**Πανεπιστήμιο Πειραιώς Τμήμα Πληροφορικής**

**Πτυχιακή Εργασία**

Όνομα Φοιτητή : Μιχαλόπουλος Κυριάκος Π/06087

Υπεύθυνοι Καθηγητές : Ευάγγελος Φούντας

Θέμα: Ανάλυση του Αλγορίθμου Bat και

Εφαρμογές του



Ακαδημαϊκό έτος : 2012-2013

***1.Εισαγωγικά στοιχεία***

*1.1 Εισαγωγικά στοιχεία για τους Μεταευριστικούς Αλγόριθμους*

*1.1.1Μέθοδος Βελτιστοποίησης με Αποικίες Μυρμηγκιών (Ant Colony*

*Optimization – ACO)*

*1.2 Firefly Algorithm*

*1.3 Λιγα λόγια για τις μη ψηφιακές νυχτερίδες*

*1.4Εισαγωγικά στοιχεία για τον αλγόριθμο bat*

*1.4.1 Συνοπτική περιγραφή του αλγόριθμου bat*

*1.4.2 Μαθηματική αναλυσή του Αλγόριθμου*

*1.4.3.1 Κίνηση των «ψηφιακών» νυχτερίδων*

*1.4.3.2 O bat algorithm με ψευδοκώδικα*

*1.4.4 Διαγραμμά ροης του ψευδοκώδικα*

*1.4.5 Ηχηρότητα και ρυθμός εκπομπής παλμών*

***2. Εφαρμογές του αλγορίθμου bat***

*2.1 Το πρόβλημα της Συνδυασμένης Κατανομής Οικονομικού Κόστους και Εκπομπής Ρυπών*

*2.2 Περιγραφή του Προβλήματος*

*2.2.1 Οικονομική κατανομή*

*2.2.2 Κατανομή Εκπομπής Ρύπων*

*2.2.3 Συνδυασμένη Κατανομή Οικονομικού Κοστους και Εκπομπής Ρυπών με μαθηματική περιγραφη*

***3.Επίλυση του προβλήματος***

*3.1 Παρουσίαση του case study*

*3.2 Σύνδεση bat algorithm και CEED*

*3.3 Υλοποίηση του αλγορίθμου σε C++*

***4.Συγκρίσεις των αποτελεσμάτων των αλγορίθμων***

***1.Εισαγωγικά στοιχεία***

*1.Εισαγωγικά στοιχεία για τους Μεταευριστικούς Αλγόριθμους*

Πριν αναλύσουμε τον bat algorithm, χρήσιμο είναι να αναφερθούμε συνοπτικά και σε αλλούς αλγόριθμους που ανήκουν στην κατηγορία του bat ,δηλαδή στους μεταευριστικους αλγόριθμους .

*1.1 Μέθοδος Βελτιστοποίησης με Αποικίες Μυρμηγκιών (Ant Colony*

*Optimization – ACO)*

Τις τρεις τελευταίες δεκαετίες υπάρχει μια έντονη δραστηριοποίηση στο χώρο των υπολογιστικών αλγορίθμων που βασίζονται στη μοντελοποίηση διαφόρων φυσικών, βιολογικών ή κοινωνικών φαινομένων με σκοπό τη δημιουργία μη αιτιοκρατικών ευριστικών αλγορίθμων για την επίλυση NP – δύσκολων Συνδυαστικών Προβλημάτων Βελτιστοποίησης (NP – hard Combinatorial Optimization Problems – COP) αλλά και για την εύρεση του ελαχίστου ή του μεγίστου συνεχών συναρτήσεων ή προβλημάτων.

Ένας από τους πιο πρόσφατους αλγορίθμους που αναπτύχθηκαν στα πλαίσια της παραπάνω έρευνας είναι και η Μέθοδος Βελτιστοποίησης με Αποικίες Μυρμηγκιών (Ant Colony Optimization – ACO) η οποία βασίζεται στη συλλογική συμπεριφορά που εμφανίζουν τα μυρμήγκια κατά την αναζήτηση και τη συγκομιδή τροφής. Τα αποτελέσματα από τη χρήση του ACO σε διακριτά προβλήματα είναι πολύ ενθαρρυντικά και συγκρίσιμα με μεθόδους που έχουν ερευνηθεί και αναπτυχθεί για πολύ μεγαλύτερο χρονικό διάστημα (πχ Γενετικοί Αλγόριθμοι).

