Τ.Ε.Ι. Δυτικής Ελλάδας
Τμήμα Μηχανικών Πληροφορικής Τ.Ε

ΤΙΤΛΟΣ: ΒΕΛΤΙΣΤΗ ΚΑΤΑΝΟΜΗ ΕΝΟΣ ΔΙΚΤΥΟΥ ΜΕ ΧΡΗΣΗ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Αιγινήτου Ευδοκία Ζωή ΑΜ: 1148

Επιβλέπων Καθηγητής : Ιωάννης Κούγιας

Αντίρριο, ημερομηνία

ΕΥΧΑΡΙΣΤΙΕΣ

Πριν τη παρουσίαση της διπλωματικής μου, αισθάνομαι την ανάγκη να ευχαριστήσω ορισμένους ανθρώπους που συνεργάστηκα και έπαιξαν βασικό ρόλο κατά τη διάρκεια εκπόνησης της εργασίας μου. Αρχικά, θα ήθελα να ευχαριστήσω το καθηγητή μου Ιωάννη Κούγια για την υποστήριξη και τη άμεση βοήθεια που μου πρόσφερε όλο αυτό τον καιρό. Στη συνέχεια θα ήθελα να ευχαριστήσω όσους με βοήθησαν είτε πρακτικά είτε ψυχολογικά. Ιδιαίτερη ήταν η συμβολή της οικογένειας μου που χωρίς αυτούς δεν θα είχα καταφέρει να φτάσω ως εδώ. Επίσης θα ήθελα να ευχαριστήσω τους φίλους μου Γιώργο, Χρήστο ,Αντρέα Κρυσταλλία και Νεκταρία.

Αιγινήτου Ευδοκία- Ζωή

Μάιος 2016

<u>**ΠΕΡΙΕΧΟΜΕΝΑ**</u>

ΣΥΝΟΨΗ

Η παρούσα διπλωματική εργασία εκπονήθηκε στο Τμήμα Τηλεπικοινωνιακών Συστημάτων και Δικτύων της σχολής Μηχανικών Πληροφορικής Δυτικής Ελλάδος.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η προσέγγιση των μαθηματικών προβλημάτων των δικτύων και των γενετικών αλγόριθμων, όπως το πρόβλημα του πλανόδιου πωλητή (TSP) και το disc scheduling problem (DSP). Υπολογιστική εφαρμογή πραγματοποιείται σε προγραμματιστικό περιβάλλον Matlab με χρήση κώδικα γενετικών αλγορίθμων.

Λέξεις κλειδιά: Πρόβλημα πλανόδιου πωλητή (TSP) , γενετικοί αλγόριθμοι, πρόβλημα του προγραμματισμού της βέλτιστης σειράς ανάγνωσης δεδομένων (DSP).

ABSTRACT

This thesis is based on the scientific area of mathematical problems in networks. It selects a well-recognized problem, the Travelling Salesman Problem (TSP) and implements a genetic algorithm to solve it. The traveling salesman problem and a set of its variations are presented in Chapter 1. Chapter 2 presents the theory of genetic algorithms. In Chapter 3, the symmetric TSP is solved with the use of genetic algorithms in the Matlab programming environment. The optimal tour covering the fifty (50) largest population cities in Greece is found. This application's benefit is that by visualizing the cities on a Cartesian system, the reliability and the computational behavior of the genetic algorithms code is verified and examined. Chapter 4 presents the problem of scheduling the optimal order of data reading from a hard disk (Disk Scheduling Problem, DSP). This problem is classified as an instance of the non-symmetric TSP. More specifically this chapter illustrates how a typical contemporary hard drive physically operates and the corresponding mathematical relationship are developed so as to establish the relationship between the DSP and the TSP. Chapter 5 presents a numerical application of the DSP. Finally, summarizes the main conclusions of this study.

Key words: Travel salesman problem (TSP), Genetic algorithms, Disc scheduling problem (DSP)

ΕΙΣΑΓΩΓΗ

Η παρούσα διπλωματική εκκινεί από το επιστημονικό πεδίο των μαθηματικών προβλημάτων σε δίκτυα, επιλέγει ένα ιδιαίτερα αναγνωρισμένο πρόβλημα, το πρόβλημα του περιοδεύοντος πωλητή (Travelling Salesman Problem, TSP) και υλοποιεί ένα γενετικό αλγόριθμο επίλυσής του. Το πρόβλημα του περιοδεύοντος πωλητή μαζί με ένα σύνολο παραλλαγών του παρουσιάζονται στο Κεφάλαιο 1. Στο Κεφάλαιο 2 παρουσιάζεται η θεωρία των γενετικών αλγορίθμων. Στο Κεφάλαιο 3 υλοποιείται με την βοήθεια των γενετικών αλγορίθμων σε προγραμματιστικό περιβάλλον Matlab μία εφαρμογή του συμμετρικού προβλήματος του περιοδεύοντος πωλητή με αντικείμενο την εύρεση του βέλτιστου κυκλικού μονοπατιού που καλύπτει τις πενήντα (50) μεγαλύτερες πληθυσμιακά πόλεις της Ελλάδας. Ο λόγος είναι ότι με τη βοήθεια της απεικόνισης των πόλεων σε καρτεσιανό σύστημα είναι δυνατό να ελεγχθεί η αξιοπιστία και η υπολογιστική συμπεριφορά του χρησιμοποιούμενου κώδικα γενετικών αλγορίθμων. Στο Κεφάλαιο 4 παρουσιάζονται το πρόβλημα του προγραμματισμού της βέλτιστης σειράς ανάγνωσης δεδομένων από σκληρό δίσκο (Disk Scheduling Problem, DSP), το οποίο εντάσσεται στην ειδική τάξη του μη συμμετρικού προβλήματος του περιοδεύοντος πωλητή. Ειδικότερα περιγράφεται ο τρόπος λειτουργίας ενός τυπικού σύγχρονου σκληρού δίσκου και αναπτύσσονται οι αντίστοιχες μαθηματικές σχέσεις ώστε να τεκμηριώνεται η σχέση του DSP με το TSP. Στο Κεφάλαιο 5 υλοποιείται μία αριθμητική εφαρμογή του DSP. Τέλος συνοψίζονται τα κύρια συμπεράσματα της παρούσας εργασίας.

ΚΕΦΑΛΑΙΟ 1: ΠΡΟΒΛΗΜΑ ΠΛΑΝΟΔΙΟΥ ΠΩΛΗΤΗ (TSP)

1.1 ΔΙΑΤΥΠΩΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ένας πλανόδιος πωλητής έχει ως στόχο να κάνει μια περιοδεία από όλες τις πόλεις που του δίνονται μια φορά. Για να επιτευχτεί αυτό, ξεκινά από μία πόλη του και στο τέλος πρέπει να επιστρέψει στη πόλη από την οποία ξεκίνησε. Ο πωλητής μπορεί να διαλέξει τη σειρά με την οποία θα επισκεφτεί τις πόλεις έτσι ώστε η συνολική απόσταση που θα διανύσει να είναι η μικρότερη.

1.2 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Οι απαρχές του TSP δεν είναι σαφείς. Ένα εγχειρίδιο για πλανόδιους πωλητές το 1832 αναφέρει το πρόβλημα και περιλαμβάνει παραδείγματα περιηγήσεων μέσω της Γερμανίας και της Ελβετίας, αλλά δεν περιέχει καμία μαθηματική προσέγγιση. Τον 19ο αιώνα το TSP ορίστηκε από τον Ιρλανδό μαθηματικό William Rowan Hamilton και τον Βρετανό μαθηματικό Thomas Kirkman. Ο Hamilton δημιούργησε το Icosian Puzzle (το όνομα είναι από το αρχαίο ελληνικό είκοσι), που περιλαμβάνει την εύρεση ενός κύκλου από τις άκρες ενός δωδεκάεδρου, έτσι ώστε κάθε κόμβος να επισκέπτεται μία μόνο φορά, κανένας κόμβος να μην επισκέπτεται δεύτερη φορά και το τελικό σημείο να είναι ίδιο με το αρχικό. Ο κύκλος αυτός ονομάστηκε κύκλος του Hamilton ο οποίος είναι ένα μονοπάτι με την ιδιότητα αυτή. Οπότε ένα μονοπάτι του Hamilton είναι ένα μονοπάτι σε μη κατευθυνόμενο γράφημα το οποίο επισκέπτεται κάθε κόμβο ακριβώς μια φορά.

Η γενική μορφή του TSP φαίνεται να έχει μελετηθεί για πρώτη φορά κατά την δεκαετία του 1930 στην Βιέννη και στο Χάρβαρντ κυρίως από τον Karl Menger ο οποίος καθόρισε και το πρόβλημα. Ο Hasslier Whitney στο Πανεπιστήμιο του Princeton εισήγαγε το όνομα TSP αμέσως μετά. Στην δεκαετία του 1950 και 1960, έγινε όλο και πιο δημοφιλές στους επιστημονικούς κύκλους της Ευρώπηςκαι της Αμερικής. Αξιοσημείωτη ήταν η συμβολή των George Dantzig, Delbert Ray Fulkerson και Selmer Johnson στο συνέδριο RAND στην Santa Monica, οι οποίοι εξέφρασαν το πρόβλημα ως ένα πρόβλημα ακέραιου γραμμικού προγραμματισμού. Το TSP έχει ένα χαρακτήρα πρότυπο για πολλούς κλάδους των Μαθηματικών, των Υπολογιστών και της Επιχειρησιακής Έρευνας. Οι κύριες συνιστώσες για τις σημερινές πιο επιτυχείς προσεγγίσεις για δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης είναι οι αλγόριθμοι Ευριτικής Αναζήτησης, ο Γραμμικός Προγραμματισμός και οι αλγόριθμοι Branch-and-Bound1 (B&B), οι οποίες αρχικά διατυπώθηκαν για το TSP και συνήθιζαν να επιλύουν περιπτώσεις πρακτικών προβλημάτων οι Dantzig, Fulkerson και Johnson.

Στις επόμενες δεκαετίες το πρόβλημα μελετήθηκε από πολλούς ερευνητές διαφόρων κλάδων. Ο Richard Karp, έδειξε το 1972, ότι το πρόβλημα με τον κύκλο του Hamilton ήταν ένα NP2-πλήρες πρόβλημα (δηλαδή μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο), από το οποίο προκύπτει για το TSP ότι ανήκει στα NP-hard προβλήματα λόγω της πολυπλοκότητας του. Αυτό παρέχει μια μαθηματική εξήγηση για την προφανή υπολογιστική δυσκολία εύρεσης βέλτιστων περιηγήσεων.Αρχικά αναπτύχθηκαν νέες αλγοριθμικές τεχνικές κα εφαρμόστηκαν στο TSP για να αποδείξουν την αποτελεσματικότητα τους. Παραδείγματα τέτοιων τεχνικών είναι η μέθοδος B&B, η μέθοδος χαλάρωσης του Lagrange3, ο αλγόριθμος και η ευριστίκη συνάρτηση των Lin-Kernighan, η προσομοιωμένη ανάκτηση και το πεδίο των συνδυαστικών πολύεδρων για δύσκολα συνδυαστικά προβλήματα βελτιστοποίησης.

Μεγάλη πρόοδος επιτεύχθηκε στα τέλη των δεκαετιών 1970 και 1980, όταν οι Grotschel, Padberg, Rinaldi και άλλοι, κατάφεραν να επιλύσουν ακριβώς περιπτώσεις με έως και 2392 πόλεις, χρησιμοποιώντας την μέθοδο των cutting-planes4 και B&B. Κατά την δεκαετία του 1990, οι Applegate, Bixby, Chvatal και Cook ανάπτυξαν το πρόγραμμα Concorde που έχει χρησιμοποιηθεί σε πολλές πρόσφατες καταγεγραμμένες λύσεις. Ο Gerhard Reinelt το 1991 δημοσίευσε το TSPLIB, το οποίο είναι μια συλλογή από συγκριτικές αξιολογήσεις περιπτώσεων διαφορετικής δυσκολίας και έχει χρησιμοποιηθεί επίσης από πολλούς ερευνητές για σύγκριση των αποτελεσμάτων. Το 2005, ο Cook και κάποιοι άλλοι, υπολόγισαν μια περίπτωση βέλτιστης περιήγησης διαμέσου 33810 πόλεων, η οποία δόθηκε από ένα πρόβλημα διάταξης ενός μικροτσίπ, επί του παρόντος η μεγαλύτερη λυμένη περίπτωση. Για πολλές άλλες περιπτώσεις με εκατομμύρια πόλεις, οι λύσεις που μπορούν να βρεθούν εγγυούνται μόνο σε ποσοστό 1% να είναι μια βέλτιστη περιήγηση. Ένας από τους λόγους που έκαναν το TSP ένα τόσο δημοφιλές πρόβλημα ήταν η στενή σχέση του με θέματα των συνδυαστικών προβλημάτων που προέκυπταν τότε από την νέα μέθοδο του γραμμικού προγραμματισμού, ιδίως του προβλήματος εκχώρησης και, πιο γενικά, του προβλήματος μεταφοράς που αναφέραμε στο προηγούμενο κεφάλαιο. Το TSP ήταν σαν όλα αυτά τα προβλήματα, αλλά προφανώς πιο δύσκολο στο να λυθεί, και η πρόκληση έγινε ακόμα πιο ενδιαφέρουσα. Και,φυσικά, το TSP έγινε τόσο δημοφιλές επειδή έχει ένα όνομα που υπενθύμιζε και υπενθυμίζει σε όλους άλλα πράγματα.

1.3 ΠΑΡΑΛΛΑΓΕΣ TSP

Εκτός από το κλασσικό πρόβλημα του Πλανόδιου Πωλητή που αναλύσαμε προηγουμένως ,υπάρχουν και κάποιες προεκτάσεις/είδη του συγκεκριμένου προβλήματος οι οποίες θα αναφερθούν παρακάτω.

1.3.1 Ασύμμετρο TSP

Το ασύμμετρο TSP ή aTSP είναι η γενικότερη παραλλαγή του προβλήματος και αντιστοιχεί σε ένα κατευθυνόμενο βεβαρημένο γράφημα. Κάθε ακμή ανάλογα με τον προσανατολισμό της έχει και διαφορετικό βάρος. Τα περισσότερα πραγματικά προβλήματα είναι συνήθως ασύμμετρα.

1.3.2 Συμμετρικό TSP

Το συμμετρικό (symmetric TSP) αντιστοιχεί σε ένα μη κατευθυνόμενο βεβαρυμμένο γράφημα. Κάθε ακμή μεταξύ δύο πόλεων – σημείων του προβλήματος έχει ακριβώς το ίδιο κόστος ανεξαρτήτως της κατεύθυνσης που θα κινηθεί ο πωλητής. Παρατηρούμε ότι ο πίνακας κόστους του αντίστοιχου γραφήματος είναι συμμετρικός. Η μαθηματική τυποποίηση του προβλήματος ως πρόβλημα ακέραιου γραμμικού προγραμματισμού σύμφωνα με τους Dantzig, Fulkerson και Johnson ,είναι η ακόλουθη:

Δοθέντος πλήρους μη κατευθυνόμενου γραφήματος G = (V,A) οπου V= {1,2,… n}

1.3.3 Μετρικό TSP

Στο Μετρικό TSP (Metric-TSP ή delta – TSP ή Δ -TSP), τα βάρη των ακμών είναι συμμετρικά και παράλληλα ικανοποιούν την τριγωνική ανισότητα (triangle inequality). Η τριγωνική ανισότητα είναι χαρακτηριστική ιδιότητα μιας Μετρικής συνάρτησης. Υπενθυμίζουμε τον ορισμό της Μετρικής:

Μετρική ονομάζεται μια συνάρτηση $d : V \to R$, όπου $V \neq \emptyset$ σύνολο, η οποία ικανοποιεί τις παρακάτω ιδιότητες για κάθε $x,y,z \in V$:

- $d(x,y)=0$ αν και μόνο αν $x=y$
- $d(x,y)=d(y,x)$
- $d(x,y) \leq d(x,z)+ d(z,y)$ (τριγωνική ανισότητα)

Η πρώτη ιδιότητα αποκλείει TSP που μπορεί το γράφημα τους να έχει Βρόγχους. Η δεύτερη ιδιότητα απαιτεί το γράφημα του προβλήματος να είναι συμμετρικό.

1.3.4 Πολλαπλό TSP

Αντί για ένα μόνο πωλητή έχουμε m πωλητές διαθέσιμους οι οποίοι βρίσκονται όλοι στην πόλη n +1 και πρέπει να επισκεφτούν τις πόλεις 1,2,….n . Το κόστος της λύσης είναι η συνολική απόσταση που διανύεται από το σύνολο των πωλητών (όλοι πρέπει να ταξιδέψουν). Αυτή είναι η βασική περίπτωση όταν στην δρομολόγηση οχημάτων m οχήματα, βρίσκονται σε ένα κοινό σταθμό και πρέπει να εξυπηρετήσουν πελάτες.

Μπορούμε να μετασχηματίσουμε αυτό το πρόβλημα σε TSP χωρίζοντας την πόλη n+1 σε m πόλεις n+1,n+2,…n+m . Οι ακμές (i,n+k), $1 \leq i \leq$ n και $2 \leq k \leq m$,λαμβάνουν βάρος c(i,n+k)=c (i,n+1), και όλες οι ακμές που συνδέουν τους κόμβους n+1, n+2, …n+m λαμβάνουν ένα μεγάλο βάρος M.

1.3.5 Το Ευκλείδειο - TSP

Το Ευκλείδειο - TSP (Euclidean TSP) είναι ένα Μετρικό-TSP στο οποίο Μετρική είναι η Ευκλείδεια Μετρική. Το πρόβλημα έχει αποδειχθεί ότι είναι NP-Πλήρες [20,p. 237–244], Η Ευκλείδεια μετρική ορίζεται για d-διάστατους χώρους αν x,y∈ V με x = (x1, x2, . . . , xd) και y = (y1, y2, . . ., yd) τότε η απόσταση τους είναι d(x,y)= $\sum_{i=1}^{d}\sqrt{(x_i - y_i)}$.

Αν d=2 τότε το πρόβλημα βρίσκεται στο συνήθη δισδιάστατο χώρο και καλείται επίπεδο-TSP (planar).

