
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end

%%%%%Initialize populations and evaluate of fitness function

for i=1:numberFirms
    for j=1:populationSize  %this loop runs for all chromosomes in
        %the initial population
        c=0;     %a handy variable
        for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

            %assingment of random values
            if rand <0.5
                populationMatrix(i,j,k)=0;
            else
                populationMatrix(i,j,k)=1;
            end
```

```
        end %for j=1:chromosomeSize
    end %for j=1:populationSize
end %for i=1:numberFirms

    %calculation of the total quantity, the cost and the price.
totalQuantity=0;
for i=1:numberFirms
   index=ceil(populationSize*rand);
   for j=i:populationSize
       q=0;
       for k=1:chromosomeSize
            q=q+populationMatrix(i,j,k)*2^(k-1);
            %calculate quantity
       end %for j=i:chromosomeSize
       q=q*normalizationFactor; %normalize quantity
       %totalQuantity=totalQuantity+q;
       %calculate total quantity
       populationQuantity(i,j)=q;
       populationCost(i,j)=x_cost*q+1/2*y_cost*q^2;
       %calculate cost
   end %for j=1:populationSize
   totalQuantity=totalQuantity+populationQuantity(i,index);
end %i=1:numberFirms
price= a_demand-b_demand*totalQuantity; %calculate price
for i=1:numberFirms %calculate profit and fitness
    for j=1:populationSize
        populationProfit(i,j)=price*populationQuantity(i,j)-populationCost(i,j);
    end
    [b,ix]=sort(populationProfit(i,:)); %sort in ascending order, get indices in
ix
    for j=1:populationSize
        populationFitness(i,ix(j))=j;
        %chrom with min profit gets fitness 1, next
        %gets 2, etc.
    end
end %for i=1:numberFirms


%%%% main algorithm

for generationNumber=1:numberIterations % Now the main algorithm starts,
    % and it executes for the given time of
    %iterations, stored in numberIterations
    for i=1:numberFirms
        for s=1:fix(populationSize/2)   %s is half the population size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

            c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
            for j=1:populationSize
                c=c+populationFitness(i,j);
                cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
            end
```

```
if rand<probabilityCrossover
    doCrossover=1;
else
    doCrossover=0;
end     %determine if we are going to do crossover or not

%picking first parent
x=c*rand;   %x is a random number between 0 and total cumulative
    %fitness
flag=1; %this flag is used to indicate if we are still searching
    %for the parent. If not, it is set to 0 and the loop remains
    %idle in the following iterations. I used this because there
    %were some problems when I used a "break" statement
for j=1:populationSize  %loop begins
    if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
        %the cumulative fitness until chromosome i is greater or
        %equal than x. In this case we pick chromosome i to be the
        %first parent. It also checks the value of the flag, since
        %the succeeding chromosomes (after i) will also have
        %cumulative fitness greater or equal than x, so we must
        %set the flag to 0 after we pick the parent, else the last
        %chromosome of the population will be always picked after
        %the last iteration.
        for k=1:chromosomeSize  %this loop sets the bits of the
            %parent1 variable, according to the bits of the
            %parent chromosome in the population pool
            parent1(k)=populationMatrix(i,j,k);
        end % for j=1:chromosomeSize
        flag=0;
    end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

%the same loop applies for the second parent as well. The
%difference is of course in the variable we use to store
%the second parent.
x=c*rand;
flag=1;
for j=1:populationSize
    if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(i,j,:);  %this is an easier
        %way to set a vector variable. We use the : operator
        %which means "all" (it is used in every "for" loop).
        %Therefore we assign the parent2 vector from the
        %corresponding vector of the population matrix, with
        %just one comand.
    end
end
%notice the fact that we keep the current selected parents,
%in the "Parents' pool", i.e. a chromosome can become a
%parent more than once.

if doCrossover==0   %no crossover at all
    child1=parent1;
    child2=parent2;
else    %doCrossover==1
    %we pick a random cutting point for crossover 1...n
    cuttingPoint=ceil(rand*chromosomeSize);

    %1st child generation.
    %crossover
```

```
            for k=1:chromosomeSize
                if k<=cuttingPoint +0.5
                    child1(k)=parent1(k);
                else
                    child1(k)=parent2(k);
                end %j<=cuttingpoint
            end %for k=1:chromosomeSize

        %%We do the same to create the 2nd child,
            for k=1:chromosomeSize
                if k<cuttingPoint+0.5
                    child2(k)=parent2(k);
                else
                    child2(k)=parent1(k);
                end
            end
        end %if doCrossover

        %mutation
        for k=1:chromosomeSize
            if rand<probabilityMutation %child is mutated with probability
                child2(k)=~child2(k);
            end
            if rand<probabilityMutation
                child1(k)=~child1(k);
            end
        end %for


        %store children in children matrix and calculate their quantities
        p1=fix(2*s-1);  %since a pairing produces 2 children
        p2=2*s;
        q1=0;q2=0;
        for k=1:chromosomeSize
            childrenMatrix(i,p1,k)=child1(k);
            childrenMatrix(i,p2,k)=child2(k);
            q1=q1+childrenMatrix(i,p1,k)*2^(k-1);
            q2=q2+childrenMatrix(i,p2,k)*2^(k-1);
        end
    end %pairing loop for s=1:fix(populationSize/2)
end %for i=1:numberFirms


%we update the population matrices, from the offsping matrices, in
%order to use the next generation as the population in the succesive
%iteration.
populationMatrix=childrenMatrix;

%calculate each firm's active quantity for the current
%round. A chromosome is picked with probability proportional
%to it's fitness, as it is determined by the price of the
%previous round.
totalQuantity=0;
for i=1:numberFirms
    for j=i:populationSize
        q=0;
        for k=1:chromosomeSize
            q=q+populationMatrix(i,j,k)*2^(k-1);
            %calculate quantity
        end %for j=i:chromosomeSize
        q=q*normalizationFactor; %normalize quantity
        populationQuantity(i,j)=q;
```

```
            populationCost(i,j)=x_cost*q+1/2*y_cost*q^2;
        end %for j=1:populationSize
    end %i=1:numberFirms

    for i=1:numberFirms %calculate profit and fitness
        for j=1:populationSize
            populationProfit(i,j)=price*populationQuantity(i,j)-
populationCost(i,j);
        end
        [b,ix]=sort(populationProfit(i,:)); %sort in ascending order, get
indices in ix
        for j=1:populationSize
            populationFitness(i,ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
        end
        c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
        for j=1:populationSize
            c=c+populationFitness(i,j);
            cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
        end
        %pick active quantity
        x=c*rand;   %x is a random number between 0 and total cumulative
            %fitness
        flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
        q=0;
        for j=1:populationSize  %loop begins
            if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
                    %the cumulative fitness until chromosome i is greater or
                    %equal than x. In this case we pick chromosome i to be the
                    %first parent. It also checks the value of the flag, since
                    %the succeeding chromosomes (after i) will also have
                    %cumulative fitness greater or equal than x, so we must
                    %set the flag to 0 after we pick the parent, else the last
                    %chromosome of the population will be always picked after
                    %the last iteration.
             for k=1:chromosomeSize  %this loop sets the bits of the
                        %parent1 variable, according to the bits of the
                        %parent chromosome in the population pool
                q=q+populationMatrix(i,j,k)*2^(k-1);
                    %calculate quantity
             end % for j=1:chromosomeSize
             q=q*normalizationFactor; %normalize quantity
             flag=0;
           end %if ( x<cumulativeFitness(i) && flag==1 )
       end %for j=1:populationSize
       firmQuantityHistory(generationNumber,i)=q;
       firmProfitHistory(generationNumber,i)=price*q-x_cost*q-1/2*y_cost*q^2;
       totalQuantity=totalQuantity+q;
    end %for i=1:numberFirms