Οι ομοιότητες των «ψηφιακών» μυρμηγκιών με τα φυσικά έχουν ως εξής :

**1)Η αποικία.** Όπως στα φυσικά μυρμήγκια έτσι και στα «ψηφιακά» υπάρχει μια αποικία ατόμων τα οποία συνεργάζονται προκειμένου να δημιουργήσουν μια συγκεκριμένη δομή στο σύστημα τους. Ο πληθυσμός των «ψηφιακών» μυρμηγκιών είναι μια παράμετρος η οποία είναι σε άμεση συνάρτηση με το είδος του προβλήματος.

**2)Η φερομόνη.** Ορισμένες «ποικιλίες» πραγματικών μυρμηγκιών εναποθέτουν φερομόνη στο έδαφος κατά την κίνησή τους. Ομοίως και τα «ψηφιακά» μυρμήγκια θα εναποθέτουν στις διάφορες θέσεις τις οποίες επισκέπτονται μια συγκεκριμένη αριθμητική πληροφορία, την «**ψηφιακή» φερομόνη**, οπότε και θα δημιουργούνται **μονοπάτια φερομόνης** (*pheromone* *trails*). Αυτή η αριθμητική πληροφορία θα εξαρτάται από την επίδοση και την προηγούμενη ιστορία του «ψηφιακού» μυρμηγκιού και θα είναι προσβάσιμη και από τα υπόλοιπα μέλη της αποικίας εγκαθιδρύοντας με αυτόν τον τρόπο στιγμεργετική (*stigmergetic*) επικοινωνία. Τέλος, η «ψηφιακή» φερομόνη όπως και η τεχνητή θα πρέπει να υφίσταται εξάτμιση έτσι ώστε να υπάρχει αρνητική ανάδραση και να μην εγκλωβίζονται τα άτομα σε τοπικά βέλτιστα.

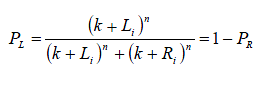
**3)Η απόφαση.** Όπως τα φυσικά έτσι και τα «ψηφιακά» μυρμήγκια αποφασίζουν την επόμενη κίνησή τους με κάποια στοχαστικότητα. Τα δεδομένα που έχουν στη διάθεσή τους είναι καθαρά τοπικά στο χώρο και το χρόνο. Αναλυτικότερα, τα μυρμήγκια, είτε πραγματικά είτε «ψηφιακά», δεν έχουν γνώση του τι συμβαίνει σε κάποιο άλλο σημείο όπου δραστηριοποιούνται μυρμήγκια τις αποικίας αλλά ούτε και τη γενικότερη κατάσταση της αποικίας. Επίσης, δεν γνωρίζουν τι

έγινε τις προηγούμενες χρονικές στιγμές στο σημείο όπου βρίσκονται αλλά ούτε μπορούν να προβλέψουν μελλοντικές καταστάσεις.

Ο Deneubourg θέλοντας να επιβεβαιώσει τα αποτελέσματα των πειραμάτων του δημιούργησε ένα μαθηματικό μοντέλο της συμπεριφοράς των πραγματικών.Έστω *Li* και *Ri* τα μυρμήγκια που έχουν επιλέξει την αριστερή και τη δεξιά διαδρομή αντίστοιχα μετά το πέρας της κίνησης του *i*-οστού μυρμηγκιού (δηλαδή *i = Li+Ri* ο συνολικός αριθμός των μυρμηγκιών που έχουν περάσει το σημείο επιλογής διακλάδωσης Α). Τότε η

πιθανότητα *PL* το επόμενο μυρμήγκι (*i+1*) να επιλέξει την αριστερή διαδρομή δίνεται από τη

σχέση:



Οι **επιπλέον ικανότητες** που προαιρετικά μπορούν να προστεθούν στα «ψηφιακά» μυρμήγκια έχουν ως εξής:

**Α)Ύπαρξη «μνήμης»:** για να θυμούνται τις προηγούμενες πράξεις τους ή άλλες χρήσιμες πληροφορίες.

**Β)Ύπαρξη «όρασης» ή άλλων «αισθητηρίων»:** ώστε να εκμεταλλεύονται καλύτερα τις πληροφορίες του περιβάλλοντος (πχ. να μπορούν να υπολογίσουν εκ των προτέρων τις

αποστάσεις των πόλεων στο TSP και όχι αφού έχουν ολοκληρώσει μια διαδρομή).