1.3.6 Μέγιστο –TSP

Το Μέγιστο TSP (Max – TSP) είναι μια παραλλαγή στην οποία αναζητούμε την περιοδεία με το μεγαλύτερο κόστος. Το πρόβλημα ανάγετε εύκολα σε κανονικό TSP με την χρήση αρνητικών βαρών, ενώ αν η μέθοδος επίλυσης που επιλέγουμε απαιτεί θετικά βάρη στο γράφημα μας, μπορούμε να προσθέσουμε μια μεγάλη σταθερά χωρίς βλάβη της άριστης περιοδείας.

1.3.7. Bottleneck TSP

Στο TSP λαιμού μπουκαλιού (bottleneck TSP) σκοπός μας είναι να βρούμε μια περιοδεία της οποίας το μεγαλύτερο βάρος μιας ακμή να είναι όσο το δυνατόν μικρότερο. Το πρόβλημα έχει τις ρίζες του στην κίνηση των αυτοκινήτων, καθώς επιθυμούμε να ελαχιστοποιήσουμε το χρόνο οδήγησης από μια πόλη σε μια άλλη και όχι κατ' ανάγκην ολόκληρης της διαδρομής.

1.4 ΕΦΑΡΜΟΓΕΣ TSP

1.4.1 Εκτύπωση της Πλακέτας Κυκλωμάτων

Μια άμεση εφαρμογή του TSP είναι το πρόβλημα της εκτύπωσης μιας πλακέτας κυκλωμάτων του οποίου η λύση διαδραματίζει σημαντικό ρόλο στην οικονομική παραγωγή των πινάκων κυκλωμάτων (printed circuit boards - PCB).

Για να ενώσουμε ένα αγωγό πάνω σε ένα επίπεδο με ένα αγωγό που είναι πάνω σε ένα άλλο επίπεδο, ή για να τοποθετήσουμε (σε μεταγενέστερο στάδιο της παραγωγής των πλακετών) τις ακίδες των ολοκληρωμένων κυκλωμάτων, πρέπει να δημιουργήσουμε τρύπες στην πλακέτα. Οι τρύπες μπορεί να είναι και διαφορετικής διαμέτρου. Για να ανοίξουμε τρύπες διαφορετικής διαμέτρου διαδοχικά, η κεφαλή της μηχανής πρέπει να κινηθεί σε μια εργαλειοθήκη και να αλλάξει τον εξοπλισμό τρυπήματος. Αυτό δαπανεί αρκετό χρόνο. Έτσι είναι ξεκάθαρο στην αρχή ότι κάποιος πρέπει να επιλέξει κάποια διάμετρο, να ανοιχτούν όλες οι τρύπες της ίδιας διαμέτρου, να αλλαχτεί ο εξοπλισμός, να ανοιχτούν πάλι οι τρύπες της επόμενης ίδιας διαμέτρου, κλπ.

Έτσι, το πρόβλημα της εκτύπωσης μπορεί να θεωρηθεί ως μια ακολουθία περιπτώσεων TSP,μια για κάθε διάμετρο τρύπας, όπου οι "πόλεις" είναι η αρχική θέση και το σύνολο όλων των τρυπών που μπορεί να τρυπηθούν με το ίδιο τρυπάνι. Η “απόσταση” μεταξύ 2 πόλεων δίνεται από τον χρόνο που χρειάζεται η κεφαλή του τρυπανιού να κινηθεί από την μια θέση στην άλλη. Ο στόχος εδώ είναι η ελαχιστοποίηση του χρόνου της περιήγησης της κεφαλής του τρυπανιού.

1.4.2 Προετοιμασία Παραγγελιών σε Αποθήκες

Αυτό το πρόβλημα συνδέεται με την διαχείριση υλικού μιας αποθήκης.Υποθέτουμε ότι σε μια αποθήκη φτάνει μια παραγγελία για ένα ορισμένο υποσύνολο των αντικειμένων που είναι αποθηκευμένα στην αποθήκη. Κάποιο όχημα πρέπει να μαζέψει όλα τα αντικείμενα της παραγγελίας και να τα στείλει στον πελάτη. Η σχέση με το TSP διαφαίνεται αμέσως. Οι θέσεις αποθήκευσης των αντικειμένων αντιστοιχούν στους κόμβους του γραφήματος. Η απόσταση μεταξύ δύο κόμβων δίνεται από τον χρόνο που χρειάζεται να μετακινηθεί το όχημα από μια θέση σε μια άλλη. Το πρόβλημα εύρεσης της μικρότερης διαδρομής για το όχημα με ελάχιστο χρόνο ανάληψης μπορεί τώρα να λυθεί ως ένα TSP. Σε ειδικές περιπτώσεις αυτό το πρόβλημα μπορεί να λυθεί εύκολα.

1.4.3 Καλωδίωση Υπολογιστή

Μια ειδική περίπτωση σύνδεσης εξαρτημάτων σε μια κάρτα υπολογιστή αναφέρεται στο Lenstra & Rinnoy Kan [1974]. Οι λειτουργικές μονάδες που βρίσκονται πάνω σε μια κάρτα υπολογιστή και ένα δοθέν σύνολο ακίδων πρέπει να συνδεθούν. Σε αντίθεση με την συνήθη περίπτωση όπου ένα δέντρο σύνδεσης Steiner είναι επιθυμητό, εδώ η προϋπόθεση είναι ότι σε κάθε ακίδα επισυνάπτονται όχι περισσότερα από δύο καλώδια. Ως εκ τούτου έχουμε το

πρόβλημα της εύρεσης του συντομότερου μονοπατιού Hamilton με ακαθόριστο σημείο έναρξης και περάτωσης.

Μια παρόμοια περίπτωση εμφανίζεται στην επονομαζόμενη καλωδίωση testbus. Για να ελεγχθεί η κατασκευασμένη κάρτα κάποιος πρέπει να αντιληφθεί ότι μια σύνδεση η οποία εισάγεται στην κάρτα σε καθορισμένο σημείο, διατρέχει όλες τις λειτουργικές μονάδες, και τερματίζει σε ένα καθορισμένο σημείο. Για κάθε τέτοια μονάδα έχουμε επίσης μια καθορισμένη είσοδο και ένα σημείο εξόδου γι' αυτόν τον έλεγχο καλωδίωσης. Αυτό το πρόβλημα επίσης συνιστά στην επίλυση ενός προβλήματος μονοπατιού Hamilton με τη διαφορά ότι οι αποστάσεις δεν είναι συμμετρικές και ότι τα σημεία αφετηρίας και τερματισμού καθορίζονται.

1.4.4 Έλεγχος των Κινήσεων ενός Ρομπότ

Για να κατασκευάσουμε ένα κομμάτι μιας μηχανής, ένα ρομπότ πρέπει να εκτελέσει μια σειρά από δραστηριότητες σε αυτό (τρύπημα οπών διαφόρων διαμέτρων, κοπή σχισμής, σχεδιασμός κλπ). Η δουλειά που πρέπει να γίνει, είναι να καθοριστεί μια ακολουθία των απαραίτητων εργασιών που οδηγούν στον συντομότερο συνολικό χρόνο διαδικασίας. Μια δυσκολία προκύπτει από αυτή την εφαρμογή επειδή υπάρχει προτεραιότητα στους περιορισμούς οι οποίοι πρέπει να παρατηρηθούν. Έτσι εδώ έχουμε το πρόβλημα εύρεσης του συντομότερου μονοπατιού Hamilton (όπου οι αποστάσεις που αντιστοιχούν στους χρόνους χρειάζονται για την τοποθέτηση και τις πιθανές αλλαγές του τεμαχίου) που ικανοποιεί ορισμένες σχέσεις προτεραιότητας μεταξύ των δραστηριοτήτων.

1.4.5 Disk Scheduling Problem σε Hard Disk Drive

Μία ακόμα εφαρμογή του TSP με ιδιαίτερο ενδιαφέρον σε εφαρμογές hardware και γενικότερα πληροφορικής είναι το πρόβλημα της βέλτιστης σύγχρονης κίνησης της κεφαλής ενός σκληρό δίσκου και του σκληρού δίσκου μεταξύ μιας ομάδας δεδομένων επί του σκληρού δίσκου κατά τρόπο ώστε να ελαχιστοποιηθεί ο χρόνος κίνησης δίσκου και κεφαλής ανάγνωσης μεταξύ διαδοχικών αναγνώσεων. Το πρόβλημα αυτό ονομάζεται πρόβλημα προγραμματισμού σειράς ανάγνωσης δεδομένων σε δίσκο και είναι γνωστό στην διεθνή βιβλιογραφία ως Disk Scheduling Problem (DSP). Το DSP είναι ένα πολύ σημαντικό βιομηχανικής κλίμακας πρόβλημα και συσχετίζεται με διάφορα πρωτόκολλα που χρησιμοποιούν οι μικροελεγκτές εντός ενός σκληρού δίσκου (HDD).

Το συγκεκριμένο πρόβλημα εξετάζεται αναλυτικότερα στο Κεφάλαιο 6 προκειμένου να υλοποιηθεί με πλήθος δοκιμαστικών αριθμητικών δεδομένων που έχουν εκτιμηθεί με συμβατικά τεχνικά στοιχεία ενός τυπικού εμπορικού σκληρού δίσκου.

ΚΕΦΑΛΑΙΟ 2 : ΑΛΓΟΡΙΘΜΟΙ - ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

2.1 ΑΛΓΟΡΙΘΜΟΙ

Ως αλγόριθμος ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά αλγόριθμο ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και εκτελέσιμες που σκοπό έχουν την επίλυση κάποιου προβλήματος. Η λέξη αλγόριθμος προέρχεται από μία μελέτη

του Πέρση μαθηματικού του 8ου αιώνα μ.Χ. Αλ Χουαρίζμι (Abu Ja'far Mohammed ibn Musa Al-Khwarismi), η οποία περιείχε συστηματικές τυποποιημένες λύσεις αλγεβρικών προβλημάτων και αποτελεί ίσως την πρώτη πλήρη πραγματεία άλγεβρας. Πέντε αιώνες αργότερα η μελέτη μεταφράστηκε στα Λατινικά και άρχιζε με τη φράση "Algorithmus dixit ...." (ο Αλγόριθμος είπε...). Έτσι η λέξη αλγόριθμος καθιερώθηκε αργά τα επόμενα χίλια χρόνια με την έννοια «συστηματική διαδικασία αριθμητικών χειρισμών». Τη σημερινή της σημασία την οφείλει στη γρήγορη ανάπτυξη των ηλεκτρονικών υπολογιστών στα μέσα του20ου αιώνα. Ο όρος αλγόριθμος χρησιμοποιείται για να δηλώσει μεθόδους που εφαρμόζονται σε προγράμματα για την επίλυση προβλημάτων. Ωστόσο, ένας πιο αναλυτικός ορισμός της έννοιας αυτής είναι ο εξής: «Αλγόριθμος είναι ένα πεπερασμένο σύνολο εντολών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο οι οποίες αν ακολουθηθούν επιτυγχάνεται ένα επιθυμητό αποτέλεσμα.»

Από τον ορισμό προκύπτει ότι για να χαρακτηριστεί μια ακολουθία βημάτων – διαδικασία ως αλγόριθμος, θα πρέπει να έχει τα εξής χαρακτηριστικά:

Είσοδος (Input) : Καμία, μία ή περισσότερες ποσότητες να δίνονται ως είσοδοι στον αλγόριθμο.

Έξοδος(Output) : Ο αλγόριθμος να δημιουργεί τουλάχιστον μια ποσότητα ως αποτέλεσμα.

Καθορισμός(Definiteness) : Κάθε εντολή της διαδικασίας να καθορίζεται χωρίς

καμία αμφιβολία για τον τρόπο εκτέλεσής της.

Περατότητα (Finiteness) : Ο αλγόριθμος να τελειώνει μετά από πεπερασμένα βήματα εκτέλεσης των εντολών του. Μια διαδικασία που δεν τελειώνει μετά από πεπερασμένο αριθμό βημάτων, λέγεται απλώς υπολογιστική διαδικασία.

Αποτελεσματικότητα (Effectiveness) : Κάθε μεμονωμένη εντολή του αλγορίθμου να είναι απλή. Αυτό σημαίνει ότι μια εντολή δεν αρκεί να έχει ορισθεί, αλλά πρέπει να είναι και εκτελέσιμη.

Σημαντικά χαρακτηριστικά στην μελέτη αλγορίθμων, είναι η εκτίμηση της επίδοσης και της αποδοτικότητάς τους.

2.2 ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

2.2.1 ΙΣΤΟΡΙΚΗ ΑΝΑΔΡΟΜΗ

Τις δεκαετίες του 1950–1960 πολλοί επιστήμονες της επιστήμης υπολογιστών πραγματοποίησαν μελέτες σε εξελικτικά συστήματα με την σκέψη ότι η εξέλιξη θα μπορούσε να χρησιμοποιηθεί ως εργαλείο βελτιστοποίησης για προβλήματα που απασχολούσαν μηχανικούς. Η ιδέα σε όλα αυτά τα συστήματα ήταν να αναπτύξουν έναν πληθυσμό «υποψηφίων» λύσεων σε ένα δεδομένο πρόβλημα χρησιμοποιώντας τελεστές παρόμοιους με αυτούς που χρησιμοποιεί η φύση κατά την διάρκεια γενετικών παραλλαγών και φυσικής επιλογής.

Η φύση έχει έναν πολύ ισχυρό μηχανισμό εξέλιξης των οργανισμών, που βασίζεται στον ακόλουθο κανόνα της φυσικής επιλογής: οι οργανισμοί που δε μπορούν να επιβιώσουν στο περιβάλλον τους πεθαίνουν, ενώ οι υπόλοιποι πολλαπλασιάζονται μέσω της αναπαραγωγής. Οι απόγονοι παρουσιάζουν μικρές διαφοροποιήσεις από τους προγόνους τους, ενώ συνήθως υπερισχύουν αυτοί που συγκεντρώνουν τα καλύτερα χαρακτηριστικά.

Αν το περιβάλλον μεταβάλλεται με αργούς ρυθμούς, τα διάφορα είδη μπορούν να εξελίσσονται σταδιακά ώστε να προσαρμόζονται σε αυτό. Αν όμως συμβούν ραγδαίες μεταβολές, αρκετά είδη οργανισμών θα εξαφανιστούν. Σποραδικά, συμβαίνουν τυχαίες μεταλλάξεις, από τις οποίες οι περισσότερες οδηγούν τα μεταλλαγμένα άτομα στο θάνατο, αν και είναι πιθανό, πολύ σπάνια όμως, να οδηγήσουν στη δημιουργία νέων «καλύτερων» οργανισμών.

Η θεωρία της εξέλιξης (evolution) έχει χρησιμοποιηθεί σε μία κατηγορία αλγορίθμων επίλυσης προβλημάτων, που ονομάζονται γενετικοί αλγόριθμοι (genetic algorithms).

Η εισαγωγή των αλγορίθμων αυτών έγινε το 1958 από τον Friedberg, ο οποίος επιχείρησε να συνδυάσει μικρά προγράμματα Fortran, ωστόσο τα προγράμματα που προέκυψαν τις περισσότερες φορές δεν ήταν εκτελέσιμα.

Οι γενετικοί αλγόριθμοι τελικά επινοήθηκαν από τον John Holland την δεκαετία του 1960 και αναπτύχθηκαν από τους φοιτητές του και τους συνεργάτες του στο πανεπιστήμιο του Μίσιγκαν.

Είναι μια μέθοδος για μεταφορά από έναν πληθυσμό «χρωμοσωμάτων» σε έναν άλλο πληθυσμό χρησιμοποιώντας ένα είδος φυσικής επιλογής μαζί με γενετικούς τελεστές όπως :

Διασταύρωσης (crossover), δηλαδή ανταλλαγή γενετικού υλικού μεταξύ δύο χρωμοσωμάτων από απλοειδείς γονείς.

Μετάλλαξης (mutation), δηλαδή αναστροφή γενετικού υλικού σε έναν τυχαία επιλεγμένο τόπο.

Αντιστροφής (inversion), δηλαδή ανακατανομή της διάταξης των γονιδίων σε ένα χρωμόσωμα.

Επιλογής (selection), δηλαδή του διαχωρισμού των χρωμοσωμάτων που θα χρησιμοποιηθούν για την παραγωγή της επόμενης γενεάς.

Ο γενετικός αλγόριθμος εκτελεί μία αναζήτηση στο χώρο των υποψηφίων λύσεων, με στόχο την εύρεση κάποιας λύσης που μεγιστοποιεί τη συνάρτηση καταλληλότητας (αντικειμενική συνάρτηση κόστους). Η αναζήτηση αυτή είναι παράλληλη, καθώς σε κάθε υποψήφια λύση μπορεί να εκτελεστεί ξεχωριστή αναζήτηση. Η μέθοδος της αναζήτησης μπορεί να θεωρηθεί σαν αναρρίχηση λόφου, καθώς γίνονται μικρές αλλαγές στις υποψήφιες λύσεις του πληθυσμού και επιλέγονται πάντα οι καλύτερες, βάσει της συνάρτησης καταλληλότητας. Η αναζήτηση επικεντρώνεται στις περισσότερο κατάλληλες λύσεις, χωρίς όμως να αγνοούνται οι υπόλοιπες, καθώς υπάρχει πάντα ο κίνδυνος να παγιδευτεί η διαδικασία σε τοπικό μέγιστο.

Μέχρι τις αρχές της δεκαετίας του 1980 η μελέτη των γενετικών αλγορίθμων περιορίζονταν σε θεωρητικό επίπεδο, με λίγες πρακτικές εφαρμογές. Στην περίοδο αυτή οι γενετικοί αλγόριθμοι χρησιμοποιούνταν κυρίως για προβλήματα βελτιστοποίησης συναρτήσεων, χρησιμοποιώντας χρωμοσώματα συγκεκριμένου μήκους, δυαδικής κωδικοποίησης. Χαρακτηριστικές είναι οι μελέτες των Hollstien και De Jong την περίοδο αυτή.

Ο Hollstien πραγματοποίησε ανάλυση για το πώς επιδρούν οι τελεστές της επιλογής και της διασταύρωσης στην επίδοση του αλγορίθμου. Ο De Jong προσπάθησε να προσδιορίσει τα στοιχεία εκείνα που εμποδίζουν τους αλγορίθμους να φτάσουν γρήγορα σε σύγκλιση.