    price= a_demand-b_demand*totalQuantity; %calculate new price
```

```
    %calculate the fitness values of all of the chromosomes in the
    %population, using the new price
    for i=1:numberFirms
        for j=i:populationSize
            q=0;
            for k=1:chromosomeSize
                q=q+populationMatrix(i,j,k)*2^(k-1);
                %calculate quantity
            end %for j=i:chromosomeSize
            q=q*normalizationFactor; %normalize quantity
            populationQuantity(i,j)=q;
            populationCost(i,j)=x_cost*q+1/2*y_cost*q^2;
        end %for j=1:populationSize
    end %i=1:numberFirms
    for i=1:numberFirms %calculate profit and fitness
        for j=1:populationSize
            populationProfit(i,j)=price*populationQuantity(i,j)-
populationCost(i,j);
        end
        [b,ix]=sort(populationProfit(i,:)); %sort in ascending order, get
indices in ix
        for j=1:populationSize
            populationFitness(i,ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
        end
    end %for i=1:numberFirms

   %measure the hebbian distance of the current population
   d=0;
   for firm=1:numberFirms
        for chrom=1:populationSize
            for k=1:chromosomeSize
                if populationMatrix(firm,chrom,k)~=target(k)
                    d=d+1;
                end
            end
        end
   end
   %update transitionMatrix,fixedVector and currentState
    md=ceil(d/(numberFirms*populationSize));
    if generationNumber>1
        transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
    end
    fixedVector(md+1)=fixedVector(md+1)+1;
    currentState=md;

    priceHistory(generationNumber)=price;
    averageFirmProfitHistory(generationNumber)=mean(firmProfitHistory(generation
Number,:));
    averageMarketQuantityHistory(generationNumber)=mean(firmQuantityHistory(gene
rationNumber,:));


if mod(generationNumber,10)==0
    generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
```

```
for i=1:chromosomeSize+1
    fixedVector(i)=fixedVector(i)/generationNumber;
    for j=1:chromosomeSize+1
        transitionMatrix(i,j)=transitionMatrix(i,j)/generationNumber;
    end
end
```

2. Vriend's multi-population, individual learning algorithm

```
%cost function
x_cost=56;
y_cost=0;
numberFirms=20;        %number of firms
%Cost = xq, y=0 always in alkemade et al

%demand curve
a_demand=256;
b_demand=1;
%price=a-(q_1+q_2+...+q_n) %b=1 always in Alkemade et al (2007)


%genetic algorithm
populationSize= 50; %number of chromosomes in population
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.001; %probability that a childs'bit undergoes mutation
numberIterations=100000;   %number of generations
chromosomeSize=20;

%Markov chain variables
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;

%epoch
GARate=50;

%solutions' space
maxQuantity=24;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);

%target chromosome = Qmax/3
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end

firmQuantityHistory=zeros(numberIterations,numberFirms);
averageMarketQuantityHistory=zeros(1,numberIterations);
%%%%%Initialize populations

for i=1:numberFirms
    for j=1:populationSize  %this loop runs for all chromosomes in
        %the initial population
        c=0;    %a handy variable
        for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

            %assingment of random values
            if rand <0.5
                populationMatrix(i,j,k)=0;
            else
                populationMatrix(i,j,k)=1;
```

```
            end
            populationProfit(i,j)=rand; %random initial profits
        end %for j=1:chromosomeSize
    end %for j=1:populationSize
end %for i=1:numberFirms

%%%% main algorithm
for generationNumber=1:numberIterations % Now the main algorithm starts,
    % and it executes for the given time of
    %iterations, stored in numberIterations

    %calculation of the total quantity, the cost and the price.
    totalQuantity=0;
    for i=1:numberFirms
        j=ceil(populationSize*rand);
        currentIndex(i)=j;
        q=0;
        for k=1:chromosomeSize
            q=q+populationMatrix(i,j,k)*2^(k-1);
                %calculate quantity
        end %for j=i:chromosomeSize
        q=q*normalizationFactor; %normalize quantity
        totalQuantity=totalQuantity+q;
        firmQuantityHistory(generationNumber,i)=q;
        populationQuantity(i,j)=q;
        populationCost(i,j)=x_cost*q+1/2*y_cost*q^2;
            %calculate cost
    end %i=1:numberFirms
    price= a_demand-b_demand*totalQuantity; %calculate price
    for i=1:numberFirms %calculate profit and fitness
        populationProfit(i,currentIndex(i))=price*populationQuantity(i,currentIn
dex(i))-populationCost(i,currentIndex(i));
    end %for i=1:numberFirms
    averageMarketQuantityHistory(generationNumber)=mean(firmQuantityHistory(gene
rationNumber,:));
    %priceHistory(generationNumber)=price;

if mod(generationNumber,GARate)==0
    for i=1:numberFirms %calculate fitness
        [b,ix]=sort(populationProfit(i,:)); %sort in ascending order, get
indices in ix
        for j=1:populationSize
            populationFitness(i,ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
        end
    end %for i=1:numberFirms

    for i=1:numberFirms
        for s=1:fix(populationSize/2)   %s is half the population size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

            c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
            for j=1:populationSize
                c=c+populationFitness(i,j);
                cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
```

```
    %at every iteration, so the vector holds the values of the
    %cumulative fitness until the given chromosome of the
    %population. c also holds the value of the total cumulative
    %fitness after the last iteration.
end

if rand<probabilityCrossover
    doCrossover=1;
else
    doCrossover=0;
end     %determine if we are going to do crossover or not

%picking first parent
x=c*rand;   %x is a random number between 0 and total cumulative
    %fitness
flag=1; %this flag is used to indicate if we are still searching
    %for the parent. If not, it is set to 0 and the loop remains
    %idle in the following iterations. I used this because there
    %were some problems when I used a "break" statement
for j=1:populationSize  %loop begins
    if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
        %the cumulative fitness until chromosome i is greater or
        %equal than x. In this case we pick chromosome i to be the
        %first parent. It also checks the value of the flag, since
        %the succeeding chromosomes (after i) will also have
        %cumulative fitness greater or equal than x, so we must
        %set the flag to 0 after we pick the parent, else the last
        %chromosome of the population will be always picked after
        %the last iteration.
        for k=1:chromosomeSize  %this loop sets the bits of the
            %parent1 variable, according to the bits of the
            %parent chromosome in the population pool
            parent1(k)=populationMatrix(i,j,k);
        end % for j=1:chromosomeSize
        flag=0;
    end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

%the same loop applies for the second parent as well. The
%difference is of course in the variable we use to store
%the second parent.
x=c*rand;
flag=1;
for j=1:populationSize
    if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(i,j,:);  %this is an easier
        %way to set a vector variable. We use the : operator
        %which means "all" (it is used in every "for" loop).
        %Therefore we assign the parent2 vector from the
        %corresponding vector of the population matrix, with
        %just one comand.
    end
end
%notice the fact that we keep the current selected parents,
%in the "Parents' pool", i.e. a chromosome can become a
%parent more than once.