**Γ)Διακριτοποίηση του χώρου και του χρόνου:** για την επίλυση διακριτών προβλημάτων. Για παράδειγμα, οι κινήσεις των μυρμηγκιών στο TSP είναι διακριτές από πόλη σε πόλη και

πραγματοποιούνται σε διακριτές χρονικές στιγμές.

**Δ)Εναπόθεση φερομόνης σε διαφορετικές χρονικές στιγμές:** ενώ τα φυσικά μυρμήγκια εναποθέτουν συνεχώς φερομόνη, τα «ψηφιακά» το κάνουν συνήθως αφού ολοκληρώσουν τις κινήσεις τους.

**Ε)Χρησιμοποίηση αλγορίθμων για τη βελτίωση της απόδοσης του συστήματος:** μπορεί να

χρησιμοποιηθούν αλγόριθμοι τοπικής βελτιστοποίησης κτλ.

*1.2 Firefly Algorithm*

Ο **αλγόριθμος πυγολαμπίδα (FA)** είναι ενάς metaheuristic αλγόριθμος , εμπνευσμένος από την συμπεριφορά των πυγολαμπίδων που αναβοσβήνουν . Ο πρωταρχικός σκοπός ενός flash για μία πυγολαμπίδα είναι να δράσει ως ένα σήμα για το σύστημα μας και να προσελκύσει άλλες πυγολαμπίδες. Ο Xin-She Yang διατύπωσε αυτόν τον αλγόριθμο με τις εηής παραδοχές:

1. Όλες οι πυγολαμπίδες είναι μονογενή(δεν διαχωρίζονται σε αρσενικές και θυληκές), έτσι ώστε το ένα firefly θα πρέπει να προσελκύονται σε όλες τις άλλες πυγολαμπίδες.
2. Η ελκυστικότητα μιας πυγολαμπίδας είναι ανάλογη της φωτεινοτητάς τους, καθώς για οποιεσδήποτε δύο πυγολαμπίδες, η λιγότερο φωτεινή προσελκύεται από (και συνεπώς κινηθεί προς) τη φωτεινότερη. Ωστόσο, η φωτεινότητα μπορεί να μειωθεί καθώς αυξάνεται η απόστασή τους.
3. Εάν δεν υπάρχουν πυγολαμπίδες ιδιαίτερα φωτεινές σε σύγρκτιση με τις άλλες,θα κινούνται τυχαία.

Η φωτεινότητα θα πρέπει να συνδέεται με την αντικειμενική συνάρτηση. Ο Firefly αλγόριθμος είναι ενας εμπνευσμένος από την φύση μεταευριστικός αλγόριθμος βελτιστοποίησης.

Ο ψευδοκώδικας του αλγορίθμου έχει ως εξής:

ΑΡΧΗ

1) Στόχος της λειτουργίας:



2) Δημιουργία ένος αρχικού πληθυσμού πυγολαμπίδων



3) Διατύπωση της έντασης του φωτός έτσι ώστε να συνδέεται με το



(Για παράδειγμα, για τα προβλήματα μεγιστοποίησης, ή απλά



4) Καθορισμός συντελεστή απορρόφησης



Ενώ (t MaxGeneration <)

για i = 1: n (όλα πυγολαμπίδες n)

για j = 1: n (n πυγολαμπίδες)

αν ( ),



κινούνται προς πυγολαμπίδα i j

τέλος, αν

Μεγάλη ελκυστικότητα με απόσταση r μέσω



Αξιολογήση νέων λύσεων και να ενημερώση της έντασης του φωτός

τέλος για την ι

τέλος για το i

Δημιουργία σειράς από πυγολαμπίδες και να εύρεση της τρέχουσας καλύτερης

ενώ τέλος

Μετα-επεξεργασία των αποτελεσμάτων και την απεικόνιση

ΤΕΛΟΣ

Ο κύριος τύπος για την ενημέρωση των τιμών ενός ζεύγους πυγολαμπίδων xi και xj είναι:



όπου είναι μία παράμετρος που ελέγχει το μέγεθος του βήματος, ενώ είναι ένας φορέας που λαμβάνεται από μία Gaussian κατανομή.