Από την αρχή της δεκαετίας του ’80, η επιστημονική κοινότητα που ασχολείται με τη θεωρία και εφαρμογή των γενετικών αλγόριθμων, έχει δημιουργήσει μία πληθώρα πρακτικών εφαρμογών που εκτείνονται σε πολλούς τομείς ερευνητικής και όχι μόνο δραστηριότητας. Βελτιώνοντας την απόδοση των γενετικών με τη ρύθμιση και βελτιστοποίηση των γενετικών τελεστών, αποδεικνύεται ότι οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν σε διάφορες κατηγορίες προβλημάτων και αποτελούν μία ισχυρή μέθοδο βελτιστοποίησης.

2.3 ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΓΕΝΕΤΙΚΟΥΣ ΑΛΓΟΡΙΘΜΟΥΣ

2.3.1 ΓΕΝΙΚΑ

Τα τελευταία τριάντα χρόνια, έχει παρατηρηθεί ένα συνεχώς αυξανόμενο ενδιαφέρον για την ανάπτυξη συστημάτων επίλυσης προβλημάτων βασισμένων στις αρχές της Γενετικής Εξέλιξης και της Κληρονομικότητας. Τα μειονεκτήματα των κλασσικών μεθόδων αναζήτησης και βελτιστοποίησης, καθώς και η συνεχώς αυξανομένη ανάγκη για παραγωγή λογισμικού που να μπορεί να εκμεταλλεύεται πιο αποδοτικά τις τεράστιες δυνατότητες του υλικού, ήταν η βασική αιτία που ώθησε τους επιστήμονες σ' αυτήν την αναζήτηση. Αυτού του είδους τα συστήματα λειτουργούν διατηρώντας ένα πληθυσμό κωδικοποιημένων πιθανών λύσεων και εφαρμόζοντας πάνω σε αυτό διάφορες διαδικασίες επιλογής του καλύτερου, καθώς και διάφορους γενετικούς τελεστές. Οι τελεστές αυτοί αντιγράφουν τον τρόπο με τον οποίο αναπαράγονται και μεταλλάσσονται τα χρωμοσώματα των κυττάρων των ζωντανών οργανισμών. Έτσι,περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων χρησιμοποιώντας, τόσο κομμάτια και στοιχεία από την προηγούμενη γενιά, όσο και εντελώς καινούρια κομμάτια που δοκιμάζονται για τυχόν καλή απόδοσή τους. Επανειλημμένες δοκιμές και πειράματα έχουν δείξει ότι μια "φυσική" αναπαράσταση των πιθανών λύσεων για ένα δεδομένο πρόβλημα, σε συνδυασμό με την εφαρμογή σε αυτή μιας οικογένειας γενετικών τελεστών, αποτελεί πολύ χρήσιμο εργαλείο στην προσπάθεια προσέγγισης των πραγματικών λύσεων σε μια πολύ μεγάλη ποικιλία προβλημάτων και εφαρμογών.

Αυτό το γεγονός, μετατρέπει αυτή τη "φυσικού μοντέλου" προσέγγιση σε μια πολλά υποσχόμενη κατεύθυνση, όσον αφορά την επίλυση προβλημάτων γενικότερα. Ένας Γ.Α. για ένα συγκεκριμένο πρόβλημα πρέπει να αποτελείται από τα παρακάτω πέντε

συστατικά:

1.Μια γενετική αναπαράσταση των πιθανών λύσεων του προβλήματος.

2.Ένα τρόπο δημιουργίας ενός αρχικού πληθυσμού των πιθανών λύσεων.

3.Μια αντικειμενική συνάρτηση αποτίμησης, που παίζει το ρόλο του Περιβάλλοντος κατατάσσοντας τις λύσεις με βάση την καταλληλότητά τους.

4.Γενετικούς τελεστές που μετατρέπουν τη σύνθεση των παιδιών.

5.Τιμές για διάφορες παραμέτρους που χρησιμοποιεί ο Γ.Α. (μέγεθος πληθυσμού, πιθανότητες εφαρμογής των γενετικών τελεστών, κτλ.).

2.3.2 ΔΟΜΗ ΕΝΟΣ ΓΕΝΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ένας τυπικός Γ.Α. περιλαμβάνει απλές λειτουργίες, που όμως κρύβουν μέσα τους μεγάλη ισχύ. Αυτός ο συνδυασμός απλοϊκότητας και ισχύος είναι το μεγαλύτερο θέλγητρο της τεχνικής τους. Στην παράγραφο αυτή παρουσιάζονται τα βασικά στοιχεία, που πρέπει να έχει ένας αλγόριθμος, ώστε να θεωρείται γενετικός και δίνεται ένα παράδειγμα που κάνει αντιληπτή την εφαρμογή της θεωρίας. Αρχικά σε ένα Γ.Α. πρέπει να υπάρχουν στοιχεία που θα τον συνδέουν με το πρόβλημα που επιλύει. Η κωδικοποίηση και η αντικειμενική συνάρτηση επιτελούν αυτό το σκοπό και είναι "εκ των ων ουκ άνευ" συστατικά για ένα Γ.Α. Η κωδικοποίηση αφορά ένα σύνολο πιθανών λύσεων του προβλήματος. Η αναπαράσταση των λύσεων πρέπει να γίνει με ένα μαθηματικό, φορμαλιστικό τρόπο, ώστε να είναι δυνατή η επεξεργασία από τον υπολογιστή. Εξ' άλλου, κωδικοποίηση υπάρχει και στο φυσικό μοντέλο (χρωμοσώματα) και μάλιστα, όλες οι αλλαγές που παρατηρούνται στους οργανισμούς γίνονται πάνω στα κωδικοποιημένα χαρακτηριστικά των χρωμοσωμάτων.

Κύριος στόχος της κωδικοποίησης είναι να αναπαριστά με ικανοποιητικό τρόπο τα επιμέρους χαρακτηριστικά των λύσεων, ώστε να διευκολύνει τις επόμενες λειτουργίες του αλγορίθμου (κυρίως την επιλογή). Αποτέλεσμα της κωδικοποίησης πρέπει να είναι η ύπαρξη ομοιοτήτων ανάμεσα στα άτομα με σκοπό την κατάλληλη εκμετάλλευση τους, διότι οι ομοιότητες βοηθούν την κατεύθυνση του ψαξίματος. Διάφορα είναι τα είδη της κωδικοποίησης που μπορούν να γίνουν από πρόβλημα σε πρόβλημα. Η πιο απλή είναι η κωδικοποίηση με δυαδικά ψηφία (bits): κάθε λύση αναπαρίσταται από μια δυαδική συμβολοσειρά (binary string) καθορισμένου μήκους. Πάντως, έχουν αναφερθεί ποικίλες μορφές κωδικοποιήσεων, που κάθε μια εξαρτάται από το υπό εξέταση πρόβλημα. Καμιά δεν είναι αποτελεσματική για όλα τα προβλήματα, ενώ είναι πιθανό ένα πρόβλημα να επιδέχεται περισσότερες από μια κωδικοποιήσεις. Το σίγουρο είναι ότι η κωδικοποίηση είναι ένα κρίσιμο βήμα στην εφαρμογή του Γ.Α. και αν δεν είναι προσεκτική, πιθανότατα θα αποβεί μοιραία για την επιτυχία του. Η καταλληλότητα της κωδικοποίησης εξαρτάται σε μεγάλο βαθμό από τη διαίσθηση και την πείρα του σχεδιαστή. Συμβαίνει μερικές φορές μάλιστα, προφανείς τρόποι κωδικοποίησης να είναι λίγο (ή και καθόλου) αποτελεσματικοί. Κατά συνέπεια προκύπτει το κρίσιμο ερώτημα: ποιοι είναι οι παράγοντες που καθορίζουν το είδος της κωδικοποίησης που πρέπει να επιλεχθεί για ένα συγκεκριμένο πρόβλημα; Δεν υπάρχει ξεκάθαρη απάντηση που να καλύπτει όλες τις περιπτώσεις. Μερικές γενικού τύπου συμβουλές θα φανούν στην παραπέρα ανάπτυξη του θέματος.

Για παράδειγμα, έστω η συνάρτηση f(x)=x2, με x=[0,3] και x ακέραιος. Ζητείται το μέγιστο της συνάρτησης στο πεδίο ορισμού της. Για να λυθεί το πρόβλημα από ένα Γ.Α. πρέπει να επινοηθεί ένας τρόπος κωδικοποίησης των πιθανών λύσεων. Ο πιο προφανής και τελικά, όπως θα αποδειχθεί, πιο αποτελεσματικός τρόπος κωδικοποίησης είναι να αναπαρασταθεί η κάθε λύση με μια δυαδική συμβολοσειρά μήκους 5, που αριθμητικά θα ισοδυναμεί μα την αντίστοιχη δεκαδική τιμή της λύσης. Έτσι καλύπτεται όλο το πεδίο ορισμού από τις 32 δυνατές συμβολοσειρές αυτού του είδους. Πχ η συμβολοσειρά 10010 αντιστοιχεί, κατά τα γνωστά, στην τιμή 18 του δεκαδικού συστήματος. Συνήθως, σε προβλήματα βελτιστοποίησης μαθηματικών συναρτήσεων, η δυαδική είναι η πιο βολική και πιο αποδοτική κωδικοποίηση.

Το δεύτερο βασικό στοιχείο της σύνδεσης ενός Γ.Α. με το πρόβλημα που λύνει, είναι η αντικειμενική συνάρτηση. Αυτή παίρνει ως είσοδο μια αποκωδικοποιημένη συμβολοσειρά και επιστρέφει μια τιμή (συνήθως πραγματική), που είναι ανάλογη του πόσο καλά λύνει το πρόβλημα η συγκεκριμένη συμβολοσειρά. Η τιμή αυτή αποτελεί και τον καθοριστικό παράγοντα επιβίωσης και πολλαπλασιασμού ή όχι του ατόμου.Η αντικειμενική συνάρτηση παίζει το ρόλο του περιβάλλοντος στο τεχνικό μοντέλο. Ουσιαστικά, είναι η μόνη πληροφορία που δέχεται ο αλγόριθμος για το πρόβλημα που λύνει. Είναι σημαντικό αυτή η συνάρτηση να είναι εύκολα υπολογίσιμη, ώστε να μην επιβραδύνει τους ρυθμούς της διαδικασίας. Επιστρέφοντας στο παράδειγμα, η συνάρτηση ικανότητας του προβλήματος μεγιστοποίησης είναι φανερό ότι πρέπει να είναι η ίδια η f, γιατί ουσιαστικά το ζητούμενο είναι η μεγιστοποίηση αυτής της συνάρτησης. Έτσι σε κάθε λύση, δηλαδή σε κάθε πιθανή τιμή της μεταβλητής x,αντιστοιχεί μια τιμή ικανότητας ή απόδοσης (fitness ή score), μια τιμή που αξιολογεί το πόσο καλή είναι η λύση για τη μεγιστοποίηση της συνάρτησης και που, για αυτή την περίπτωση είναι η ίδια η εικόνα της στην f. Με τον καθορισμό της κωδικοποίησης και της αντικειμενικής συνάρτησης ορίζεται πλέον το πρόβλημα και ολοκληρώνεται το πρώτο στάδιο εφαρμογής ενός Γ.Α. Αξίζει να σημειωθεί η αυτονομία και ανεξαρτησία αυτού του σταδίου από τα επόμενα μέρη. Οι λειτουργίες που ακολουθούν από εδώ και πέρα δεν εξαρτώνται από το πώς γίνεται η αναπαράσταση των ατόμων στο τεχνητό περιβάλλον και με ποιο τρόπο αξιολογούνται οι ικανότητές τους. Αυτό είναι σπουδαίο χαρακτηριστικό, διότι επιτρέπει την διαπραγμάτευση πολλών προβλημάτων με μια απλή αλλαγή στην συνάρτηση ικανότητας, ίσως και στην κωδικοποίηση. Η φάση ορισμού της κωδικοποίησης και της αντικειμενικής συνάρτησης υπάρχουν πάντα σε κάθε Γ.Α. ανεξαρτήτως του προβλήματος.

Στο επόμενο στάδιο περιλαμβάνονται λειτουργίες που ανήκουν στη φάση τρεξίματος του Γ.Α. Εδώ γίνεται ο κύριος όγκος της εργασίας και παράγεται το αποτέλεσμα τηςβελτιστοποίησης. Η δομή της λειτουργίας ενός Γ.Α. αποτελείται από τα παρακάτω

Βήματα:

1. Δημιουργία αρχικού πληθυσμού (Initialization)

2. Επαναληπτικό μέρος του αλγόριθμου (δημιουργεί νέες γενιές βάσει των προηγούμενων)

a) Υπολογισμός της ικανότητας των ατόμων (Fitness calculation)

b) Διαβάθμιση των ικανοτήτων (scaling)

c) Επιλογή των ικανότερων ατόμων και αντιγραφή τους στην επόμενη γενιά (ελιτισμός).

d) Επιλογή γονέων

e) Αναπαραγωγή

• Διασταύρωση ζευγαριών ατόμων (Crossover)

• Μετάλλαξη μεμονωμένων ατόμων (Mutation)

f) Αντικατάσταση της παλιάς γενεάς με την καινούρια.

g) Ο αλγόριθμος ολοκληρώνεται όταν ικανοποιηθούν τα κριτήρια τερματισμού

Η αρχικοποίηση είναι το βήμα στο οποίο ορίζεται ο αρχικός πληθυσμός, πάνω στον οποίο θα λάβουν χώρα οι λειτουργίες του Γ.Α. Ο πληθυσμός αυτός διαλέγεται με τυχαίο τρόπο ανάμεσα σε όλες τις δυνατές τιμές των μεταβλητών του προβλήματος,

ενώ το μέγεθός του ορίζεται από το χρήστη (συνήθως, όμως, εξαρτάται από τους πόρους τους που έχει στη διάθεσή του). Σε μερικές μερικές υλοποιήσεις, η επιλογή των αρχικών σημείων γίνεται με ευρετικές μεθόδους, δίνοντας ένα εξ αρχής πλεονέκτημα στην αναζήτηση. Έστω στο παράδειγμά μας, ότι το μέγεθος του πληθυσμού είναι 4. Μένει να επιλεχθούν τυχαία τέσσερις συμβολοσειρές από τις 32 πιθανές. Αυτό μπορεί να γίνει με 20 διαδοχικές ρίψεις ενός τίμιου νομίσματος, ώστε να προκύψουν 4 συμβολοσειρές μήκους 5 η κάθε μία. Ένα πιθανό σενάριο θα μπορούσε να βγάλει τις συμβολοσειρές 01101, 11000, 01000 και 10011.

Αφού προκύψει η πρώτη γενιά, ο Γ.Α. εισέρχεται στο επαναληπτικό μέρος του. Ο πληθυσμός πρέπει να αξιολογηθεί, δηλαδή να μετρηθεί η ικανότητα επιβίωσης του κάθε ατόμου χωριστά. Για να συμβεί αυτό πρέπει να γίνει αποκωδικοποίηση χαρακτηριστικών και έπειτα υπολογισμός της απόδοσης των ατόμων. Ο παραλληλισμός με το φυσικό μοντέλο ίσως βοηθά στην κατανόηση αυτής της διαδικασίας: Στη φύση τα χρωμοσώματα ενός οργανισμού έχουν στα γονίδιά τους κωδικοποιημένα τα χαρακτηριστικά τους. Το σύνολο αυτής της κωδικοποιημένης γενετικής πληροφορίας ονομάζεται, όπως είπαμε, γονότυπος. Ο γονότυπος δεν είναι αντιληπτός με τις φυσικές αισθήσεις των έμβιων όντων. Αντίθετα, αντιληπτή γίνεται η αλληλεπίδραση του με το περιβάλλον, που έχει ως αποτέλεσμα την ορατή εμφάνιση των χαρακτηριστικών αυτών. Ανάλογος είναι ο ρόλος της αποκωδικοποίησης στο τεχνητό μοντέλο. Εδώ το ρόλο του γονότυπου παίζει η δομή της συμβολοσειράς με τα δυαδικά ψηφία ως αντίστοιχο των γονιδίων. Ο φαινότυπος αναφέρεται στην παρατηρήσιμη εμφάνιση μιας συμβολοσειράς, δηλαδή στο πώς φαίνεται στο περιβάλλον της. Περιβάλλον όμως, θεωρείται η αντικειμενική συνάρτηση, άρα ο φαινότυπος μιας συμβολοσειράς αντιστοιχεί στην αποκωδικοποιημένη τιμή του, που ανήκει στο σύνολο ορισμού της αντικειμενικής συνάρτησης.

Σκοπός της λειτουργίας αξιολόγησης είναι να υπολογιστεί για κάθε άτομο του πληθυσμού η ικανότητα του για επιβίωση. Στη φύση οι ικανότητες των ατόμων δεν μπορούν να προσδιοριστούν με αυστηρό τρόπο. Είναι όμως καθορισμένες από το γενετικό υλικό των χρωμοσωμάτων τους. Εύκολα πάντως θα μπορούσε κανείς να ισχυριστεί, πχ για τα ζώα, ότι μεγαλύτερη τύχη για επιβίωση έχουν όσα μπορούν να ξεφεύγουν από άρπαγες, να αντέχουν σε αρρώστιες και γενικά να αντιπαρέρχονται τις όποιες αντιξοότητες παρουσιάζονται κατά τη διάρκεια της ζωής τους. Συνεπώς, ο υπολογισμός της ικανότητας είναι θεμελιώδης λειτουργία για το Γ.Α. Η εφαρμογή της είναι πολύ απλή (τουλάχιστον για απλά προβλήματα): για κάθε συμβολοσειρά του τρέχοντος πληθυσμού υπολογίζεται η απόδοσή της από την ήδη γνωστή αντικειμενική συνάρτηση. Σε πιο σύνθετα προβλήματα, ο υπολογισμός ικανότητας μπορεί να ισοδυναμεί με την εκτέλεση μιας εργαστηριακής προσομοίωσης.