if doCrossover==0   %no crossover at all
    child1=parent1;
    child2=parent2;
else    %doCrossover==1
```

```
            %we pick a random cutting point for crossover 1...n
            cuttingPoint=ceil(rand*chromosomeSize);

            %1st child generation.
            %crossover
            for k=1:chromosomeSize
                if k<=cuttingPoint +0.5
                    child1(k)=parent1(k);
                else
                    child1(k)=parent2(k);
                end %j<=cuttingpoint
            end %for k=1:chromosomeSize

        %%We do the same to create the 2nd child,
            for k=1:chromosomeSize
                if k<cuttingPoint+0.5
                    child2(k)=parent2(k);
                else
                    child2(k)=parent1(k);
                end
            end
        end %if doCrossover

        %mutation
        for k=1:chromosomeSize
            if rand<probabilityMutation %child is mutated with probability
                child2(k)=~child2(k);
            end
            if rand<probabilityMutation
                child1(k)=~child1(k);
            end
        end %for


        %store children in children matrix and calculate their quantities
        p1=fix(2*s-1);  %since a pairing produces 2 children
        p2=2*s;
        q1=0;q2=0;
        for k=1:chromosomeSize
            childrenMatrix(i,p1,k)=child1(k);
            childrenMatrix(i,p2,k)=child2(k);
            q1=q1+childrenMatrix(i,p1,k)*2^(k-1);
            q2=q2+childrenMatrix(i,p2,k)*2^(k-1);
        end
    end %pairing loop for s=1:fix(populationSize/2)
end %for i=1:numberFirms


%we update the population matrices, from the offsping matrices, in
%order to use the next generation as the population in the succesive
%iteration.
populationMatrix=childrenMatrix;

%Markov chain statistics
d=0;
for firm=1:numberFirms
    for chrom=1:populationSize
        for k=1:chromosomeSize
            if populationMatrix(firm,chrom,k)~=target(k)
                d=d+1;
            end
        end
```

```
        end
    end
   %update transitionMatrix,fixedVector and currentState
    md=ceil(d/(numberFirms*populationSize));
    if generationNumber/GARate>1
        transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
    end
    fixedVector(md+1)=fixedVector(md+1)+1;
    currentState=md;


end %if GARate

if mod(generationNumber,100)==0
    generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
for i=1:chromosomeSize+1
    fixedVector(i)=fixedVector(i)*GARate/numberIterations;
    for j=1:chromosomeSize+1
        transitionMatrix(i,j)=transitionMatrix(i,j)*GARate/numberIterations;
    end
end
```

3. Coevolutionary programming, individual learning version

```
%cost function
x_cost=100;
y_cost=10;
numberFirms=20;        %number of firms
%Cost = xq+y,

%demand curve
a_demand=-1;
b_demand=8300;
%price=aQ^(3/2)+b

%genetic algorithm
populationSize= 50; %number of chromosomes in population
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.001; %probability that a childs'bit undergoes mutation
numberIterations=10000;   %number of generations
chromosomeSize=20;

NashCounter=0;
%Markov chain variables
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;

%solutions' space
NashEquilibrium=19.3749; % 82.2143;
maxQuantity=3*NashEquilibrium;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);

%target chromosome = Qmax/3
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end
```

```
firmQuantityHistory=zeros(numberIterations*populationSize,numberFirms);
averageMarketQuantityHistory=zeros(1,numberIterations*populationSize);
populationCost=zeros(numberFirms,populationSize);
populationProfit=zeros(numberFirms,populationSize);
populationQuantity=zeros(numberFirms,populationSize);

matches=zeros(numberFirms,populationSize);

%%%%%Initialize populations

for i=1:numberFirms
    for j=1:populationSize  %this loop runs for all chromosomes in
        %the initial population
        c=0;    %a handy variable
        for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

            %assingment of random values
            if rand <0.5
                populationMatrix(i,j,k)=0;
            else
                populationMatrix(i,j,k)=1;
            end
            %populationProfit(i,j)=rand; %random initial profits
        end %for j=1:chromosomeSize
    end %for j=1:populationSize
end %for i=1:numberFirms

%%%% main algorithm
for generationNumber=1:numberIterations % Now the main algorithm starts,
    % and it executes for the given time of
    %iterations, stored in numberIterations

    %assingments
    totalQuantity=0;
    for i=1:numberFirms
        matches(i,:)=randperm(populationSize);
    end
    for j=1:populationSize
        totalQuantity=0;
        for i=1:numberFirms
            currentIndex=matches(i,j);
            q=0;
            for k=1:chromosomeSize
                q=q+populationMatrix(i,currentIndex,k)*2^(k-1);
                    %calculate quantity
            end %for j=i:chromosomeSize
            q=q*normalizationFactor; %normalize quantity
            totalQuantity=totalQuantity+q;
            firmQuantity(i)=q;
            firmQuantityHistory((generationNumber-1)*populationSize+j,i)=q;
            populationQuantity(i,matches(i,j))=q;
            populationCost(i,currentIndex)=x_cost*q+y_cost;
                %calculate cost
        end %i=1:numberFirms
        price= a_demand*totalQuantity^(3/2)+b_demand; %calculate price
        for i=1:numberFirms %calculate profit and fitness
            currentIndex=matches(i,j);
            populationProfit(i,currentIndex)=price*populationQuantity(i,currentI
ndex)-populationCost(i,currentIndex);
```

```
        end %for i=1:numberFirms
        averageMarketQuantityHistory((generationNumber-1)*populationSize+j)=mean
(firmQuantityHistory((generationNumber-1)*populationSize+j,:));
        %check if NE
        flag=1;
        for k=1:numberFirms
            if flag && (firmQuantity(k)== NashEquilibrium)
                flag=1;
            else flag=0;
            end
        end
        if flag==1
            NashCounter=NashCounter+1;
        end
    %priceHistory(generationNumber)=price;
    end %for j=1:populationSize

    for i=1:numberFirms %calculate fitness
        [b,ix]=sort(populationProfit(i,:)); %sort in ascending order, get
indices in ix
        for j=1:populationSize
            populationFitness(i,ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
        end
    end %for i=1:numberFirms

    for i=1:numberFirms
        for s=1:fix(populationSize/2)   %s is half the population size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

            c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
            for j=1:populationSize
                c=c+populationFitness(i,j);
                cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
            end

            if rand<probabilityCrossover
                doCrossover=1;
            else
                doCrossover=0;
            end    %determine if we are going to do crossover or not

            %picking first parent
            x=c*rand;   %x is a random number between 0 and total cumulative
                %fitness
            flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
            for j=1:populationSize  %loop begins
                if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
```

```
        %the cumulative fitness until chromosome i is greater or
        %equal than x. In this case we pick chromosome i to be the
        %first parent. It also checks the value of the flag, since
        %the succeeding chromosomes (after i) will also have
        %cumulative fitness greater or equal than x, so we must
        %set the flag to 0 after we pick the parent, else the last
        %chromosome of the population will be always picked after
        %the last iteration.
        for k=1:chromosomeSize  %this loop sets the bits of the
            %parent1 variable, according to the bits of the
            %parent chromosome in the population pool
            parent1(k)=populationMatrix(i,j,k);
        end % for j=1:chromosomeSize
        flag=0;
    end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