Μπορεί να αποδειχθεί ότι η οριακή περίπτωση αντιστοιχεί στο πρότυπο Βελτιστοποίησης με Σμήνος Σωματιδίων (PSO). Στην πραγματικότητα, εάν ο εσωτερικός βρόχος (για j) απομακρύνεται και η φωτεινότητα αντικαθίσταται από την τρέχουσα γενικά καλύτερη , Τότε ο FA γίνεται ουσιαστικά το πρότυπο ΥΔΥ.



*1.3 Λίγα λόγια για τις μη ψηφιακές νυχτερίδες*

Οι νυχτερίδες είναι από κάθε άποψη μοναδικά και εκπληκτικά πλάσματα. Είναι τα μόνα θηλαστικά που μπορούν να πετούν, δεν έχουν βέβαια φτερά, πετούν χάρη στην ιδιαίτερη κατασκευή των μπροστινών τους άκρων. Το κεφάλι τους μοιάζει πολύ με του ποντικιού τα μάτια τους είναι μικρά αλλά έχουν πολύ δυνατή όραση. Η βασική της διαίρεση είναι σε σαρκοφάγες (που τρέφονται με έντομα ή δύο είδη της Νότιας Αμερικής που τρέφονται με αίμα) και οπωροφάγες (που τρέφονται με φρούτα). Τα περισσότερα είδη νυχτερίδας έχουν ως τόπο διαμονής σπήλαια και βγαίνουν τη νύχτα για να τραφούν.

Θα μπορούσαμε να πουμέ πως είναι εξαιρετικά παρεξηγημένα ζώα, γεγονός που οδήγησε κατά τον Μεσαίωνα στις τοπικές μυθοπλασίες-δοξασίες σχετικά με τον Κομή Δράκουλα και αργότερα ενεπνεύσε την λογοτεχνία προς την δημιουργία του φανταστικού πλασματός βρυκόλακα,ένα υβρίδιο νυχτεριδάς και ανθρώπου.

Ακομά και σήμερα όμως δεν απολαμβάνουν ιδιαίτερη συμπάθεια και εκτίμηση και σίγουρα δεν μπορούμε να συναντήσουμε ανθρώπους που τα φροντίζουν σπίτι τους ως κατοικίδια. Αυτό οφείλεται στην λάθος εντύπωση ότι καποίες από αυτές τρέφονται με αίμα. Οι νυχτερίδες βαμπίρ δεν πίνουν αίμα. Απλά το γλύφουν. Υπάρχουν μόνο 3 είδη νυχτερίδων βαμπίρ παγκοσμίως (ετσι ονομάζονται οι νυχτερίδες αυτές!). Εάν ταξιδεύετε στην Κεντρική ή στη Νότιο Αμερική, υπάρχει πιθανότητα να δείτε μια νυχτερίδα βαμπίρ να δαγκώνει μια αγελάδα και να γλύφει το αίμα από την πληγή αλλά όχι να το ρουφάει!

Ο πληθυσμός των νυχτερίδων αποτελεί το 1/4 του παγκόσμιου πληθυσμού των θηλαστικών! Δηλαδή πανώ από 950 είδη νυχτερίδας ζούν σε κάθε γωνία του πλανήτη. Αυτό το γεγονός μας οδηγεί ότι οι νυχτερίδες είναι πλάσματα εκπληκτικά προικισμένα με εργαλεία για την επιβιωσή τους. Εδώ ακριβώς θα εστιάσουμε στην πιο εκπληκτική ικανότητα της νυχτερίδας ,από οπού και εμπνεύστηκε ο αλγόριθμος bat.