Μετά από την παραγωγή μερικών γενεών, ο Γ.Α. αρχίζει σιγά σιγά να συγκλίνει προς το βέλτιστο σημείο του χώρου αναζήτησης. Ο όρος “σύγκλιση”, όσον αφορά τα άτομα του πληθυσμού, σημαίνει ότι τα περισσότερα από αυτά μοιάζουν αρκετά μεταξύ τους. Άμεσο αποτέλεσμα αυτού είναι οι ικανότητές τους να διαφέρουν από ελάχιστα έως καθόλου και κατά συνέπεια στην επιλογή υπάρχει πολύ μεγάλος ανταγωνισμός για το ποιος θα δώσει απογόνους στην επόμενη γενιά. Έτσι, τα άτομα που είναι τα πιο ικανά, έστω και με ελάχιστη διαφορά, δεν κατορθώνουν -κατά πάσα πιθανότητα- να επιβληθούν μέσα στο περιβάλλον τους και η πρόοδος του αλγορίθμου από ένα σημείο και πέρα γίνεται με πολύ αργούς ρυθμούς, χωρίς όμως η σύγκλιση να γίνεται προς το βέλτιστο σημείο, αλλά προς ένα τοπικό ακρότατο. Το φαινόμενο αυτό ονομάζεται πρόωρη σύγκλιση (premature convergence). Η πρόωρη σύγκλιση μπορεί να αποφευχθεί με χρήση μηχανισμών διαβάθμισης (scaling mechanisms). Πρόκειται για τεχνικές που έχουν στόχο την επαύξηση των διαφορών – όσο μικρές κι αν είναι - των ικανοτήτων των ατόμων και την πιο εύκολη ανάδειξη των καλύτερων μέσα από την λειτουργία της επιλογής. Με τους μηχανισμούς διαβάθμισης εγκαταλείπεται το κλασσικό μοντέλο της ρουλέτας και πλέον οι ευκαιρίες των ατόμων στην επιλογή δεν είναι ανάλογες της ικανότητάς τους. Οι κυριότεροι μηχανισμοί που σήμερα

επικρατούν είναι:

● Γραμμική διαβάθμιση (Linear scaling).

● Εκθετική διαβάθμιση (proportional scaling).

● Διαβάθμιση με βάση τη σειρά (Fitness ranking).

● Παραθυροποίηση (Windowing).

Στη γραμμική διαβάθμιση οι τιμές των ικανοτήτων του πληθυσμού μετασχηματίζονται σύμφωνα με μια γραμμική σχέση της μορφής: $s=\alpha*f+\beta$ όπου τα α και β υπολογίζονται με βάση τις εξής δύο συνθήκες:

● ο μέσος όρος των ικανοτήτων πριν και μετά τον μετασχηματισμό να μην αλλάζει και

● οι μεγαλύτερες ικανότητες του πληθυσμού να μην υπερβαίνουν ένα

προκαθορισμένο πολλαπλάσιο του μέσου όρου (συνήθως διπλάσιο).

Ανάλογος είναι και ο τρόπος λειτουργίας της εκθετικής διαβάθμισης. Τα άτομα σχηματίζουν μια ταξινομημένη λίστα με βάση τις ικανότητές τους και η νέα τους ικανότητα, η εκθετικά διαβαθμισμένη ικανότητα s, προκύπτει εφαρμόζοντας τον μετασχηματισμό $s=f\kappa$ όπου κ συνήθως επιλέγεται να είναι η σειρά του κάθε ατόμου στη ταξινομημένη λίστα. Γενικά το κ μπορεί να ποικίλει από πρόβλημα σε πρόβλημα. Μπορεί ακόμη και να αλλάξει κατά το χρόνο εκτέλεσης για να περιορίζει ή να διευρύνει την ποικιλία ανάλογα με τις ανάγκες. Στη διαβάθμιση με βάση τη σειρά, στους μετασχηματισμούς που λαμβάνουν χώρα, δε χρησιμοποιείται καμία παράμετρος. Τα άτομα ταξινομούνται με βάση τις ικανότητές τους από το μικρότερο προς το μεγαλύτερο. Αποδίδεται στο μικρότερο μια νέα ικανότητα και συνεχίζοντας διαδοχικά προς το μεγαλύτερο, οι νέες ικανότητες αυξάνουν κατά μία σταθερή τιμή.

Η παραθυροποίηση είναι τεχνική σύμφωνα με την οποία οι τιμές των ικανοτήτων

τροποποιούνται πριν από την επιλογή ως εξής: Αφαιρείται από την ικανότητα κάθε ατόμου ποσό ίσο με την μικρότερη ικανότητα που υπάρχει στον πληθυσμό. Έτσι, οι διαφορές ικανοτήτων των ατόμων γίνονται πιο εμφανείς. Επειδή όμως κατ' αυτόν τον τρόπο, τα λιγότερο ικανά άτομα έχουν μηδαμινές ευκαιρίες επιλογής, μπορεί να τεθεί ένα κάτω όριο των ικανοτήτων μεγαλύτερου του μηδενός. Με την επιλογή, βρίσκει εφαρμογή στα πλαίσια του αλγόριθμου, ο νόμος της επιβίωσης του ικανότερου. Μέσω αυτής της διαδικασίας, καθορίζεται ποια άτομα από τον υπάρχοντα πληθυσμό θα έχουν την ευκαιρία να λάβουν μέρος στην αναπαραγωγή και να κληροδοτήσουν στην επόμενη γενιά μέρος ή το σύνολο των χαρακτηριστικών τους. Στόχος της λειτουργίας της επιλογής είναι να επιτρέπει εκθετική αύξηση των ικανοτέρων ατόμων και τελικά, μετά από αναπαραγωγή αρκετών γενεών, την επικράτησή τους. Ένας Γ.Α. χωρίς επιλογή στην αναπαραγωγική του διαδικασία ισοδυναμεί με τυχαίο ψάξιμο. Υπάρχουν διάφοροι τρόποι υλοποίησης της επιλογής στα πλαίσια ενός Γ.Α. Δεδομένου όμως, ότι στη βασική μορφή του αλγορίθμου το μέγεθος του πληθυσμού από γενιά σε γενιά δεν αλλάζει, κάθε τεχνική επιλογής, για να δικαιώνει τον τίτλο της, οφείλει να δίνει με κάποιο τρόπο μεγαλύτερες πιθανότητες αναπαραγωγής σε άτομα που αξιολογούνται μέσα στο τεχνητό περιβάλλον ως τα πιο ικανά. Ο τελεστής αναπαραγωγής μπορεί να εκφραστεί σε αλγοριθμική βάση, με πολλούς τρόπους. Ίσως ο ευκολότερος από αυτούς είναι η έκφραση μέσω μιας εξαναγκασμένης ρουλέτας, στην οποία κάθε συμβολοσειρά ενός πληθυσμού αντιπροσωπεύεται σε ένα μέρος της ρουλέτας, σε αναλογία με την απόδοσή της. Για να εξηγήσουμε τη χρήση της εξαναγκασμένης ρουλέτας, θεωρούμε τον πληθυσμό των τεσσάρων συμβολοσειρών, που έχουμε δημιουργήσει με τη ρίψη ενός νομίσματος 20 φορές.

2.3.3 ΕΦΑΡΜΟΓΕΣ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Στα πρώτα τους βήματα οι Γ.Α. αποτέλεσαν αντικείμενο μελέτης και ανάπτυξης σε

πανεπιστήμια και ερευνητικά κέντρα. Τα τελευταία χρόνια όμως, οι μεγάλες ανάγκες για δημιουργία αποδοτικών εφαρμογών στο χώρο της βελτιστοποίησης, σε συνδυασμό με την πολλά υποσχόμενη τεχνολογία των Γ.Α. Στη συνέχεια παρουσιάζονται μερικές αντιπροσωπευτικές εφαρμογές των γενετικών αλγορίθμων.

I. Εύρεση μέγιστης τιμής αριθμητικών συναρτήσεων

Πρόκειται για την πιο καλά μελετημένη εφαρμογή των γενετικών αλγορίθμων. Η εύρεση του μέγιστου μιας συνάρτησης δεν είναι καθόλου εύκολη υπόθεση για συναρτήσεις πολλών μεταβλητών, οι οποίες εμφανίζουν ασυνέχειες, θόρυβο, κλπ. Το πλεονέκτημα που εμφανίζει η εφαρμογή τους σε αυτά τα προβλήματα είναι ότι η συνάρτηση καταλληλότητας είναι δεδομένη.

II. Επεξεργασία εικόνων

Οι γενετικοί αλγόριθμοι χρησιμοποιούνται για την αναγνώριση προτύπων, όπως ακμές, επιφάνειες, ακόμη και αντικείμενα, σε ψηφιοποιημένες εικόνες. Το αποτέλεσμα αυτής της επεξεργασίας μπορεί να αποτελέσει τη βάση για την ψηφιακή όραση.

III. Σχεδίαση

Οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν στη σχεδίαση κατασκευών και εξαρτημάτων, όπως π.χ. γέφυρες, μηχανολογικά εξαρτήματα, όπου ζητούμενο μπορεί να είναι τόσο η εύρεση μιας λύσης, όσο και η βελτιστοποίηση της. Οι αλγόριθμοι μπορούν να δοκιμάσουν συνδυασμούς και ιδέες που ο ανθρώπινος νους δε θα δοκίμαζε ποτέ, δίνοντας ενίοτε πρωτότυπα αποτελέσματα.

IV. Μηχανική μάθηση

Στα συστήματα μηχανικής μάθησης οι γενετικοί αλγόριθμοι μπορούν να χρησιμοποιηθούν για την ανακάλυψη κανόνων if…then... Η πιο γνωστή εφαρμογή είναι αυτή των *συστημάτων κατηγοριοποίησης (classified systems),* ωστόσο οι γενετικοί αλγόριθμοι έχουν χρησιμοποιηθεί και σε παιχνίδια, επίλυση λαβυρίνθων, καθώς και για πολιτικές και οικονομικές αναλύσεις.

V. Συνδυαστική βελτιστοποίηση

Πρόκειται για το κλασσικό πρόβλημα κατανομής πόρων σε δραστηριότητες, με σκοπό τη μεγιστοποίηση του οφέλους ή την ελάττωση του κόστους. Τα προβλήματα αυτής τηςκατηγορίας παρουσιάζουν συνδυαστική έκρηξη του χώρου αναζήτησης, ως προς το μέγεθος του προβλήματος, με αποτέλεσμα ο έλεγχος όλων των υποψήφιων λύσεων να είναι αδύνατος. Το πιο γνωστό πρόβλημα αυτής της κατηγορίας είναι αυτό του *πλανόδιου πωλητή,* όπου στόχος είναι η εύρεση της συντομότερης διαδρομής για την επίσκεψη ενός συνόλου πόλεων.

Οι γενετικοί αλγόριθμοι μπορούν να δώσουν σε αυτό το πρόβλημα πολλές λύσεις κοντά στη βέλτιστη. Ένα άλλο πρόβλημα είναι η *αποθήκευση κιβωτίων (bin packing)* και αφορά την εύρεση του βέλτιστου τρόπου αποθήκευσης ενός αριθμού κιβωτίων σε περιορισμένο χώρο και έχει μεγάλη πρακτική σημασία στη βιομηχανία.

Τέλος στην κατηγορία αυτών των εφαρμογών εντάσσονται και τα προβλήματα *καταμερισμού – χρονοπρογραμματισμού εργασιών (Job shop & Flow shop scheduling).*

Γίνεται φανερό λοιπόν ότι οι γενετικοί αλγόριθμοι έχουν εφαρμοστεί σε διάφορα προβλήματα της Τεχνητής Νοημοσύνης και ιδιαίτερα σε προβλήματα βελτιστοποίησης. Σε ορισμένα προβλήματα τα αποτελέσματα ήταν πολύ καλά ενώ σε άλλα αρκετά απογοητευτικά.

2.3.4 ΔΥΣΚΟΛΙΕΣ ΠΟΥ ΕΧΟΥΝ ΠΑΡΟΥΣΙΑΣΤΕΙ ΣΤΗΝ ΕΦΑΡΜΟΓΗ ΤΟΥΣ

Η τεχνολογία των γενετικών αλγόριθμων αν και δεν αποτελεί πρόσφατη ανακάλυψη άρχισε ουσιαστικά να εφαρμόζεται τα τελευταία χρόνια. Ωστόσο, οι επιστήμονες αντιμετώπισαν πολλά προβλήματα σχετικά με την εφαρμογή τους ,κάτι που πλέον άρχισε να υποχωρεί. Ποιοι είναι όμως οι λόγοι που θα μπορούσαν να σταθούν εμπόδιο στην εξάπλωση της τεχνολογίας τους;

1)Προβλήματα εξοικείωσης με τη Γενετική. Για τους περισσότερους, που ασχολούνται με την Επιστήμη των Υπολογιστών, οι έννοιες της Εξέλιξης και της Φυσικής Επιλογής μπορεί να μην ηχούν παράξενα, αλλά δεν είναι και από τις πιο οικείες. Η Βιολογία δεν έχει άμεση σχέση με τους υπολογιστές, γι' αυτό και οι γνώσεις σχεδόν όλων είναι σε πολύ γενικό επίπεδο. Παρ' όλ' αυτά, δεν απαιτούνται γνώσεις Γενετικής και Βιολογίας. Εκείνο που συμβαίνει με τους Γ.Α. είναι ότι μιμούνται με αφαιρετικό τρόπο κάποιες διαδικασίες που παρατηρούνται στη φύση, χωρίς να ενδιαφέρει σε μεγάλο βαθμό λεπτομέρειας η λειτουργία τους και χωρίς να είναι απαραίτητο το γνωστικό υπόβαθρο που έχουν οι βιολόγοι για να μελετήσουν αυτά τα φαινόμενα. Οι όροι είναι δανεισμένοι από τη βιολογία με σκοπό την καλύτερη εισαγωγή και κατανόηση του θέματος και όχι την παραπομπή του μελετητή στα άγνωστα πεδία μιας ξένης επιστήμης και, τελικά, τη σύγχυσή του. Θα μπορούσε ίσως, να παραληφθεί η αναφορά στη Γενετική και να γίνει μια παρουσίαση των Γ.Α. ως "προσωπικές διαδικασίες για αναζήτηση και βελτιστοποίηση", αυτό όμως μάλλον θα έκανε τα πράγματα δυσκολότερα. Εξάλλου, είναι συνηθισμένο το φαινόμενο θεωρίες που είναι δανεισμένες από άλλες επιστήμες να διατηρούν την αυθεντική τους ορολογία (π.χ. στα Νευρωνικά Δίκτυα: νευρώνες, συνάψεις, κ.τ.λ.). Επιπλέον, το μέλλον και η εξέλιξη των Γ.Α. δεν εξαρτώνται σε καμία περίπτωση από τις αντίστοιχες θεωρίες της Βιολογίας. Το αρχικό μοντέλο είναι δανεισμένο από εκεί, όμως η εφαρμογή του στα Τεχνητά Συστήματα έγινε με πλήθος διαφοροποιήσεων, προσαρμόσεων και "παρεκτροπών" με στόχο πάντα τη βελτίωση της απόδοσης. Πλέον, μπορούμε να μιλάμε για εξέλιξη και απογόνους των πρώτων Γ.Α. και για μια πορεία τους στο χρόνο που είναι πλήρως ανεξάρτητη και αυτοδύναμη.

2) Το πρόβλημα του χρόνου. Στη φύση ως γνωστό, η εξέλιξη λειτουργεί με ρυθμούς πολύ αργούς. Χρειάζονται να περάσουν χιλιάδες γενιές, άρα και αρκετός χρόνος, για να αλλάξουν τα χαρακτηριστικά των ειδών και να διαφοροποιηθούν οι ικανότητες και η συμπεριφορά τους. Θέτουν έτσι ορισμένοι το ερώτημα: πώς είναι δυνατό ένα μοντέλο αναζήτησης λύσεων να έχει καλές επιδόσεις χρόνου, όταν είναι εμπνευσμένο από μια φυσική διαδικασία που εξελίσσεται με ρυθμούς απίστευτα αργούς; Η απάντηση εδώ είναι απλή. Κατ' αρχήν, ακόμη και στη φύση, η εξέλιξη δεν είναι από μόνη της μια αργή διαδικασία. Εξέλιξη των ειδών συμβαίνει όταν αλλάζει τα περιβάλλον τους και πρέπει να προσαρμοστούν στα καινούργια δεδομένα, ώστε να επιβιώσουν. Αλλαγές όμως του περιβάλλοντος γίνονται με πολύ αργούς ρυθμούς και κατά συνέπεια και η εξέλιξη ακολουθεί αυτούς τους ρυθμούς. Αν οι αλλαγές του περιβάλλοντος γίνονται με γρηγορότερο τρόπο, τότε επιταχύνεται και η εξέλιξη. Αυτό άλλωστε παρατηρείται και στα βιολογικά εργαστήρια, όπου μικροοργανισμοί αλλάζουν την συμπεριφορά τους αμέσως, όταν τοποθετούνται σε νέες συνθήκες. Επιπλέον, στο πεδίο των υπολογιστών τα άτομα κωδικοποιούνται συνήθως ως συμβολοσειρές και οι συνθήκες του περιβάλλοντος μοντελοποιούνται με απλές μαθηματικές σχέσεις. Έτσι, το μοντέλο με το οποίο δουλεύει ο υπολογιστής δεν παρουσιάζει ιδιαίτερο υπολογιστικό φόρτο, συγκρινόμενο πάντα με αντίστοιχες μεθόδους. Το πλήθος των ατόμων που κάθε φορά εξετάζεται είναι από λίγες δεκάδες έως μερικές χιλιάδες, δηλαδή αρκετές τάξεις μεγέθους κάτω από το πλήθος των γονιδίων των χρωμοσωμάτων μιας έμβιας οντότητας. Ο ρυθμός που μπορούν να ζευγαρώνουν τα άτομα στους πιο γρήγορους υπολογιστές μπορεί να φτάσει το ένα εκατομμύριο ανά δευτερόλεπτο. Όσο μεγάλος και αν είναι ο χώρος που καλείται ο αλγόριθμος να ψάξει, η επεξεργασία μερικών μόνο ατόμων αρκεί, γιατί, όπως θα αναπτυχθεί και παρακάτω, τα άτομα αυτά θεωρούνται αντιπρόσωποι ολόκληρων κλάσεων. Έτσι λοιπόν, οι ταχύτητες που μπορούν να επιτύχουν οι Γ.Α. είναι πολύ υψηλές.