%the same loop applies for the second parent as well. The
%difference is of course in the variable we use to store
%the second parent.
x=c*rand;
flag=1;
for j=1:populationSize
    if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(i,j,:);  %this is an easier
        %way to set a vector variable. We use the : operator
        %which means "all" (it is used in every "for" loop).
        %Therefore we assign the parent2 vector from the
        %corresponding vector of the population matrix, with
        %just one comand.
    end
end
%notice the fact that we keep the current selected parents,
%in the "Parents' pool", i.e. a chromosome can become a
%parent more than once.

if doCrossover==0   %no crossover at all
    child1=parent1;
    child2=parent2;
else    %doCrossover==1
    %we pick a random cutting point for crossover 1...n
    cuttingPoint=ceil(rand*chromosomeSize);

    %1st child generation.
    %crossover
    for k=1:chromosomeSize
        if k<=cuttingPoint +0.5
            child1(k)=parent1(k);
        else
            child1(k)=parent2(k);
        end %j<=cuttingpoint
    end %for k=1:chromosomeSize

%%We do the same to create the 2nd child,
    for k=1:chromosomeSize
        if k<cuttingPoint+0.5
            child2(k)=parent2(k);
        else
            child2(k)=parent1(k);
        end
    end
```

```
            end %if doCrossover

            %mutation
            for k=1:chromosomeSize
                if rand<probabilityMutation %child is mutated with probability
                    child2(k)=~child2(k);
                end
                if rand<probabilityMutation
                    child1(k)=~child1(k);
                end
            end %for


            %store children in children matrix and calculate their quantities
            p1=fix(2*s-1);  %since a pairing produces 2 children
            p2=2*s;
            q1=0;q2=0;
            for k=1:chromosomeSize
                childrenMatrix(i,p1,k)=child1(k);
                childrenMatrix(i,p2,k)=child2(k);
                q1=q1+childrenMatrix(i,p1,k)*2^(k-1);
                q2=q2+childrenMatrix(i,p2,k)*2^(k-1);
            end
        end %pairing loop for s=1:fix(populationSize/2)
    end %for i=1:numberFirms


    %we update the population matrices, from the offsping matrices, in
    %order to use the next generation as the population in the succesive
    %iteration.
    populationMatrix=childrenMatrix;

    %Markov chain statistics
    d=0;
   for firm=1:numberFirms
        for chrom=1:populationSize
            for k=1:chromosomeSize
                if populationMatrix(firm,chrom,k)~=target(k)
                    d=d+1;
                end
            end
        end
    end
   %update transitionMatrix,fixedVector and currentState
    md=ceil(d/(numberFirms*populationSize));
    if generationNumber>1
        transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
    end
    fixedVector(md+1)=fixedVector(md+1)+1;
    currentState=md;


if mod(generationNumber,100)==0
    generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
for i=1:chromosomeSize+1
    fixedVector(i)=fixedVector(i)/numberIterations;
    for j=1:chromosomeSize+1
```

```
        transitionMatrix(i,j)=transitionMatrix(i,j)/numberIterations;
    end
```

4. Alkemade et al (2007) algorithm

```
%cost function
x_cost=56;
y_cost=0;
numberFirms=20;        %number of firms
%Cost = xq, y=0 always in alkemade et al

%demand curve
a_demand=-1;
b_demand=256;
%price=b-(q_1+q_2+...+q_n) %b=1 always in Alkemade et al (2007)

%genetic algorithm
populationSize= 100; %number of chromosomes in population
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.001; %probability that a childs'bit undergoes mutation
%60 0.5 0.005 -> (4)0.004
numberIterations=100000;   %number of generations
chromosomeSize=6;

%Nash equilibria in games
NashCounter=0;
firstNashTime=-1;

%Markov chain variables
equilibriumTime=50000;
firstNashState=-1;
alphaFixedVector=zeros(1,chromosomeSize+1);
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;

%solutions' space
NashEquilibrium=9.5238; % 82.2143;
maxQuantity=3*NashEquilibrium;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);

%target chromosome = Qmax/3
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end

firmQuantityHistory=zeros(numberIterations*populationSize/numberFirms,numberFirm
s);
averageMarketQuantityHistory=zeros(1,numberIterations*populationSize/numberFirms
);
populationCost=zeros(1,populationSize);
populationProfit=zeros(1,populationSize);
populationQuantity=zeros(1,populationSize);

matches=zeros(populationSize);

%%%%%Initialize population ranomly


    for j=1:populationSize  %this loop runs for all chromosomes in
       %the initial population
        c=0;    %a handy variable
```

```
        for k=1:chromosomeSize  %internal loop that assigns random
           %values 0,1 in each bit. It loops for all bits of the
            %chromosome

            %assingment of random values
            if rand <0.5
                populationMatrix(j,k)=0;
            else
                populationMatrix(j,k)=1;
            end
            %populationProfit(j)=rand; %random initial profits
        end %for k=1:chromosomeSize
    end %for j=1:populationSize

%%%%initialize to opposite state of the equilibrium
%    for j=1:populationSize
%        for k=1:chromosomeSize
%            if mod(k,2)==1
%                populationMatrix(j,k)=0;
%            else
%                populationMatrix(j,k)=1;
%            end
%        end
%    end

%%%main algorithm
for generationNumber=1:numberIterations
    matches=randperm(populationSize);
    for i=1:populationSize/numberFirms
        totalQuantity=0;
        for j=1:numberFirms
            currentIndex=matches((i-1)*numberFirms+j);
            q=0;
            for k=1:chromosomeSize
                q=q+populationMatrix(currentIndex,k)*2^(k-1);
                    %calculate quantity
            end %for j=i:chromosomeSize
            q=q*normalizationFactor; %normalize quantity
            totalQuantity=totalQuantity+q;
            firmQuantity(j)=q;
            firmQuantityHistory((generationNumber-1)*populationSize/numberFirms+
i,j)=q;
            populationQuantity(currentIndex)=q;
            populationCost(currentIndex)=x_cost*q+y_cost;
                %calculate cost
        end %for j=1:numberFirms
        price=a_demand*totalQuantity+b_demand;
        for j=1:numberFirms %calculate profit and fitness
            currentIndex=matches((i-1)*numberFirms+j);
            populationProfit(currentIndex)=price*populationQuantity(currentIndex
)-populationCost(currentIndex);
        end %for i=1:numberFirms
        averageMarketQuantityHistory((generationNumber-1)*populationSize/numberF
irms+i)=mean(firmQuantityHistory((generationNumber-1)*populationSize/numberFirms
+i,:));
        %check if NE
        flag=1;
        for k=1:numberFirms
            if flag && (firmQuantity(k)>= NashEquilibrium-0.00000000001) &&
(firmQuantity(k)<= NashEquilibrium+0.00000000001)
                flag=1;
            else flag=0;
```

```
            end
        end
        if flag==1
            if firstNashTime==-1 %first time
                firstNashTime=(generationNumber-1)*populationSize/numberFirms+i;
            end
            NashCounter=NashCounter+1;
        end
    end %for i=1:populationSize

    %calculate fitness
    [b,ix]=sort(populationProfit); %sort in ascending order, get indices in ix
    for j=1:populationSize
        populationFitness(ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
    end

    for s=1:fix(populationSize/2)    %s is half the population size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

        c=0;     %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
        for j=1:populationSize
            c=c+populationFitness(j);
            cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
         end

         if rand<probabilityCrossover
            doCrossover=1;
         else
            doCrossover=0;
         end     %determine if we are going to do crossover or not

         %picking first parent
         x=c*rand;   %x is a random number between 0 and total cumulative
                %fitness
         flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
         for j=1:populationSize  %loop begins
            if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
                    %the cumulative fitness until chromosome i is greater or
                    %equal than x. In this case we pick chromosome i to be the
                    %first parent. It also checks the value of the flag, since
                    %the succeeding chromosomes (after i) will also have
                    %cumulative fitness greater or equal than x, so we must
                    %set the flag to 0 after we pick the parent, else the last
                    %chromosome of the population will be always picked after
                    %the last iteration.
             for k=1:chromosomeSize  %this loop sets the bits of the
                         %parent1 variable, according to the bits of the
```

```
                %parent chromosome in the population pool
        parent1(k)=populationMatrix(j,k);
     end % for j=1:chromosomeSize
        flag=0;
     end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

   %the same loop applies for the second parent as well. The
   %difference is of course in the variable we use to store
   %the second parent.
x=c*rand;
flag=1;
for j=1:populationSize
   if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(j,:);  %this is an easier
            %way to set a vector variable. We use the : operator
            %which means "all" (it is used in every "for" loop).
            %Therefore we assign the parent2 vector from the
            %corresponding vector of the population matrix, with
            %just one comand.
      end
 end
   %notice the fact that we keep the current selected parents,
   %in the "Parents' pool", i.e. a chromosome can become a
   %parent more than once.

 if doCrossover==0   %no crossover at all
   child1=parent1;
   child2=parent2;
 else    %doCrossover==1
        %we pick a random cutting point for crossover 1...n
    cuttingPoint=ceil(rand*chromosomeSize);

        %1st child generation.
        %crossover
    for k=1:chromosomeSize
        if k<=cuttingPoint +0.5
            child1(k)=parent1(k);
        else
            child1(k)=parent2(k);
        end %j<=cuttingpoint
    end %for k=1:chromosomeSize

    %%We do the same to create the 2nd child,
    for k=1:chromosomeSize
        if k<cuttingPoint+0.5
            child2(k)=parent2(k);
        else
            child2(k)=parent1(k);
        end
    end
end %if doCrossover

%mutation
for k=1:chromosomeSize
   if rand<probabilityMutation %child is mutated with probability
        child2(k)=~child2(k);
   end
   if rand<probabilityMutation
        child1(k)=~child1(k);
   end
```

```
        end %for


        %store children in children matrix and calculate their quantities
        p1=fix(2*s-1);  %since a pairing produces 2 children
        p2=2*s;
        q1=0;q2=0;
        for k=1:chromosomeSize
               childrenMatrix(p1,k)=child1(k);
               childrenMatrix(p2,k)=child2(k);
               q1=q1+childrenMatrix(p1,k)*2^(k-1);
               q2=q2+childrenMatrix(p2,k)*2^(k-1);
        end
  end %pairing loop for s=1:fix(populationSize/2)


   %we update the population matrices, from the offsping matrices, in
   %order to use the next generation as the population in the succesive
   %iteration.
   populationMatrix=childrenMatrix;

   %Markov chain statistics
   d=0;
       for chrom=1:populationSize
           for k=1:chromosomeSize
               if populationMatrix(chrom,k)~=target(k)
                   d=d+1;
               end
           end
       end
  %update transitionMatrix,fixedVector and currentState
   md=ceil(d/(populationSize));
   if generationNumber>1
       transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
   end
   if md==0 && firstNashState==-1%check for first Nash state
       firstNashState=generationNumber;
   end
   fixedVector(md+1)=fixedVector(md+1)+1;
   if generationNumber>equilibriumTime
       alphaFixedVector(md+1)=fixedVector(md+1)+1;
   end
   currentState=md;


if mod(generationNumber,100)==0
    generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
for i=1:chromosomeSize+1
    fixedVector(i)=fixedVector(i)/(numberIterations);
    alphaFixedVector(i)=alphaFixedVector(i)/(numberIterations);
    for j=1:chromosomeSize+1
        transitionMatrix(i,j)=transitionMatrix(i,j)/(numberIterations);
    end
end
```

5. Vriend's multi-population, social learning algorithm

```
x_cost=10;
y_cost=10;
numberFirms=20;        %number of firms
%Cost = xq+y,

%demand curve
a_demand=-1;
b_demand=7.36*10^7+10;
%price=aQ^3+b

%genetic algorithm
populationSize= 20; %number of chromosomes in population
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.001; %probability that a childs'bit undergoes mutation
numberIterations=10000;   %number of generations
chromosomeSize=20;

%Nash equilibria in games
NashCounter=0;
firstNashTime=-1;
%epoch
GARate=20;
%Markov chain variables
equilibriumTime=50000;
firstNashState=-1;
alphaFixedVector=zeros(1,chromosomeSize+1);
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;

%solutions' space
NashEquilibrium=20; % 82.2143;
maxQuantity=3*NashEquilibrium;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);

%target chromosome = Qmax/3
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end

firmQuantityHistory=zeros(numberIterations,numberFirms);
averageMarketQuantityHistory=zeros(1,numberIterations);
populationMatrix=zeros(numberFirms*populationSize,chromosomeSize);
%%%%%Initialize populations

for i=1:numberFirms
    for j=1:populationSize  %this loop runs for all chromosomes in
        %the initial population
        c=0;    %a handy variable
        for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

            %assingment of random values
            if rand <0.5
                firmPopulation(i,j,k)=0;
            else
                firmPopulation(i,j,k)=1;
            end
```

```
            populationProfit((i-1)*populationSize+j)=rand; %random initial
profits
        end %for j=1:chromosomeSize
    end %for j=1:populationSize
end %for i=1:numberFirms

%%%% main algorithm
for generationNumber=1:numberIterations % Now the main algorithm starts,
    % and it executes for the given time of
    %iterations, stored in numberIterations

    %calculation of the total quantity, the cost and the price.
    totalQuantity=0;
    for i=1:numberFirms
        j=ceil(populationSize*rand);
        currentIndex(i)=j;
        q=0;
        for k=1:chromosomeSize
            q=q+firmPopulation(i,j,k)*2^(k-1);
                %calculate quantity
        end %for j=i:chromosomeSize
        q=q*normalizationFactor; %normalize quantity
        totalQuantity=totalQuantity+q;
        firmQuantityHistory(generationNumber,i)=q;
        firmQuantity(i)=q;
        populationQuantity(i,j)=q;
        populationCost(i,j)=x_cost*q+y_cost;
            %calculate cost
    end %i=1:numberFirms
    price= a_demand*totalQuantity^3+b_demand; %calculate price
    for i=1:numberFirms %calculate profit and fitness
       populationProfit((i-1)*populationSize+currentIndex(i))=price*populationQu
antity(i,currentIndex(i))-populationCost(i,currentIndex(i));
    end %for i=1:numberFirms
    averageMarketQuantityHistory(generationNumber)=mean(firmQuantityHistory(gene
rationNumber,:));
    %priceHistory(generationNumber)=price;