2.3.5 ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι Γ.Α. αποτελούν μια πρωτότυπη μεταφορά ενός μοντέλου που λειτουργεί με επιτυχία για εκατομμύρια χρόνια στη φύση. Από αυτή τη σκοπιά, δημιουργείται τουλάχιστον έκπληξη, αν όχι θαυμασμός, στον απλό χρήστη για την πρωτοτυπία της ιδέας. Επιβεβαιώνεται, και σε αυτή την περίπτωση, η τάση της Επιστήμης να εμπνέεται από την ανθρώπινη ζωή. Από λειτουργική άποψη, οι Γ.Α. αποτελούν ένα ισχυρό και εύρωστο εργαλείο βελτιστοποίησης. Είναι σε θέση να αντιμετωπίζουν μεγάλη ποικιλία προβλημάτων μεγάλης δυσκολίας και να προσαρμόζονται σε πολλά περιβάλλοντα υλοποίησης. Παρ' όλα αυτά, για προβλήματα όχι μεγάλης πολυπλοκότητας και, όπου υπάρχουν εξειδικευμένες μέθοδοι βελτιστοποίησης, ίσως οι Γ.Α. να μην είναι η καλύτερη επιλογή, γιατί είναι εργαλείο γενικού σκοπού. Πολύ δημοφιλής και αποδοτικές είναι οι εφαρμογές που συνδυάζουν Γ.Α. με άλλες μεθόδους (υβριδικοί Γ.Α.), γιατί έτσι εξουδετερώνονται αμοιβαία τα μειονεκτήματά τους. Σήμερα ο αριθμός των εφαρμογών που χρησιμοποιούν Γ.Α. αυξάνει με γρήγορους ρυθμούς, πράγμα που δείχνει ότι το μέλλον των Γ.Α. είναι ευοίωνο. Ο επιστημονικός και επιχειρηματικός κόσμος έχει πειστεί ότι η δύναμη που προσφέρουν οι Γ.Α. είναι ικανή να δίνει γρήγορες και αξιόπιστες λύσεις σε προβλήματα πολύ μεγάλης δυσκολίας και πολυπλοκότητας, γι' αυτό και αναμένονται σοβαρές επενδύσεις σε κόπο και χρήματα για την παραπέρα εξέλιξή τους.

Μέσα στο ευρύτερο πλαίσιο της Τεχνητής Νοημοσύνης, οι Γ.Α. αποτελούν ένα πολλά υποσχόμενο πεδίο. Υπάρχουν προβλέψεις που υποστηρίζουν ότι το τοπίο της Τεχνητής Νοημοσύνης ίσως αλλάξει ριζικά με την χρήση Γ.Α. Τα Νευρωνικά Δίκτυα επίσης αναμένεται να γνωρίσουν μεγάλη ανάπτυξη συνδυαζόμενα με Γ.Α., ενώ γίνεται λόγος ακόμη και για μελλοντική τους ενοποίηση Ένα καυτό σημείο έρευνας αποτελεί σήμερα η υλοποίηση Γ.Α. σε παράλληλες μηχανές. Σε προηγούμενο κεφάλαιο επισημάνθηκε ότι οι Γ.Α. έχουν έντονα στοιχεία παραλληλισμού,ενώ κάποιοι ερευνητές τους χαρακτηρίζουν υψηλού επιπέδου παράλληλους αλγόριθμους (highly parallel algorithms). Από την άλλη μεριά, οι παράλληλοι υπολογιστές έχουν αρχίσει να κάνουν δειλά βήματα στην αγορά με συστήματα που περιέχουν από λίγες δεκάδες μέχρι λίγες χιλιάδες επεξεργαστές. Όλη αυτή η επεξεργαστική ισχύς μπορεί να γίνει πολύ καλά εκμεταλλεύσιμη από τους Γ.Α. Αυτό βασίζεται κυρίως στο γεγονός ότι οι Γ.Α. λειτουργούν πάνω σε πληθυσμό ατόμων, ο οποίος πλέον μπορεί να διαμοιραστεί σε αρκετούς επεξεργαστές. Έτσι, ίσως να εμπεριέχει αρκετή αλήθεια η άποψη ότι οι Γ.Α. θα βοηθήσουν την εξάπλωση και βελτίωση των επιδόσεων των παράλληλων μηχανών και το αντίστροφο. Το μέλλον θα δείξει, αλλά το σίγουρο είναι ότι οι Γ.Α. δεν πρόκειται να εξαφανιστούν, τουλάχιστον για όσο θα μπορούν να δίνουν καλύτερες λύσεις από τους ανταγωνιστές τους. Κι αυτό μάλλον θα συμβαίνει για αρκετό καιρό, αφού είναι ανοικτά για έρευνα πολλά θέματα που αναμένεται να δώσουν ακόμη πιο εντυπωσιακά αποτελέσματα και να καθιερώσουν τους Γ.Α. ως καθοριστικό εργαλείο για την περαιτέρω εξέλιξη της Επιστήμης των Υπολογιστών.

ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΤΟΥ TSP ΣΤΙΣ 50 ΜΕΓΑΛΥΤΕΡΕΣ ΠΟΛΕΙΣ ΤΗΣ ΕΛΛΑΔΑΣ

3.1 ΣΚΟΠΟΣ

Σε αυτό το κεφάλαιο για να γίνει εύκολα αντιληπτός τόσο ο ορισμός του TSP όσο και ο αλγόριθμος επίλυσης, επιλέγουμε γνωστά εμπειρικά δεδομένα, τις πενήντα μεγαλύτερες πόλεις βάση πληθυσμού στο χάρτη της Ελλάδας, και παρουσιάζουμε μερικά χαρακτηριστικά βήματα της διαδικασίας επίλυσης.

A	B	C	D	E	F	G	H
1	Αθήνα	2.769.221 [α]	472488,094	4204010,03		472000	4204000
2	Θεσσαλονίκη	789.191[β]	411655,237	4497563,45		412000	4498000
3	Πειραιάς	448.997[γ]	472337,281	4198679,99		472000	4199000
4	Πάτρα	168.202[δ]	301647,881	4235270,88		302000	4235000
5	Ηράκλειο	153.655[ε]	603749,735	3910896,36		604000	3911000
6	Λάρισα	144.651	364117,52	4388385,3		364000	4388000
7	Βόλος	125.248	408713,726	4357230,15		409000	4357000
8	Ιωάννινα	65.574	229795,9	4394922,5		230000	4395000
9	Τρίκαλα	61.653	308079,32	4380570,81		308000	4381000
10	Χαλκίδα	59.125	465387,589	4257126,17		465000	4257000
11	Σέρρες	58.287	461755,911	4547962,97		462000	4548000
12	Αλεξανδρούπολη	57.812	657786,09	4522912,96		658000	4523000
13	Ξάνθη	56.122	574361,48	4554327,4		574000	4554000
14	Κατερίνη	55.997	372271,003	4458613,79		372000	4459000
15	Αγρίνιο	55.097[ζ]	274336,212	4278042,45		274000	4278000
16	Καλαμάτα	54.100	331893,521	4100768,09		332000	4101000
17	Καβάλα	54.027	534060,992	4531623,99		534000	4532000
18	Χανιά	53.910	501483,31	3929728,38		501000	3930000
19	Λαμία	52.006	363997,598	4306503,74		364000	4307000
20	Κομοτηνή	50.990	617937,053	4552824,7		618000	4553000
21	Ρόδος	49.541	878304,97	4040112,77		878000	4040000
22	Δράμα	44.823	511693,932	4555013,47		512000	4555000
23	Βέροια	43.158	341747,055	4487655,44		342000	4488000

24	Κοζάνη	41.066	311829,456	4463199,03		312000	4463000
25	Καρδίτσα	38.554	320819,694	4358834,59		321000	4359000
26	Ρέθυμνο	32.468	543123,386	3913272,72		543000	3913000
27	Πτολεμαΐδα	32.142	303395	4487178		303000	4487000
28	Τρίπολη	30.866	356080,653	4152557,2		356000	4153000
29	Κόρινθος	30.176	405940,093	4199165,37		406000	4199000
30	Γέρακας	29.939	486872	4206879		487000	4207000
31	Γιαννιτσά	29.789	365830,37	4516713,76		366000	4517000
32	Μυτιλήνη	27.871	690351,076	4344687,29		690000	4345000
33	Χίος	26.850	686237,459	4248946,96		686000	4249000
34	Σαλαμίνα	25.370	455940,358	4201616,21		456000	4202000
35	Ελευσίνα	24.910	459338,547	4209683,37		459000	4210000
36	Κέρκυρα	24.838	149800,954	4393663,08		150000	4394000
37	Πύργος	24.359	273732,365	4172221,27		274000	4172000
38	Μέγαρα	23.456	442389,034	4205133,99		442000	4205000
39	Κιλκίς	22.914	405070	4538186		405000	4538000
40	Θήβα	22.883	440535,954	4241521,08		441000	4242000
41	Άργος	22.209	387387	4165762		387000	4166000
42	Άρτα	21.895	239206,487	4338445,76		239000	4338000
43	Άρτεμη (Λούτσα)	21.488	500688,434	4202596,82		501000	4203000
44	Λιβαδειά	21.379	401746	4254487		402000	4254000
45	Ωραιόκαστρο	20.852	408186,278	4508749,79		408000	4509000
46	Αίγιο	20.422	331952,06	4235044,73		332000	4235000
47	Κως	19.432	792876	4087800,33		793000	4088000
48	Κορωπί	19.164	488502,693	4194470,46		489000	4194000
49	Πρέβεζα	19.042	218635,422	4316809,74		219000	4317000
50	Νάουσα	18.882	336380	4499130		336000	4499000

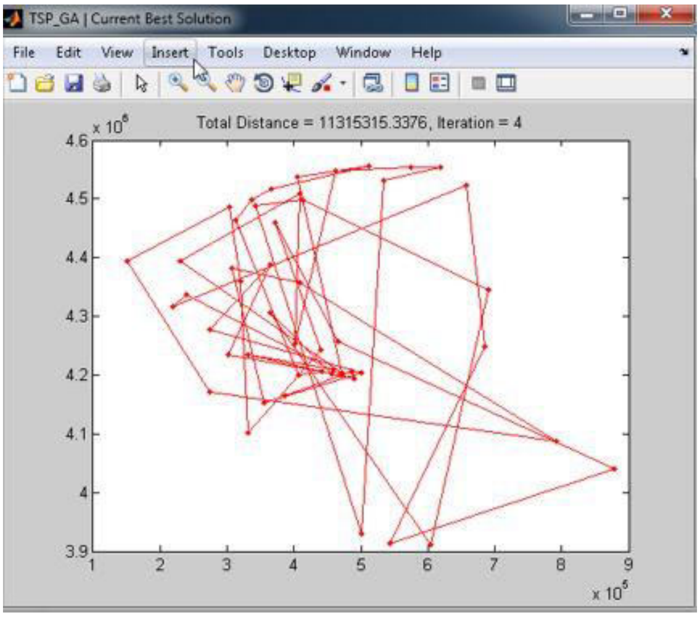
3.1 Οι πενήντα μεγαλύτερες πόλεις της Ελλάδας βάση πληθυσμού και οι συντεταγμένες (x,y)

3.2 ΣΥΝΘΗΚΕΣ- ΠΑΡΑΔΟΧΕΣ

Για να διαπιστώσουμε ότι ο αλγόριθμος του πλανόδιου πωλητή λειτουργεί και στη πράξη χρησιμοποιήσαμε ως δεδομένο τις πόλεις. Με τη χρήση του εθνικού κτηματολογίου, βρήκαμε τις συντεταγμένες (x,y) της κάθε πόλης και αυτές τις συνταγμένες τις τοποθετήσαμε στο κώδικα του tsp. Σε κάθε πόλη, οι συντεταγμένες υπολογίστηκαν σε ένα κεντρικό σημείο της πόλης όπως ένα λιμάνι, ένα δημαρχείο κλπ. Επίσης, θεωρήσαμε ως δεδομένο ότι ο πλανόδιος πωλητής θα μετακινείται από τη μια πόλη στην άλλη με ελικόπτερο. Ο λόγος είναι ότι με αυτή την παραδοχή εξυπηρετούμε καλύτερα το σκοπό επίλυσης του TSP στον χάρτη με τις πενήντα μεγαλύτερες πόλεις της Ελλάδας καθώς ορίζοντας τις πόλεις ως απλά σημεία στο καρτεσιανό σύστημα του Εθνικού Κτηματολογίου το αντιληπτό οπτικά μήκος της γραμμής που θα της ενώνει γραφικά αντιστοιχεί στην απόσταση που θα διένυε ο πωλητής με ελικόπτερο.

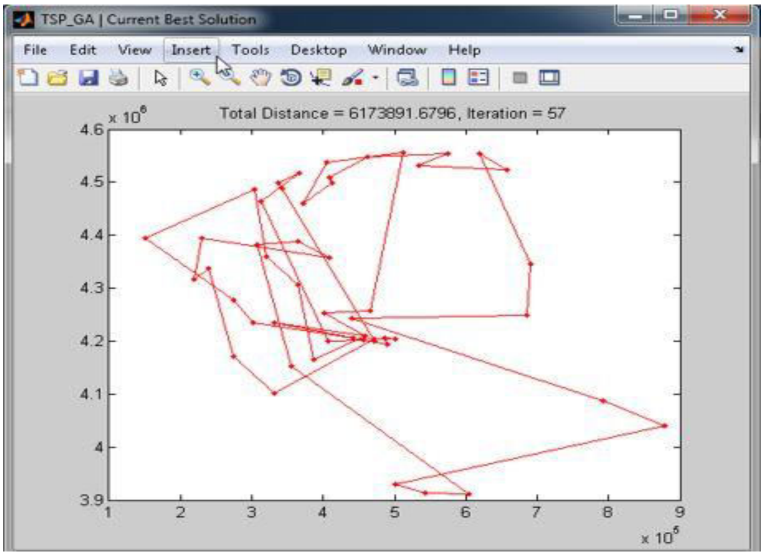
3.3 ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Η βέλτιστη λύση που έχει προκύψει μετά από 4 επαναλήψεις, δηλαδή μετά από 4 γενιές λύσεων, παρουσιάζεται στην εικόνα 3.2



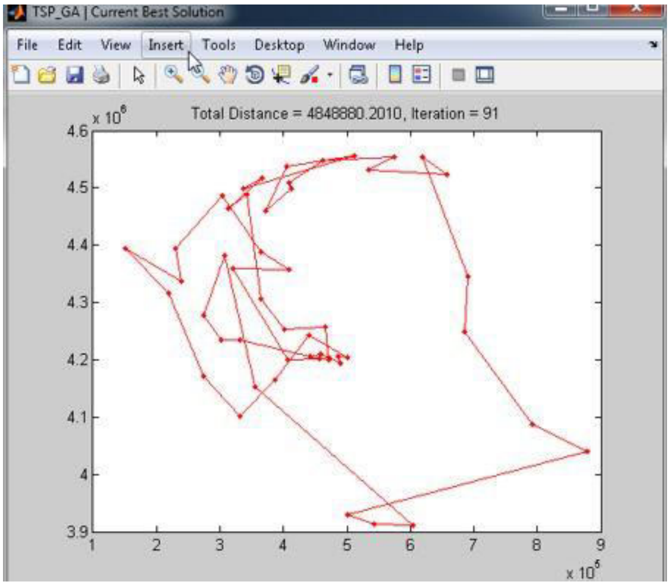
Εικόνα 3.2. Καλύτερη λύση μετά από 4 επαναλήψεις

Στην εικόνα 4.3 παρουσιάζεται η βέλτιστη λύση που προκύπτει μετά από 57 γενιές γενετικών λύσεων.



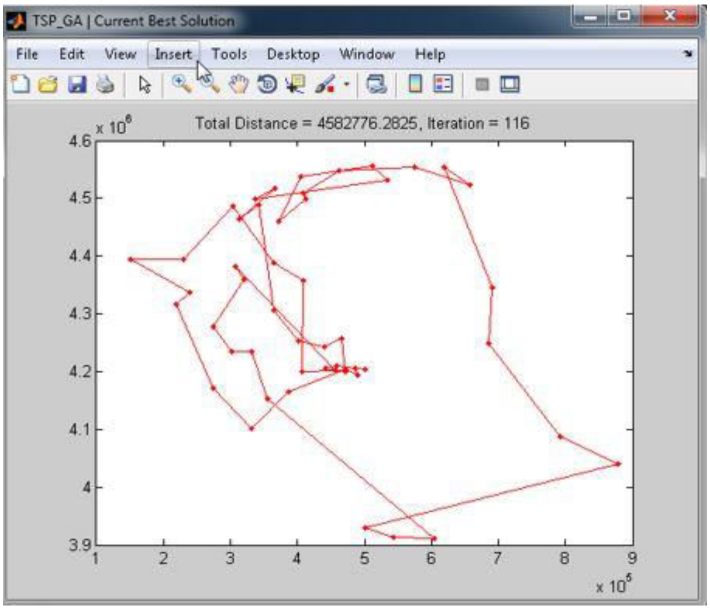
Εικόνα 3.3. Καλύτερη λύση μετά από 57 επαναλήψεις

Στην 4.4 βλέπουμε μετά από 91 γενιές λύσεων, τη βέλτιστη λύση.