    %check if NE
        flag=1;
        for k=1:numberFirms
            if flag && (firmQuantity(k)>= NashEquilibrium-0.00000000001) &&
(firmQuantity(k)<= NashEquilibrium+0.00000000001)
                flag=1;
            else flag=0;
            end
        end
        if flag==1
            if firstNashTime==-1 %first time
                firstNashTime=(generationNumber-1)*populationSize/numberFirms+i;
            end
            NashCounter=NashCounter+1;
        end


if mod(generationNumber,GARate)==0
    for i=1:numberFirms %copy chroms to big pop
        for j=1:populationSize
            populationMatrix((i-1)*populationSize+j,:)=firmPopulation(i,j,:);
        end
    end
```

```
    %calculate fitness
    [b,ix]=sort(populationProfit); %sort in ascending order, get indices in ix
    for j=1:populationSize*numberFirms
        populationFitness(ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
    end

        for s=1:fix(populationSize*numberFirms/2)   %s is half the population
size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

        c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
        for j=1:populationSize*numberFirms
            c=c+populationFitness(j);
            cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
         end

         if rand<probabilityCrossover
            doCrossover=1;
         else
            doCrossover=0;
         end     %determine if we are going to do crossover or not

         %picking first parent
         x=c*rand;    %x is a random number between 0 and total cumulative
                %fitness
         flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
         for j=1:populationSize*numberFirms  %loop begins
            if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
                    %the cumulative fitness until chromosome i is greater or
                    %equal than x. In this case we pick chromosome i to be the
                    %first parent. It also checks the value of the flag, since
                    %the succeeding chromosomes (after i) will also have
                    %cumulative fitness greater or equal than x, so we must
                    %set the flag to 0 after we pick the parent, else the last
                    %chromosome of the population will be always picked after
                    %the last iteration.
             for k=1:chromosomeSize  %this loop sets the bits of the
                       %parent1 variable, according to the bits of the
                       %parent chromosome in the population pool
                parent1(k)=populationMatrix(j,k);
             end % for j=1:chromosomeSize
                flag=0;
             end %if ( x<cumulativeFitness(i) && flag==1 )
         end %for j=1:populationSize

            %the same loop applies for the second parent as well. The
            %difference is of course in the variable we use to store
```

```
   %the second parent.
x=c*rand;
flag=1;
for j=1:populationSize*numberFirms
   if (x<cumulativeFitness(j) && flag==1)
       flag=0;
       parent2=populationMatrix(j,:);  %this is an easier
           %way to set a vector variable. We use the : operator
           %which means "all" (it is used in every "for" loop).
           %Therefore we assign the parent2 vector from the
           %corresponding vector of the population matrix, with
           %just one comand.
     end
 end
   %notice the fact that we keep the current selected parents,
   %in the "Parents' pool", i.e. a chromosome can become a
   %parent more than once.

 if doCrossover==0   %no crossover at all
   child1=parent1;
   child2=parent2;
 else    %doCrossover==1
       %we pick a random cutting point for crossover 1...n
    cuttingPoint=ceil(rand*chromosomeSize);

       %1st child generation.
       %crossover
    for k=1:chromosomeSize
       if k<=cuttingPoint +0.5
           child1(k)=parent1(k);
       else
           child1(k)=parent2(k);
       end %j<=cuttingpoint
    end %for k=1:chromosomeSize

    %%We do the same to create the 2nd child,
    for k=1:chromosomeSize
       if k<cuttingPoint+0.5
           child2(k)=parent2(k);
       else
           child2(k)=parent1(k);
       end
    end
end %if doCrossover

%mutation
for k=1:chromosomeSize
   if rand<probabilityMutation %child is mutated with probability
       child2(k)=~child2(k);
   end
   if rand<probabilityMutation
       child1(k)=~child1(k);
   end
end %for


%store children in children matrix and calculate their quantities
p1=fix(2*s-1);  %since a pairing produces 2 children
p2=2*s;
q1=0;q2=0;
for k=1:chromosomeSize
       childrenMatrix(p1,k)=child1(k);
```

```
                childrenMatrix(p2,k)=child2(k);
                q1=q1+childrenMatrix(p1,k)*2^(k-1);
                q2=q2+childrenMatrix(p2,k)*2^(k-1);
         end
   end %pairing loop for s=1:fix(populationSize/2)

    %we update the population matrices, from the offsping matrices, in
    %order to use the next generation as the population in the succesive
    %iteration.
    populationMatrix=childrenMatrix;

    %Markov chain statistics
    d=0;
        for chrom=1:populationSize
            for k=1:chromosomeSize
                if populationMatrix(chrom,k)~=target(k)
                    d=d+1;
                end
            end
        end
   %update transitionMatrix,fixedVector and currentState
    md=ceil(d/(numberFirms*populationSize));
    if generationNumber/GARate>1
        transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
    end
    if md==0 && firstNashState==-1%check for first Nash state
        firstNashState=generationNumber;
    end
    fixedVector(md+1)=fixedVector(md+1)+1;
    if generationNumber>equilibriumTime
        alphaFixedVector(md+1)=fixedVector(md+1)+1;
    end
    currentState=md;

    %update firms' populations
    for i=1:numberFirms
        for j=1:populationSize
            firmPopulation(i,j,:)=populationMatrix((i-1)*populationSize+j,:);
        end
    end

end %if GARate


if mod(generationNumber,100)==0
    generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
for i=1:chromosomeSize+1
    fixedVector(i)=fixedVector(i)*GARate/(numberIterations);
    alphaFixedVector(i)=alphaFixedVector(i)*GARate/(numberIterations);
    for j=1:chromosomeSize+1
        transitionMatrix(i,j)=transitionMatrix(i,j)*GARate/(numberIterations);
    end
end
```

6. Coevolutionary programming, social learning version

```
%cost function
```

```
x_cost=56;
y_cost=0;
numberFirms=4;        %number of firms
%Cost = xq, y=0 always in alkemade et al

%demand curve
a_demand=-1;
b_demand=256;
%price=a-(q_1+q_2+...+q_n) %b=1 always in Alkemade et al (2007)

%genetic algorithm
populationSize= 40; %number of chromosomes in population
probabilityCrossover=1; %probability that we have crossover for a pair?
probabilityMutation = 0.0005; %probability that a childs'bit undergoes mutation
numberIterations=10000;   %number of generations
chromosomeSize=20;

%Nash equilibria in games
NashCounter=0;
firstNashTime=-1;

%Markov chain variables
fixedVector=zeros(1,chromosomeSize+1);
transitionMatrix=zeros (chromosomeSize +1, chromosomeSize +1);
currentState=-1;
firstNashState=-1;

%solutions' space
NashEquilibrium=40;%19.3749; % 82.2143;
maxQuantity=3*NashEquilibrium;
normalizationFactor=maxQuantity/(2^chromosomeSize-1);

%target chromosome = Qmax/3
for i=1:chromosomeSize/2
    target(2*i)=0;
    target(2*i-1)=1;
end

firmQuantityHistory=zeros(numberIterations*populationSize,numberFirms);
averageMarketQuantityHistory=zeros(1,numberIterations*populationSize);
%populationCost=zeros(numberFirms,populationSize);
%populationProfit=zeros(numberFirms,populationSize);
%populationQuantity=zeros(numberFirms,populationSize);

matches=zeros(numberFirms,populationSize);

%%%%%Initialize populations

for i=1:numberFirms
    for j=1:populationSize  %this loop runs for all chromosomes in
        %the initial population
        c=0;    %a handy variable
        for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

           %assingment of random values
            %if rand <0.5
            %    firmPopulation(i,j,k)=0;
            %else
            %    firmPopulation(i,j,k)=1;
            %end
```

```
            %assingment to opposite nash state
            if mod(k,2)==1
                firmPopulation(i,j,k)=0;
            else
                firmPopulation(i,j,k)=1;
            end
        end %for j=1:chromosomeSize
    end %for j=1:populationSize
end %for i=1:numberFirms

%%%% main algorithm
for generationNumber=1:numberIterations % Now the main algorithm starts,
    % and it executes for the given time of
    %iterations, stored in numberIterations

    %assingments
    totalQuantity=0;
    for i=1:numberFirms
        matches(i,:)=randperm(populationSize);
    end
    for j=1:populationSize
        totalQuantity=0;
        for i=1:numberFirms
            currentIndex=matches(i,j);
            q=0;
            for k=1:chromosomeSize
                q=q+firmPopulation(i,currentIndex,k)*2^(k-1);
                    %calculate quantity
            end %for j=i:chromosomeSize
            q=q*normalizationFactor; %normalize quantity
            totalQuantity=totalQuantity+q;
            firmQuantity(i)=q;
            firmQuantityHistory((generationNumber-1)*populationSize+j,i)=q;
            populationQuantity(i,currentIndex)=q;
            populationCost(i,currentIndex)=x_cost*q+y_cost;
                %calculate cost
        end %i=1:numberFirms
        price= a_demand*totalQuantity+b_demand; %calculate price
        for i=1:numberFirms %calculate profit and fitness
            currentIndex=matches(i,j);
            populationProfit((i-1)*populationSize+currentIndex)=price*population
Quantity(i,currentIndex)-populationCost(i,currentIndex);
        end %for i=1:numberFirms
        averageMarketQuantityHistory((generationNumber-1)*populationSize+j)=mean
(firmQuantityHistory((generationNumber-1)*populationSize+j,:));
        %check if NE
        %check if NE
        flag=1;
        for k=1:numberFirms
            if flag && (firmQuantity(k)>= NashEquilibrium-0.00000000001) &&
(firmQuantity(k)<= NashEquilibrium+0.00000000001)
                flag=1;
            else flag=0;
            end
        end
        if flag==1
            if firstNashTime==-1 %first time
                firstNashTime=(generationNumber-1)*populationSize/numberFirms+i;
            end
            NashCounter=NashCounter+1;
        end
```

```
    %priceHistory(generationNumber)=price;
    end %for j=1:populationSize

    for i=1:numberFirms %copy chroms to big pop
        for j=1:populationSize
            populationMatrix((i-1)*populationSize+j,:)=firmPopulation(i,j,:);
        end
    end

    [b,ix]=sort(populationProfit); %sort in ascending order, get indices in ix
    for j=1:populationSize*numberFirms
        populationFitness(ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
    end

          for s=1:fix(populationSize*numberFirms/2)   %s is half the population
size (I used
            %fix to ensure it gets an integer value). since every pairing
            %produces 2 children and we want to have populationSize children,
            %the number of pairings should be half the number of chromosomes
            %in the population.

        c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
        for j=1:populationSize*numberFirms
            c=c+populationFitness(j);
            cumulativeFitness(j)=c; %so we add the current's chromosome
                %fitness to c at every iteration. We also store it in a vector
                %at every iteration, so the vector holds the values of the
                %cumulative fitness until the given chromosome of the
                %population. c also holds the value of the total cumulative
                %fitness after the last iteration.
         end

         if rand<probabilityCrossover
            doCrossover=1;
         else
            doCrossover=0;
         end     %determine if we are going to do crossover or not

         %picking first parent
         x=c*rand;   %x is a random number between 0 and total cumulative
                %fitness
         flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
         for j=1:populationSize*numberFirms  %loop begins
            if ( x<cumulativeFitness(j) && flag==1 ) %checking whether
                    %the cumulative fitness until chromosome i is greater or
                    %equal than x. In this case we pick chromosome i to be the
                    %first parent. It also checks the value of the flag, since
                    %the succeeding chromosomes (after i) will also have
                    %cumulative fitness greater or equal than x, so we must
                    %set the flag to 0 after we pick the parent, else the last
                    %chromosome of the population will be always picked after
                    %the last iteration.
              for k=1:chromosomeSize  %this loop sets the bits of the
                        %parent1 variable, according to the bits of the
                        %parent chromosome in the population pool
```

```
        parent1(k)=populationMatrix(j,k);
     end % for j=1:chromosomeSize
        flag=0;
     end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

   %the same loop applies for the second parent as well. The
   %difference is of course in the variable we use to store
   %the second parent.
x=c*rand;
flag=1;
for j=1:populationSize*numberFirms
   if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(j,:);  %this is an easier
            %way to set a vector variable. We use the : operator
            %which means "all" (it is used in every "for" loop).
            %Therefore we assign the parent2 vector from the
            %corresponding vector of the population matrix, with
            %just one comand.
      end
 end
   %notice the fact that we keep the current selected parents,
   %in the "Parents' pool", i.e. a chromosome can become a
   %parent more than once.

 if doCrossover==0   %no crossover at all
   child1=parent1;
   child2=parent2;
 else    %doCrossover==1
        %we pick a random cutting point for crossover 1...n
     cuttingPoint=ceil(rand*chromosomeSize);

        %1st child generation.
        %crossover
     for k=1:chromosomeSize
        if k<=cuttingPoint +0.5
            child1(k)=parent1(k);
        else
            child1(k)=parent2(k);
        end %j<=cuttingpoint
     end %for k=1:chromosomeSize

     %%We do the same to create the 2nd child,
     for k=1:chromosomeSize
        if k<cuttingPoint+0.5
            child2(k)=parent2(k);
        else
            child2(k)=parent1(k);
        end
     end
end %if doCrossover

%mutation
for k=1:chromosomeSize
   if rand<probabilityMutation %child is mutated with probability
        child2(k)=~child2(k);
   end
   if rand<probabilityMutation
        child1(k)=~child1(k);
   end
end %for
```

```
        %store children in children matrix and calculate their quantities
        p1=fix(2*s-1);  %since a pairing produces 2 children
        p2=2*s;
        q1=0;q2=0;
        for k=1:chromosomeSize
               childrenMatrix(p1,k)=child1(k);
               childrenMatrix(p2,k)=child2(k);
               q1=q1+childrenMatrix(p1,k)*2^(k-1);
               q2=q2+childrenMatrix(p2,k)*2^(k-1);
        end
  end %pairing loop for s=1:fix(populationSize/2)


   %we update the population matrices, from the offsping matrices, in
   %order to use the next generation as the population in the succesive
   %iteration.
   populationMatrix=childrenMatrix;

   %Markov chain statistics
   d=0;
       for chrom=1:populationSize*numberFirms
           for k=1:chromosomeSize
               if populationMatrix(chrom,k)~=target(k)
                   d=d+1;
               end
           end
       end
  %update transitionMatrix,fixedVector and currentState
   md=ceil(d/(numberFirms*populationSize));
   if generationNumber>1
       transitionMatrix(currentState+1,md+1)=transitionMatrix(currentState+1,md
+1)+1;
   end
   if md==0 && firstNashState==-1%check for first Nash state
       firstNashState=generationNumber;
       telos

   end
   fixedVector(md+1)=fixedVector(md+1)+1;
   currentState=md;