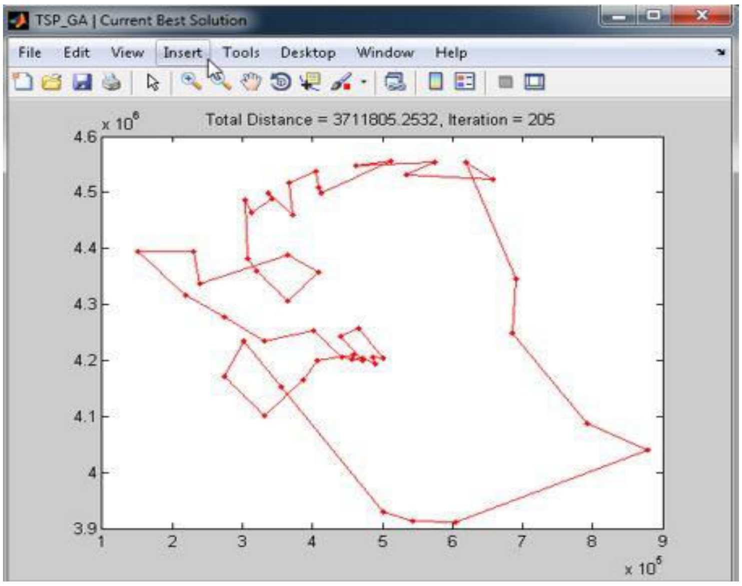
Εικόνα 3.4. Καλύτερη λύση μετά από 91 επαναλήψεις

Η βέλτιστη λύση που έχει προκύψει μετά από 116 επαναλήψεις, δηλαδή μετά από 116 γενιές λύσεων, παρουσιάζεται στην εικόνα 3.5



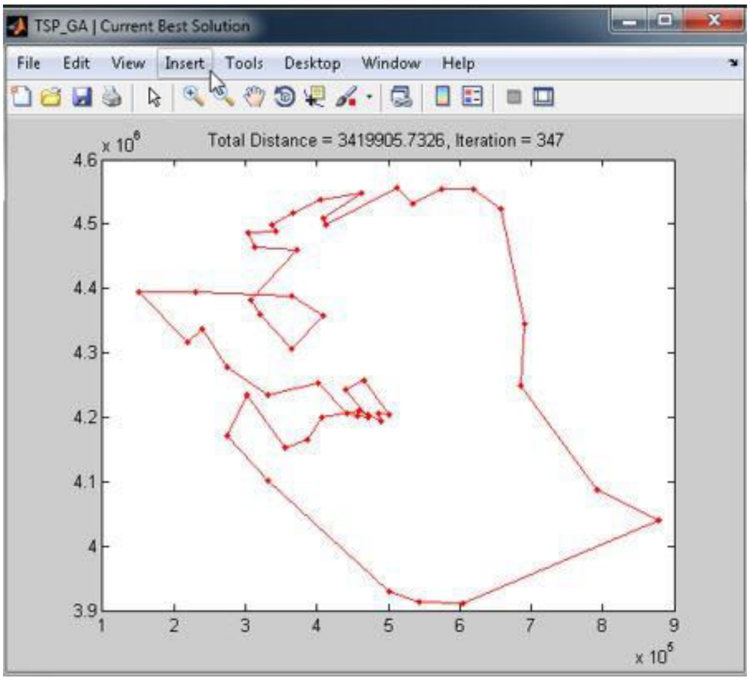
Εικόνα 3.5. Καλύτερη λύση μετά από 116 επαναλήψεις

Στην εικόνα 3.6 παρουσιάζεται η βέλτιστη λύση που προκύπτει μετά από 205 γενιές λύσεων.

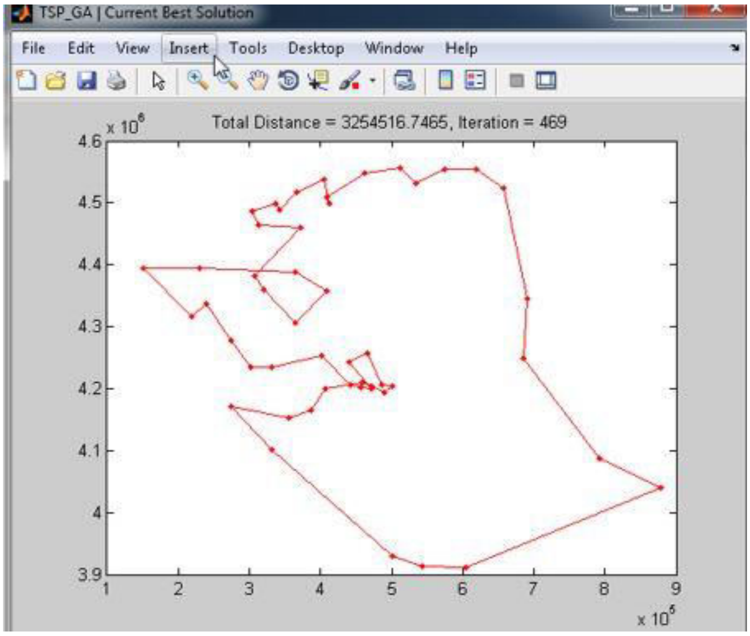


Εικόνα 3.6.: Καλύτερη λύση μετά από 205 επαναλήψεις

Στην 3.7 βλέπουμε μετά από 347 γενιές λύσεων, τη βέλτιστη λύση.

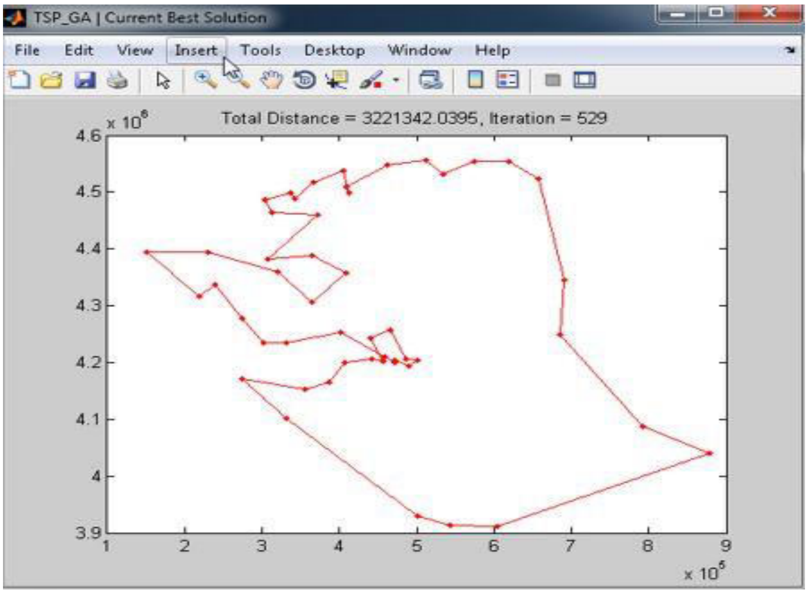


Εικόνα 4.7. Καλύτερη λύση μετά από 347 επαναλήψεις

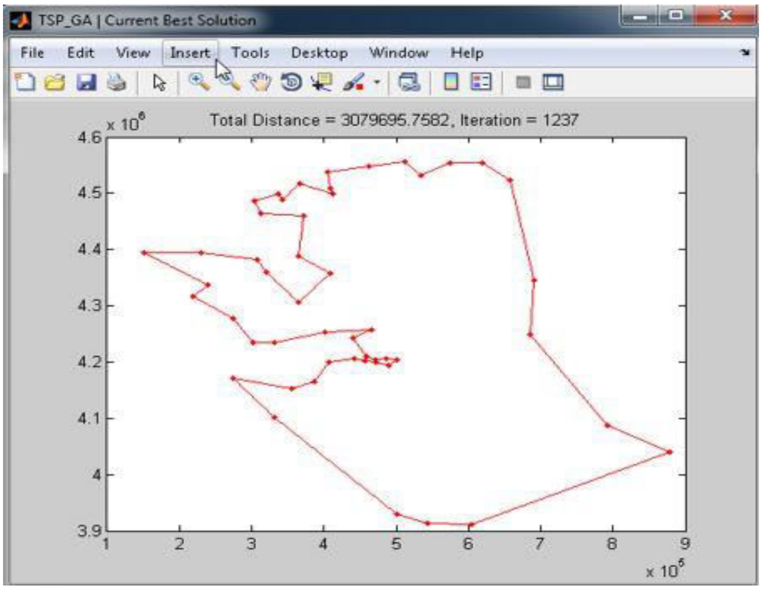


Εικόνα 3.8. Καλύτερη λύση μετά από 469 επαναλήψεις

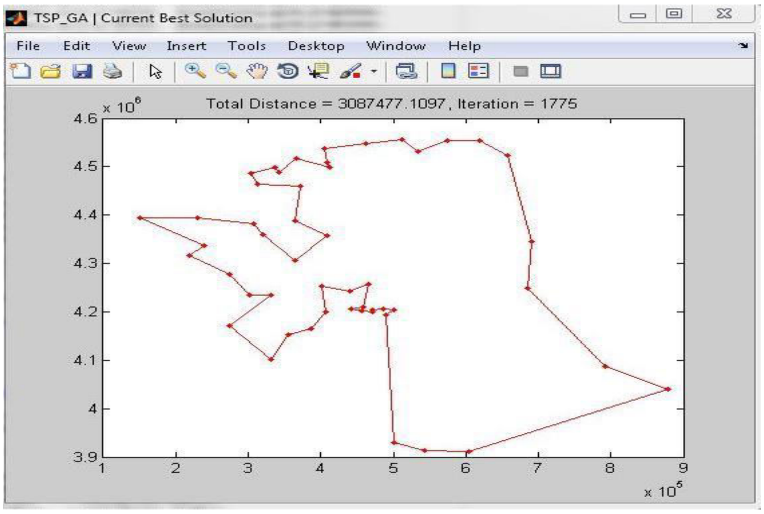
Στην εικόνα 3.9 παρουσιάζεται η βέλτιστη λύση που προκύπτει μετά από 529 γενιές γενετικών λύσεων.



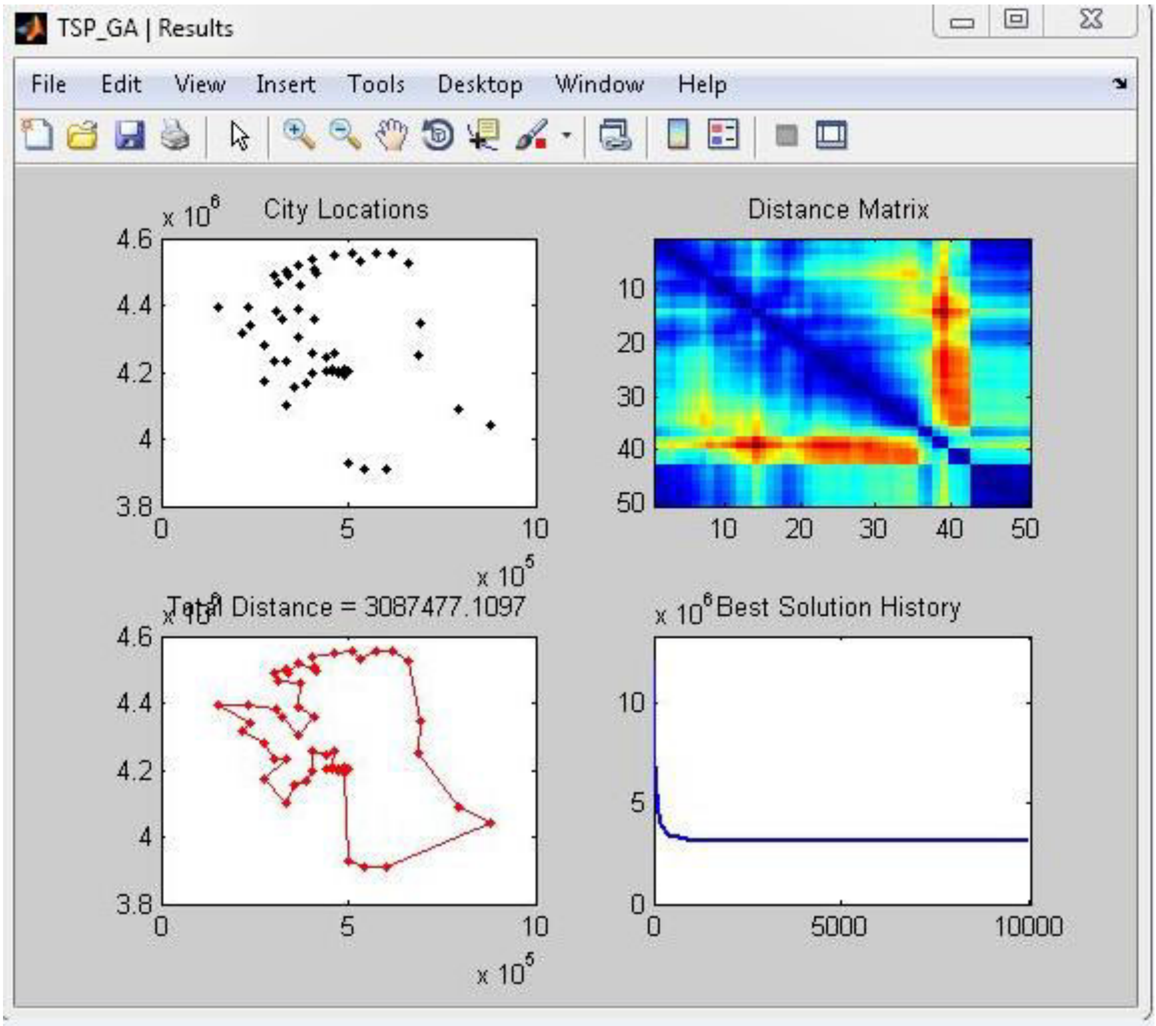
Εικόνα 3.9. Η καλυτερη λυση μετα από 529 επαναληψεις



Εικόνα 3.10. Καλύτερη λύση μετά από 1237 γενιές.



Εικόνα 3.11. Καλύτερη λύση μετά από 1775 επαναλήψεις



Εικόνα 3.12. Συγκεντρωτική παρουσίαση χαρακτηριστικών στοιχείων της διαδικασίας επίλυσης. Πάνω-αριστερά: Οι 50 μεγαλύτερες πόλεις της Ελλάδας σε καρτεσιανό σύστημα συντεταγμένων. Πάνω δεξιά: Πίνακας αποστάσεων των 50 πόλεων με μορφοποίηση «θερμοκρασίας». Κάτω-αριστερά: Καλύτερη λύση μετά από 1775 επαναλήψεις (γενιές λύσεων). Κάτω-δεξιά: Εξέλιξη της αντικειμενικής συνάρτησης μεταξύ διαδοχικών επαναλήψεων

3.4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Μέσα από τη επίλυση αυτό του προβλήματος επιβεβαιώθηκε η χρηστικότητα του TSP ,η αξιοπιστία και η λειτουργικότητα του αλγορίθμου.

ΚΕΦΑΛΑΙΟ 4:DISC SCHEDULING PROBLEM (DSP)

Σε αυτό το κεφάλαιο επιλέγεται μία ειδική εφαρμογή του TSP στο χώρο του υλικού υπολογιστών, το πρόβλημα προγραμματισμού της βέλτιστης ανάγνωσης δεδομένων από σκληρό δίσκο. Στην διεθνή βιβλιογραφία είναι γνωστό ως disk scheduling problem (DSP). Το πρόβλημα αυτό ταξινομείται ως μη συμμετρικό πρόβλημα περιοδεύοντος πωλητή. Για την καλύτερη κατανόηση του DSP παρουσιάζεται ο τρόπος ανάγνωσης δεδομένων ενός σκληρού δίσκου από την κεφαλή ανάγνωσης και οι αντίστοιχες αναλυτικές μαθηματικές εκφράσεις κρίσιμων μεγεθών.

4.1 ΑΙΤΗΜΑΤΑ

Αρχικά ορίζονται τα αιτήματα (requests) που δέχεται ένας σκληρός δίσκος ως $i,j \in N$, όπου το N υποδηλώνει το σύνολο των φυσικών αριθμών. Ένα request δεν είναι απαραίτητα ένα ακέραιο αρχείο αλλά ενδεχομένως και τμήμα αυτού (π.χ. fragmented file). Μεταξύ δύο οποιαδήποτε αιτημάτων i, j και συγκεκριμένα από το τέλος ανάγνωσης του i έως την έναρξη ανάγνωσης του j ορίζεται απαιτούμενος συνολικός χρόνος κίνησης actuator arm και platter c[i,j](σε msec).

4.2. ΧΡΟΝΟΣ ΚΙΝΗΣΗΣ ΒΕΛΟΝΑΣ

Προκειμένου να υπολογιστεί ο χρόνος c[i,j] εντοπίζονται τα ακριβή track-sector έναρξης και πέρατος του αιτήματος πάνω στον σκληρό δίσκο. Τα disk sectors ή απλά sectors περιγράφονται από τη γωνία Φi (degrees) επί του δίσκου εγγραφής (platter) και τα track από την παράμετρο Ri (ενδεικτικό της ακτίνας του platter). Οι πληροφορίες αυτές προέρχονται από το FAT (File Allocation Table) και στο σύνολο ορίζουν ένα οποιαδήποτε track-sector. Ως παραμέτρους που περιγράφουν το track-sector έναρξης ενός request i ορίζουμε τα ΦSi (Φ-start) και το RSi (R-start) και αντίστοιχα για το track-sector πέρατος της εγγραφής του ίδιου request i τα ΦEi (Φ-End) και REi (R-End). Η αρχική θέση του actuator arm ορίζεται ως BRS και του δίσκου DΦS και ως τελική θέση βελόνας και δίσκου ορίζονται τα BRE και DRE αντίστοιχα. Τα BRS και DΦS έχουν πραγματικές τιμές ενώ τα BRE και DΦE είναι εικονικά προκειμένου να αποδώσουν τη τελική θέση βελόνας, δίσκου κατόπιν ολοκλήρωσης του τελευταίου αιτήματος. Γι'αυτό το χρονικό κόστος από το τέλος οποιουδήποτε αιτήματος προς τη τελική θέση της βελόνας ορίζεται ως μηδέν (0).

Ο χρόνος κίνησης του actuator arm speed κατά την ακτίνα του δίσκου (seek time) είναι

$$TRij = | RSj – REi | / s1 \qquad \textbf{(1)}$$

, όπου i,j ένα αρχικό και τελικό αίτημα αντίστοιχα και s1 η ταχύτητα του actuator arm

σε (track/msec). Η ταχύτητα s1 υπολογίζεται αριθμητικά με average read seek time = 8.5msec => full seek time = 3*8.5msec = 25.5msec tracks per inch (TPI) 340,000. Αρά σε πραγματική εγγράψιμη ακτίνα δίσκου περί τα 3.8 cm εχουμε 3.8/2.54= 1.49 inch => 340,000 tpi * 1.49 = 500.000 tracks τα οποία τα διανύει σε 25.5msec full seek time άρα έχει s1= 500.000 tracks /25.5msec = περίπου με 20.000 track/msec.

4.3.ΤΑΧΥΤΗΤΑ ΣΤΡΟΦΗΣ ΔΙΣΚΟΥ

Η ταχύτητα στροφής του δίσκου (platter speed) *s2 υπολογίζεται ως εξής:*

s2=7200RPM => s2=7200 RPM 360degrees / 60 min/sec => s2 = 43200 degrees/sec* ή **0,12 rotations/msec**.