   %update firms' populations
   for i=1:numberFirms
       for j=1:populationSize
           firmPopulation(i,j,:)=populationMatrix((i-1)*populationSize+j,:);
       end
   end

if mod(generationNumber,100)==0
   generationNumber
end
end %for generationNumber=1:numberIterations. end of main algorithm loop

%normalize transition matrix and fixed vector to frequencies
for i=1:chromosomeSize+1
   fixedVector(i)=fixedVector(i)/numberIterations;
   for j=1:chromosomeSize+1
       transitionMatrix(i,j)=transitionMatrix(i,j)/numberIterations;
   end
```

```
end
```

7.Optimization Genetic Algorithm for alternative best-replies

```
currentState=-1;
firstNashState=-1;

%quantity transformation
minQuantity=[0 0];
maxQuantity=[7.5 7.5];
normalizationFactor=(maxQuantity-minQuantity)/(2^chromosomeSize-1);

firmQuantityHistory=zeros(numberFirms,numberIterations);
%initialize quantities
for i=1:numberFirms
    firmQuantity(i)=minQuantity(i)+rand*(maxQuantity(i)-minQuantity(i));
end

%main algorithm
for iteration=1:numberIterations
    for firmIndex=1:numberFirms

        %calculate total quantity of opponents
        otherTotalQuantity=0;
        for i=1:numberFirms
            if i~=firmIndex
                otherTotalQuantity=otherTotalQuantity+firmQuantity(i);
            end
        end

        %initialize population
        for j=1:populationSize  %this loop runs for all chromosomes in
            %the initial population
            c=0;    %a handy variable
            for k=1:chromosomeSize  %internal loop that assigns random
            %values 0,1 in each bit. It loops for all bits of the
            %chromosome

           %assingment of random values
                if rand <0.5
                    populationMatrix(j,k)=0;
                else
                    populationMatrix(j,k)=1;
                end

            %assingment to opposite nash state
              %  if mod(k,2)==1
               %     Population(i,j)=0;
                %else
                 %   Population(i,j)=1;
                %end
                end %for j=1:chromosomeSize
         end %for j=1:populationSize

        %genetic algorithm
        for generation=1:numberGenerations
            bestQuantitySoFar=-inf;
            bestProfitSoFar=-inf;

            for j=1:populationSize
                q=0;
                for k=1:chromosomeSize
                    q=q+populationMatrix(j,k)*2^(k-1);
```

```
                %calculate quantity
            end %for j=i:chromosomeSize
            q=q*normalizationFactor(firmIndex); %normalize quantity
            populationQuantity(j)=q+minQuantity(firmIndex);
            %calculate profit and check if bestSoFar
            %populationProfit(j)=(a_demand*(otherTotalQuantity+populati
            %onQuantity(j))+b_demand)*populationQuantity(j)-
(x_cost*populationQuantity(j)+y_cost*populationQuantity(j)^2);
            populationProfit(j)=21+populationQuantity(j)*sin(pi*populationQu
antity(j))+populationQuantity(j)*otherTotalQuantity*sin(pi*otherTotalQuantity);
            if populationProfit(j)>bestProfitSoFar
                bestProfitSoFar=populationProfit(j);
                bestQuantitySoFar=populationQuantity(j);
            end
        end %for j=1:populationSize
        %calculate ordered fitness
        [b,ix]=sort(populationProfit); %sort in ascending order, get indices
in ix
        for j=1:populationSize
            populationFitness(ix(j))=j;
            %chrom with min profit gets fitness 1, next
            %gets 2, etc.
        end

        %calculate scaled fitness
        %minP=min(populationProfit);
        %for j=1:populationSize
         %   populationFitness(j)=populationProfit(j)-minP;
        %end

        %make new Generation
        for s=1:fix(populationSize/2)

            c=0;    %initialization of the handy variable c
            %the following loop calculates the cumulative fitness of all the
            %chromosomes in the population
            for j=1:populationSize
                c=c+populationFitness(j);
                cumulativeFitness(j)=c;
            end

            if rand<probabilityCrossover
                doCrossover=1;
            else
                doCrossover=0;
            end     %determine if we are going to do crossover or not

            %picking first parent
            x=c*rand;   %x is a random number between 0 and total cumulative
                %fitness
            flag=1; %this flag is used to indicate if we are still searching
                %for the parent. If not, it is set to 0 and the loop remains
                %idle in the following iterations. I used this because there
                %were some problems when I used a "break" statement
            for j=1:populationSize  %loop begins
                if ( x<cumulativeFitness(j) && flag==1 )
                    for k=1:chromosomeSize  %this loop sets the bits of the
                        %parent1 variable, according to the bits of the
                        %parent chromosome in the population pool
                        parent1(k)=populationMatrix(j,k);
                    end % for j=1:chromosomeSize
                    flag=0;
```

```
    end %if ( x<cumulativeFitness(i) && flag==1 )
end %for j=1:populationSize

%the same loop applies for the second parent as well. The
%difference is of course in the variable we use to store
%the second parent.
x=c*rand;
flag=1;
for j=1:populationSize
    if (x<cumulativeFitness(j) && flag==1)
        flag=0;
        parent2=populationMatrix(j,:);  %this is an easier
        %way to set a vector variable. We use the : operator
        %which means "all" (it is used in every "for" loop).
        %Therefore we assign the parent2 vector from the
        %corresponding vector of the population matrix, with
        %just one comand.
    end
end
%notice the fact that we keep the current selected parents,
%in the "Parents' pool", i.e. a chromosome can become a
%parent more than once.

if doCrossover==0   %no crossover at all
    child1=parent1;
    child2=parent2;
else    %doCrossover==1
    %we pick a random cutting point for crossover 1...n
    cuttingPoint=ceil(rand*chromosomeSize);

    %1st child generation.
    %crossover
    for k=1:chromosomeSize
        if k<=cuttingPoint +0.5
            child1(k)=parent1(k);
        else
            child1(k)=parent2(k);
        end %j<=cuttingpoint
    end %for k=1:chromosomeSize

%%We do the same to create the 2nd child,
    for k=1:chromosomeSize
        if k<cuttingPoint+0.5
            child2(k)=parent2(k);
        else
            child2(k)=parent1(k);
        end
    end
end %if doCrossover

%mutation
for k=1:chromosomeSize
    if rand<probabilityMutation %child is mutated with
probability

        child2(k)=~child2(k);
    end
    if rand<probabilityMutation
        child1(k)=~child1(k);
    end
end %for
```

```
                %store children in children matrix and calculate their
quantities
                p1=fix(2*s-1);  %since a pairing produces 2 children
                p2=2*s;
                q1=0;q2=0;
                for k=1:chromosomeSize
                    childrenMatrix(p1,k)=child1(k);
                    childrenMatrix(p2,k)=child2(k);
                end
            end %pairing loop for s=1:fix(populationSize/2)
            populationMatrix=childrenMatrix; %update population matrix
        end %generation
        %update quantity choises
        firmQuantity(firmIndex)=bestQuantitySoFar;
        firmQuantityHistory(firmIndex,iteration)=bestQuantitySoFar;
    end  %for firmIndex=1:numberFirms


    iteration   %to know where we are
end %algorithm
```