Κατά το χρόνο κίνησης της βελόνας TR_{ij} από το request *i* στο request *j* ο δίσκος έχει γυρίσει και βρίσκεται πλέον στη γωνία $ΦTR_{ij}$. Συνεπώς έχουμε

$ΦTR_{ij} = ΦE_i + TR_{ij} * s2$ **(2)**

Από τα **(1), (2)** προκύπτει :

$ΦTR_{ij} = ΦE_i + | RS_j – RE_i | * s2 / s1$. Τέλος η βελόνα παραμένει σταθερή στο RS_j και ο δίσκος γυρίζει μέχρι την $ΦS_j$ κατά πρόσθετο χρόνο $TΦ_{ij}$ ο οποίος είναι :

$TΦ_{ij} = ΔΦ_{ij} / s2$ **(3)**

όπου ΔΦ = 1. $ΦS_j – ΦTR_{ij}$, $ΦS_j \geq ΦTR_{ij}$

2. $(ΦS_j + 360) - ΦTR_{ij}$, $ΦS_j < ΦTR_{ij}$ **(4)**

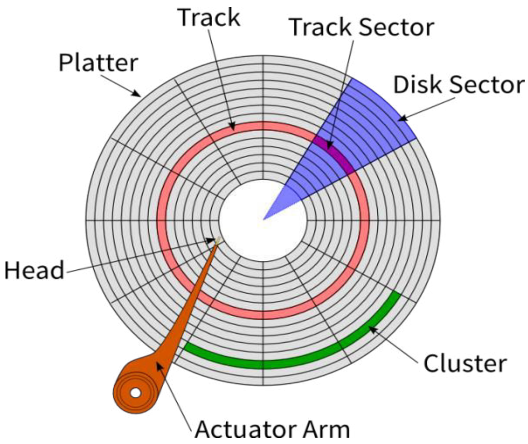
4.4 ΤΕΛΙΚΗ ΣΥΝΑΡΤΗΣΗ

Συνολικά έχουμε:

$C_{ij} = TR_{ij} + TΦ_{ij}$ **(5)**

4.5 ΠΑΡΑΔΟΧΕΣ

Πριν κλείσει αυτό το κεφάλαιο καλό θα ήταν να επισημάνουμε ότι ο χρόνος ανάγνωσης ενός αιτήματος δεν είναι μέρος του προβλήματος που εξετάζουμε και συνεπώς δεν εμφανίζεται ως παράμετρος ή και μεταβλητή απόφασης όπως και στο τυπικό TSP (Travelling Salesman Problem).



Εικόνα 4.1. :Σκληρός δίσκος HDD

<u>**ΚΕΦΑΛΑΙΟ 5: ΑΡΙΘΜΗΤΙΚΗ ΕΦΑΡΜΟΓΗ ΤΟΥ DISK SCHEDULING PROBLEM (DSP)**</u>

5.1 ΣΚΟΠΟΣ

Στο παρόν κεφάλαιο υλοποιείται ένα πλήθος δοκιμαστικών αριθμητικών δεδομένων που έχουν εκτιμηθεί με συμβατικά τεχνικά στοιχεία ενός τυπικού εμπορικού σκληρού δίσκου (Seagate Barracuda 3TB 7200RPM).

5.2 ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΩΔΙΚΑ ΜΑΤ LAB ΚΑΙ ΠΕΡΙΓΡΑΦΗ

```
        % Find the Best Route in the Population
        [minDist,index] = min(totalDist);
        distHistory(iter) = minDist;
        if minDist < globalMin
            globalMin = minDist;
            optRoute = pop(index,:);
            if showProg
                % Plot the Best Route
                rte = optRoute([1:n 1]);
                if dims > 2, plot3(hAx,xy(rte,1),xy(rte,2),xy(rte,3),'r.-');
                else plot(hAx,xy(rte,1),xy(rte,2),'r.-'); end
                title(hAx,sprintf('Total Distance = %1.4f, Iteration = %d',minDist,iter));
                drawnow;
            end
        end
```

<u>Εικόνα 5.1 :</u> Εύρεση καλύτερης διαδρομής από το σύνολο των γενεών λύσεων και εκτύπωση αυτής.

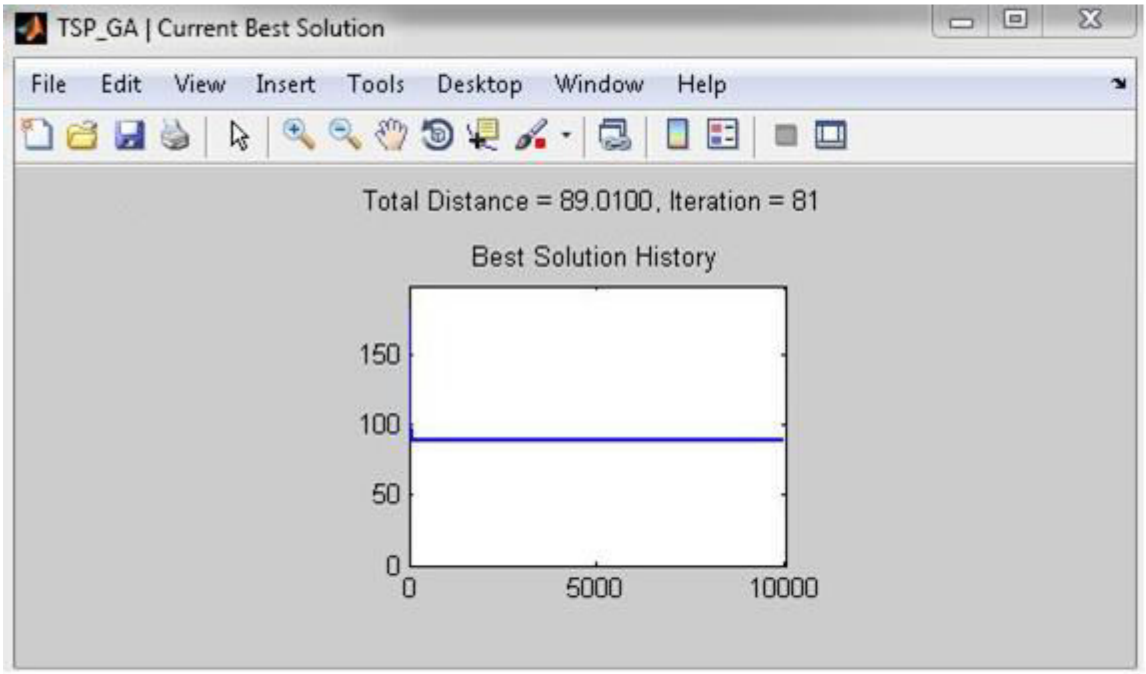
```
% Cij = TRij  + T[illegible]ij

dmat(1,1)=0.00; dmat(1,2)=4.61; dmat(1,3)=12.49; dmat(1,4)=7.57; dmat(1,5)=7.70; dmat(1,6)=5.86; dmat(1,7)=9.49; dmat(1,8)=11.00; dmat(1,9)=5.01; dmat(1,10)=10.68; dmat(1,11)=2.58; dmat(1,12)=19.53; dmat(1,13)=14.91; dmat(1,
dmat(2,1)=4.26; dmat(2,2)=0.54; dmat(2,3)=8.42; dmat(2,4)=11.85; dmat(2,5)=11.97; dmat(2,6)=1.80; dmat(2,7)=13.75; dmat(2,8)=15.36; dmat(2,9)=0.95; dmat(2,10)=14.96; dmat(2,11)=6.85; dmat(2,12)=15.47; dmat(2,13)=19.19;
dmat(3,1)=15.91; dmat(3,2)=12.18; dmat(3,3)=3.41; dmat(3,4)=23.50; dmat(3,5)=23.62; dmat(3,6)=13.45; dmat(3,7)=17.07; dmat(3,8)=2.01; dmat(3,9)=12.60; dmat(3,10)=1.61; dmat(3,11)=10.17; dmat(3,12)=10.46; dmat(3,
dmat(4,1)=10.39; dmat(4,2)=14.99; dmat(4,3)=22.87; dmat(4,4)=1.30; dmat(4,5)=1.42; dmat(4,6)=7.92; dmat(4,7)=3.21; dmat(4,8)=21.48; dmat(4,9)=7.07; dmat(4,10)=21.07; dmat(4,11)=12.97; dmat(4,12)=21.59; dmat(4,13)=25.30;
dmat(5,1)=13.19; dmat(5,2)=9.46; dmat(5,3)=17.34; dmat(5,4)=4.00; dmat(5,5)=4.23; dmat(5,6)=10.72; dmat(5,7)=6.00; dmat(5,8)=15.95; dmat(5,9)=9.86; dmat(5,10)=15.55; dmat(5,11)=15.76; dmat(5,12)=24.39; dmat(5,13)=19.78;
dmat(6,1)=2.37; dmat(6,2)=6.97; dmat(6,3)=14.85; dmat(6,4)=9.93; dmat(6,5)=10.07; dmat(6,6)=8.23; dmat(6,7)=11.85; dmat(6,8)=13.47; dmat(6,9)=7.37; dmat(6,10)=13.05; dmat(6,11)=4.95; dmat(6,12)=13.56; dmat(6,13)=17.28;
dmat(7,1)=7.88; dmat(7,2)=12.49; dmat(7,3)=20.37; dmat(7,4)=7.12; dmat(7,5)=7.25; dmat(7,6)=13.75; dmat(7,7)=0.70; dmat(7,8)=18.98; dmat(7,9)=12.90; dmat(7,10)=18.57; dmat(7,11)=10.47; dmat(7,12)=19.09; dmat(7,13)=22.8
dmat(8,1)=14.97; dmat(8,2)=11.24; dmat(8,3)=2.45; dmat(8,4)=22.55; dmat(8,5)=22.67; dmat(8,6)=12.50; dmat(8,7)=16.12; dmat(8,8)=1.07; dmat(8,9)=11.65; dmat(8,10)=0.66; dmat(8,11)=9.22; dmat(8,12)=9.50; dmat(8,
dmat(9,1)=4.12; dmat(9,2)=8.72; dmat(9,3)=16.60; dmat(9,4)=11.68; dmat(9,5)=11.82; dmat(9,6)=1.63; dmat(9,7)=13.60; dmat(9,8)=15.22; dmat(9,9)=0.80; dmat(9,10)=14.80; dmat(9,11)=6.70; dmat(9,12)=15.31; dmat(9,13)=19.0
dmat(10,1)=12.25; dmat(10,2)=8.52; dmat(10,3)=8.07; dmat(10,4)=19.82; dmat(10,5)=19.96; dmat(10,6)=9.77; dmat(10,7)=21.73; dmat(10,8)=6.69; dmat(10,9)=8.93; dmat(10,10)=6.28; dmat(10,11)=14.84; dmat(10,12)=6.79;
dmat(11,1)=9.48; dmat(11,2)=5.75; dmat(11,3)=13.64; dmat(11,4)=17.05; dmat(11,5)=8.85; dmat(11,6)=7.00; dmat(11,7)=10.64; dmat(11,8)=12.25; dmat(11,9)=6.17; dmat(11,10)=11.84; dmat(11,11)=3.72; dmat(11,12)=12.35;
dmat(12,1)=19.46; dmat(12,2)=15.74; dmat(12,3)=6.95; dmat(12,4)=27.05; dmat(12,5)=18.84; dmat(12,6)=17.00; dmat(12,7)=20.62; dmat(12,8)=5.57; dmat(12,9)=16.15; dmat(12,10)=5.16; dmat(12,11)=13.72; dmat(12,12)=5.67;
dmat(13,1)=18.17; dmat(13,2)=14.44; dmat(13,3)=5.65; dmat(13,4)=25.75; dmat(13,5)=25.86; dmat(13,6)=15.69; dmat(13,7)=19.32; dmat(13,8)=4.26; dmat(13,9)=14.85; dmat(13,10)=3.86; dmat(13,11)=12.42; dmat(13,12)=4.37;
dmat(14,1)=12.26; dmat(14,2)=8.53; dmat(14,3)=8.09; dmat(14,4)=19.85; dmat(14,5)=19.96; dmat(14,6)=9.80; dmat(14,7)=21.76; dmat(14,8)=6.70; dmat(14,9)=8.95; dmat(14,10)=6.28; dmat(14,11)=6.51; dmat(14,12)=6.80;
dmat(15,1)=10.21; dmat(15,2)=14.81; dmat(15,3)=22.60; dmat(15,4)=9.46; dmat(15,5)=9.57; dmat(15,6)=16.06; dmat(15,7)=3.03; dmat(15,8)=21.30; dmat(15,9)=15.23; dmat(15,10)=20.80; dmat(15,11)=12.80; dmat(15,12)=21.42;
dmat(16,1)=6.99; dmat(16,2)=3.25; dmat(16,3)=11.14; dmat(16,4)=14.56; dmat(16,5)=14.60; dmat(16,6)=4.51; dmat(16,7)=8.15; dmat(16,8)=9.75; dmat(16,9)=3.67; dmat(16,10)=9.35; dmat(16,11)=1.24; dmat(16,12)=18.19;
dmat(17,1)=7.34; dmat(17,2)=3.61; dmat(17,3)=11.49; dmat(17,4)=6.59; dmat(17,5)=6.70; dmat(17,6)=4.87; dmat(17,7)=8.50; dmat(17,8)=10.10; dmat(17,9)=4.01; dmat(17,10)=18.03; dmat(17,11)=9.92; dmat(17,12)=18.55;
dmat(18,1)=9.10; dmat(18,2)=13.80; dmat(18,3)=21.69; dmat(18,4)=8.45; dmat(18,5)=8.57; dmat(18,6)=15.05; dmat(18,7)=10.35; dmat(18,8)=20.30; dmat(18,9)=14.22; dmat(18,10)=19.89; dmat(18,11)=11.78; dmat(18,12)=20.40;
dmat(19,1)=13.05; dmat(19,2)=9.34; dmat(19,3)=8.88; dmat(19,4)=20.64; dmat(19,5)=20.76; dmat(19,6)=10.59; dmat(19,7)=14.22; dmat(19,8)=7.50; dmat(19,9)=9.75; dmat(19,10)=7.09; dmat(19,11)=7.30; dmat(19,12)=7.60;
dmat(20,1)=5.55; dmat(20,2)=10.15; dmat(20,3)=18.03; dmat(20,4)=4.80; dmat(20,5)=4.92; dmat(20,6)=3.08; dmat(20,7)=6.72; dmat(20,8)=16.65; dmat(20,9)=10.57; dmat(20,10)=16.25; dmat(20,11)=8.14; dmat(20,12)=16.76;
```

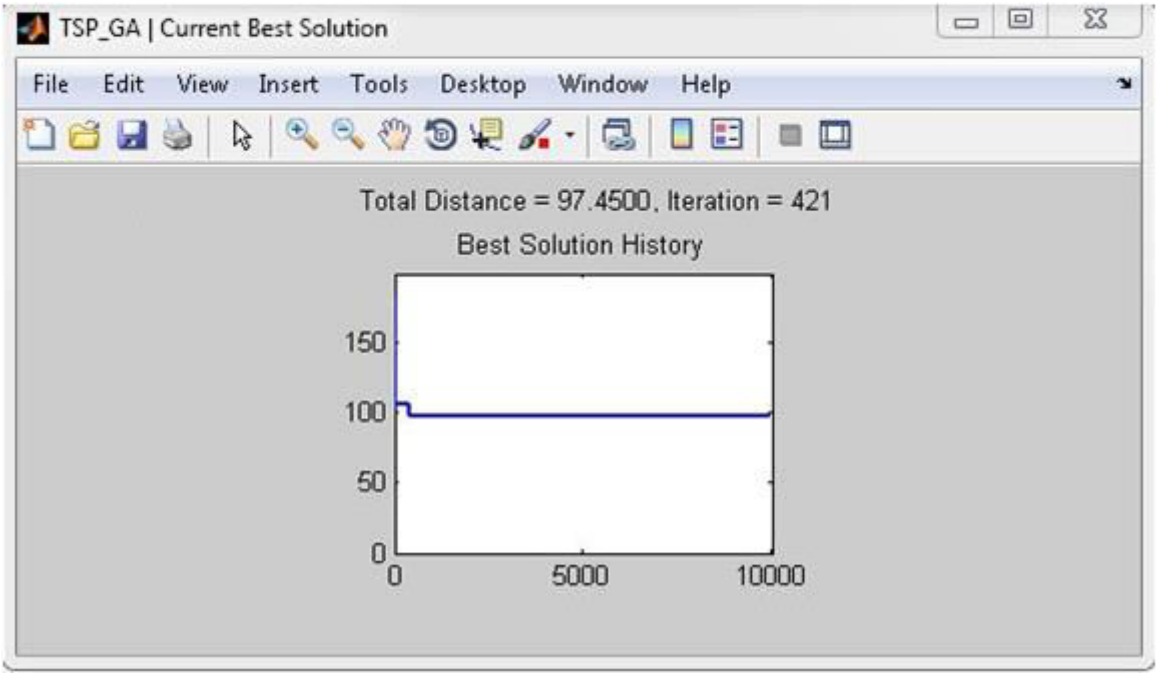
<u>Εικόνα 5.2 :</u> Αρχικοποίηση πίνακα 20x20 αποστάσεων .

```
% Genetic Algorithm Operators
randomOrder = randperm(popSize);
for p = 4:4:popSize
    rtes = pop(randomOrder(p-3:p),:);
    dists = totalDist(randomOrder(p-3:p));
    [ignore,idx] = min(dists); %#ok
    bestOf4Route = rtes(idx,:);
    routeInsertionPoints = sort(ceil(n*rand(1,2)));
    I = routeInsertionPoints(1);
    J = routeInsertionPoints(2);
    for k = 1:4 % Mutate the Best to get Three New Routes
        tmpPop(k,:) = bestOf4Route;
        switch k
            case 2 % Flip
                tmpPop(k,I:J) = tmpPop(k,J:-1:I);
            case 3 % Swap
                tmpPop(k,[I J]) = tmpPop(k,[J I]);
            case 4 % Slide
                tmpPop(k,I:J) = tmpPop(k,[I+1:J I]);
            otherwise % Do Nothing
        end
    end
    newPop(p-3:p,:) = tmpPop;
end
```

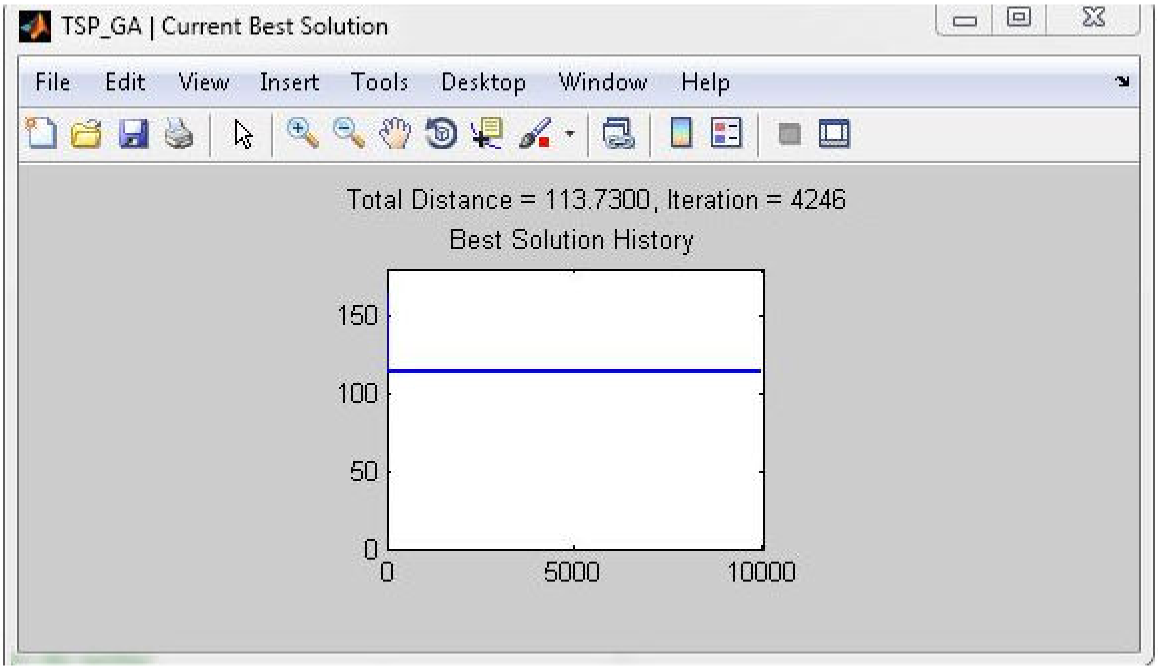
Εικόνα 5.3: Κώδικας παραγωγής «μεταλλαγμένων» λύσεων με τις μεθόδους flip, swap, slide.



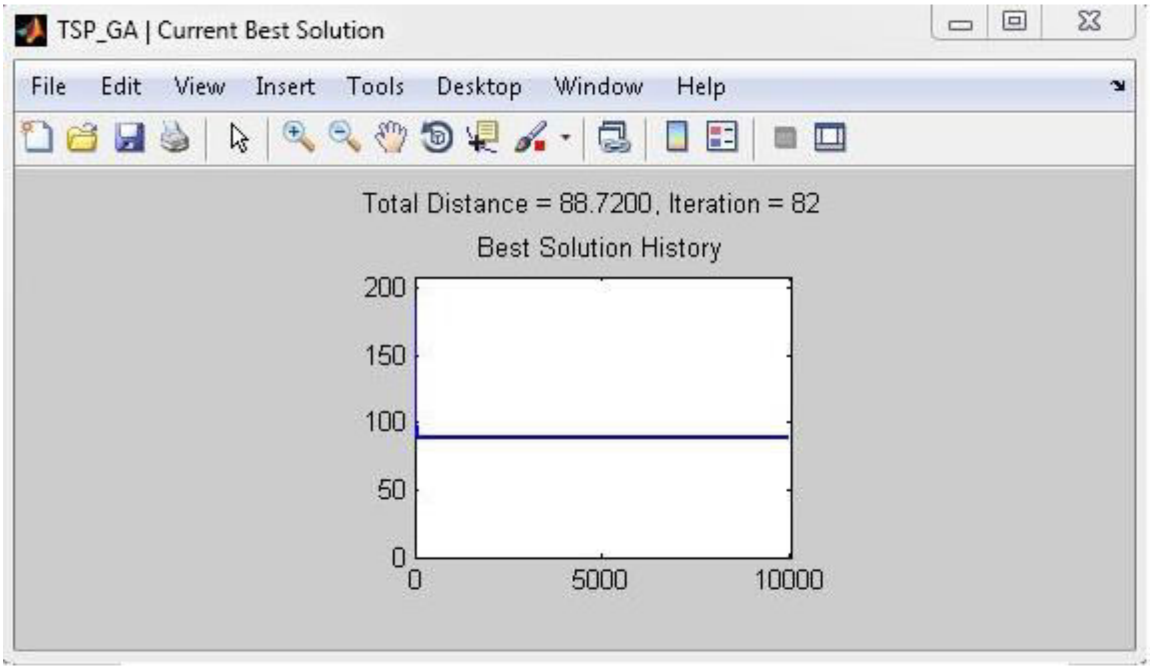
Εικόνα 5.4: Αποτελέσματα με τη μέθοδο «μετάλλαξης» Flip μόνο.



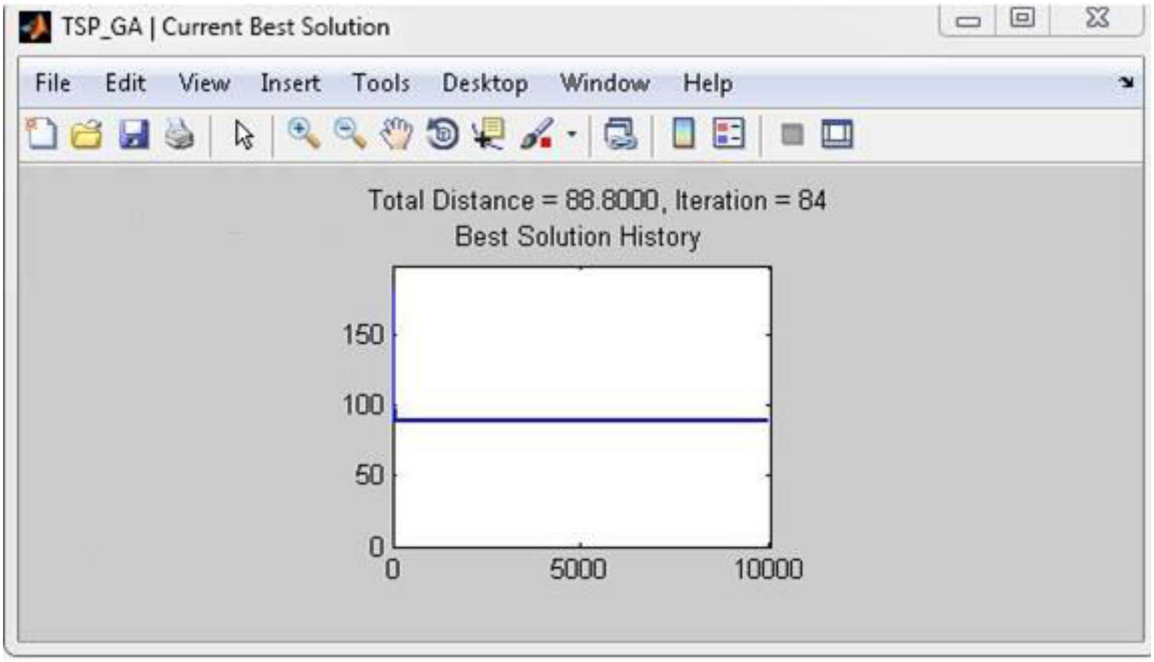
<u>Εικόνα 5.5:</u> Ααποτελέσματα «μετάλλαξης» με τη μέθοδο Slide μόνο.



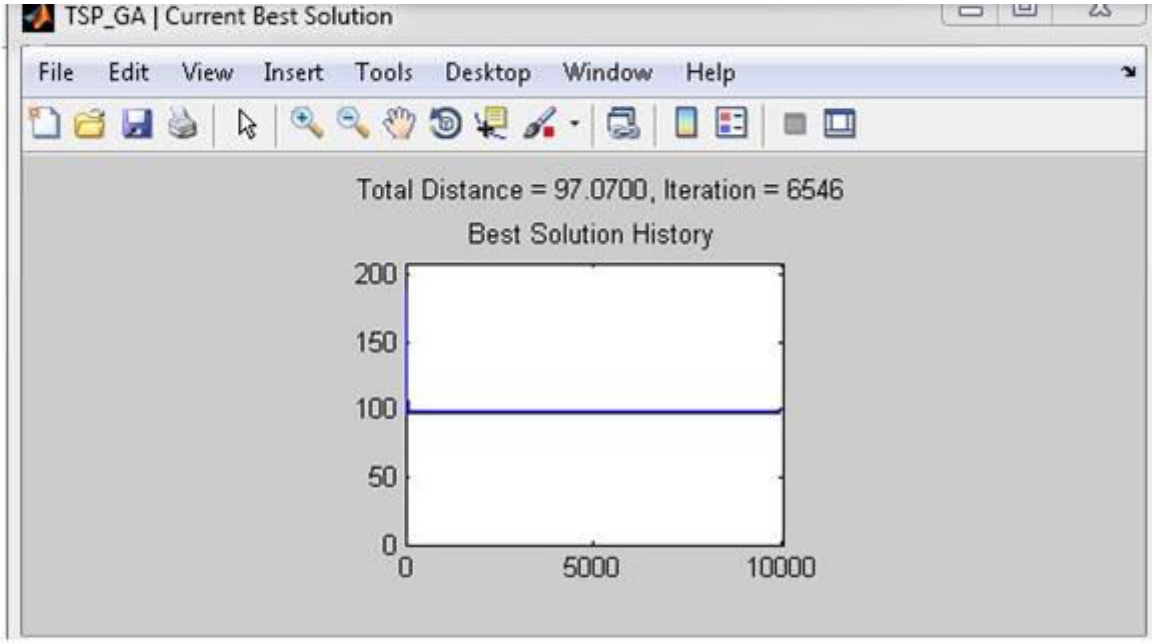
<u>Εικόνα 5.6:</u> Αποτελέσματα «μετάλλαξης» με τη μέθοδο Swap μόνο.



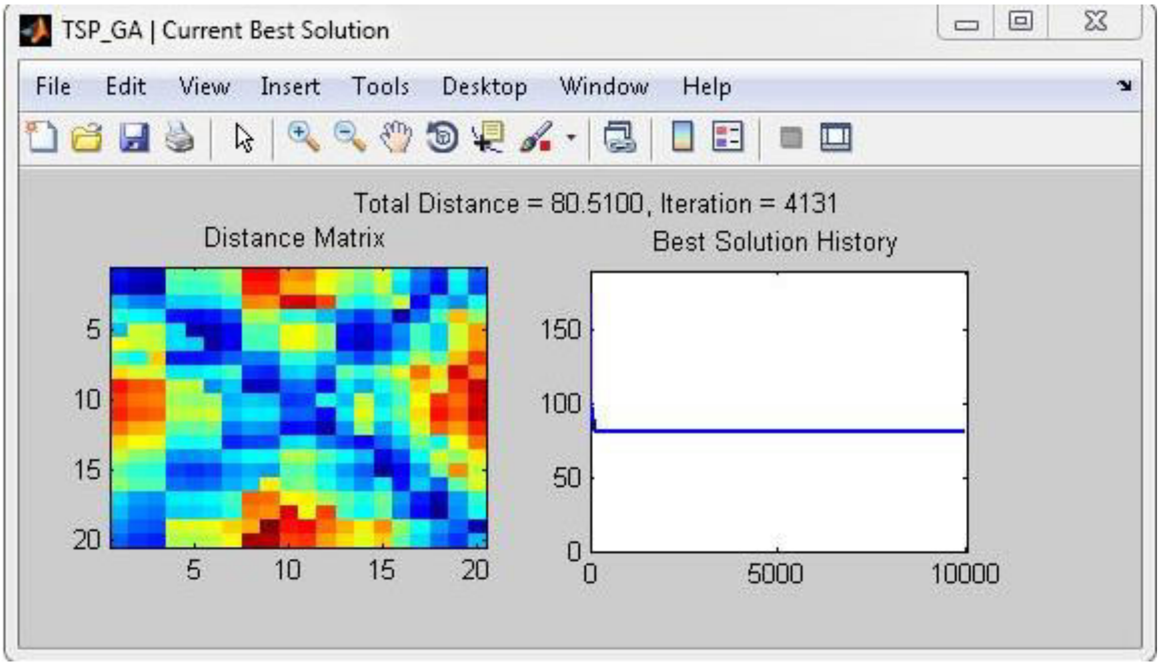
Εικόνα 5.7 : Αποτελέσματα «μετάλλαξης χωρίς τη μέθοδο flip



Εικόνα 5.8 : Αποτελέσματα «μετάλλαξης» χωρίς τη μέθοδο swap



Εικόνα 5.9 : Αποτελέσματα «μετάλλαξης» χωρίς τη μέθοδο slide



Εικόνα 5.10: Αποτελέσματα με όλες τις μεθόδους

5.3 ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο παρόν Κεφάλαιο υλοποιήθηκε μία σειρά υπολογιστικών πειραμάτων για το DSP με την χρήση κώδικα γενετικών αλγορίθμων σε προγραμματιστικό περιβάλλον Matlab.

Οι αρχικοποίηση των παραμέτρων έγινε με αριθμητικές τιμές που είναι ενδεικτικές της πραγματικής συμπεριφοράς του μαθηματικού προβλήματος, όπως αυτές τεκμηριώθηκαν στο Κεφάλαιο 4.

Για την ανεύρεση νέων «μεταλλαγμένων» γενεών λύσεων χρησιμοποιήθηκαν τρεις μέθοδοι «μετάλλαξης», οι flip, swap και slide.

Κάθε μία από αυτές υλοποιήθηκε μία φορά ανεξάρτητη, κατόπιν υλοποιήθηκαν σε συνδυασμό με μία ακόμα και τέλος και οι τρεις μαζί.

Από τα αριθμητικά αποτελέσματα παρατηρούμε ότι στις μεθόδους μου χρησιμοποιήθηκαν η κάθε μια μόνη της, καλύτερη και πιο αποδοτική φάνηκε να είναι η μέθοδος flip διότι στις 81 επαναλήψεις βρέθηκε η βέλτιστη λύση σε σχέση με τις άλλες δυο μεθόδους. Όσον αφορά το συνδυασμό αυτών των μεθόδων ο πιο αποδοτικός ήταν ο συνδυασμός slide swap , καθώς βρήκε τη λύση στις 81 επαναλήψεις.

ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα διπλωματική, εκκινώντας από το μαθηματικό πεδίο των δικτύων (θεωρία γράφων), και πιο συγκεκριμένα από το πρόβλημα του περιοδεύοντος πωλητή (Traveling Salesman Problem - TSP), εξετάσαμε πως επιτυγχάνεται η εύρεση μίας ικανοποιητικής ποιότητας λύσης με τη βοήθεια των γενετικών αλγορίθμων.

Μελετήσαμε το συμμετρικό TSP, στις πενήντα μεγαλύτερες πληθυσμιακά πόλεις της Ελλάδας, και το μη συμμετρικό TSP, στη μορφή του προβλήματος ανάγνωσης δεδομένων από ένα σκληρό δίσκο (Disk Scheduling Problem - DSP) κατά τρόπο ώστε να εμπίπτει στο TSP.

Επιλύωντας το TSP με τη βοήθεια των γενετικών αλγορίθμων διαπιστώνουμε πως σε ένα ιδιαίτερα υπολογιστικά πολύπλοκο και απαιτητικό πρόβλημα (NP-Hard), βρίσκουμε μια ικανοποιητικής ποιότητας λύση σε σύντομο χρονικό διάστημα. Η λύση που παίρνουμε δεν είναι απαραίτητα η βέλτιστη, ωστόσο όμως είναι αποδοτική δεδομένου του υπολογιστικού χρόνου που επιτυγχάνουμε είτε στην περίπτωση του περιοδεύοντος πωλητή στις πενήντα μεγαλύτερες πόλεις της Ελλάδας είτε στην περίπτωση ανάγνωσης δεδομένων από ένα σκληρό δίσκο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

ΙΣΤΟΣΕΛΙΔΕΣ

http://dspace.lib.ntua.gr/bitstream/123456789/862/4/sakellariouf_manufacturing.pdf

http://www.math4u.gr/files/bibliaeap/LIKOTHANASIS.pdf

http://edu.eap.gr/pli/pli31/docs/GAs_introduction.pdf

http://www.math4u.gr/files/bibliaeap/LIKOTHANASIS.pdf

http://www.hep.upatras.gr/class/download/eyf_an_ded_anagn_prot/E8PEES_GeneticAlgorithms.pdf

http://poseidon.library.tuc.gr/artemis/MT2009-0005/MT2009-0005.pdf

http://ikee.lib.auth.gr/record/131424/files/%CF%80%CF%84%CF%85%CF%87%CE%B9%CE%B1%CE%BA%CE%B7-final.pdf

http://nemertes.lis.upatras.gr/jspui/bitstream/10889/6347/1/diplomatiki_Nikolas_Stylianou.pdf

http://arkolakis.gr/goods/elements/2015/10/genetikoi-algori8moi-traveling-salesman-problem.pdf

http://hothardware.com/reviews/Seagate-Barracuda-3TB-Review?page=2

http://nemertes.lis.upatras.gr/jspui/bitstream/10889/6347/1/diplomatiki_Nikolas_Stylianou.pdf

https://www.google.gr/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjprcrKy67LAhVBsRQKHSMbA3MQFggoMAI&url=http%3A%2F%2Fwww.tem.uoc.gr%2F~vagelis%2FCourses%2FTEM231%2FProject_1_TSP.pptx&usg=AFQjCNEccgBBjlYNdKDM8qXyIpHOtActug&sig2=Cl6fAgEPFtKBAoa2mt4d6g

ΒΙΒΛΙΑ

Εθνικό Μετσόβιο Πολυτεχνείο ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΠΑΡΑΓΩΓΗΣ ΜΕ ΧΡΗΣΗ ΓΕΝΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ Νικόλαος Μιχ. Ουλκέρογλου

23rd – 27th April 2011 5th conference shanghai :

Network structure optimazation by usinga GA(Genetic Algorithm) Network evolution technologies.