Οι νυχτερίδες χρησιμοποιούν ένα είδος σόναρ για να μετακινηθούν στο σκοτάδι! Δεν βλέπουν τόσο καλά και ζουν σχεδόν αποκλειστικά το βράδυ, γι’ αυτό και πρέπει να στηρίζονται σε άλλες μεθόδους “πλοήγησης” κατά την πτήση εκτός της όρασης. Οι νυχτερίδες στέλνουν σήματα ηχούς στο περιβάλλον και λαμβάνουν πίσω τις αντανακλάσεις αυτές, ξέροντας έτσι τι βρίσκεται γύρω τους. Είναι νυχτερινοί τύποι, περισσότερο γιατί είναι εύκολο την νύχτα να κυνηγήσουν έντομα κι από την άλλη αποφεύγουν και τις ατυχείς συναντήσεις με τους κυνηγούς τους. Την ισχνή τους όραση την χρησιμοποιούν για την ημέρα αλλά τις περισσότερες “δουλειές” τους, τις κάνουν με το κάλλυμα της νύχτας. Ετσί σε συνθήκες απολυτού σκότους εντοπίζουν με ακρίβεια την θέση της λείας τους ,αλλά και την θέση των εχθρών τους.

*1.4 Εισαγωγικά στοιχεία για τον αλγόριθμο bat*

Ο bat algorithm εμπνευσμένος από την συμπεριφορά των νυχτερίδων ,αλλά και το σπάνιο σύστημα ηχοεντοπισμού που χρησιμοποίουν είναι ενάς μεταευριστικός αλγόριθμος βελτιστοποίησης (metaheuristic optimization algorithm) που ανέπτυξε ο Xin-She Yang (Oxford University) το 2010.

*1.4.1 Συνοπτική περιγραφή του αλγόριθμου bat*

Επί της ουσίας ο αλγόριθμος ακολουθεί πιστά την βιολογική συμπεριφορά των νυχτερίδων που χρησιμοποιούν ηχοεντοπισμό. Κάθε νυχτερίδα πετάει τυχαία με ταχύτητα υi στην θέση xi με κυμαινόμενη συχνότητα μήκους κύματος Ai . Όσο αναζητεί και βρίσκει την λεία της, αλλάζει συχνότητα, ηχηρότητα και ρυθμό στον σφιγμό της r . Κανουμέ αναζήτηση με random walk για να αξιολογήσουμε τις νυχτερίδες. Επιλέγουμε τις καλύτερες μέχρι να ικανοποιήθουν καποία κριτήρια τερματισμού(stop criteria). Δηλαδή χρησιμοποιούμε μια τεχνική βασισμένη στο να αλλάζουμε την συχνότητα για να ελέγξουμε την δυναμική αλλαγή συμπεριφοράς ενός σμήνους από νυχτερίδες και η ισορροπία μεταξύ της ανακάλυψης και εκμεταλλεύσης αποτελεσμάτων μπορεί να επιτευχθεί αλλάζοντας δοκιμαστικά αλγοριθμικούς παράγοντες στον αλγορίθμο bat.

## H λεπτομερής εισαγωγή του Xin-She Yang στους μεταευριστικούς αλγόριθμους (metaheuristic algorithms) και τον αλγόριθμο bat φυσικά ,άλλα και η εκτενής κριτικη των Parpinelli και Lopes ,με στόχο την περεταίρω εξέλιξη του αλγορίθμου σε ένα εξελίσιμο αλγόριθμο bat (evolving bat algorithm- EBA),οδήγησε στην δημιουργία του Multi-objective Bat Algorithm (MOBA).

## Ο Multi-objective Bat Algorithm χρησιμοποιεί ,και εδώ επι της ουσιάς διαφέρει από τον αρχικό αλγόριθμο, ένα απλό σταθμισμένο σύνολο από τυχαία βάρη. Ενας τέτοιος αλγόριθμος αναπτύχθηκε για να λύσει multi-object προβλήματα σχεδίασης μηχανικής. Και ενας αλλός συνδυασμένος με NSGA-II αποδείχθεικε πολύ πιο αποτελεσματικός από τον «απλό» NSGA-II.

## Γενικότερα αυτή είναι η μεγάλη χρησιμότητα αυτού του αλγορίθμου από μόνος του μπορεί να είναι εφαρμόσιμος ή μη,βέλτιστος για ενά πρόβλημα ή όχι ,αλλά συνήθως συνδυασμένος με έναν αλλό αλγόριθμο δινεί βέλτιστα αποτελέσματα.

*1.4.2 Εκτενέστερη αναλυσή του Αλγόριθμου και από μαθηματική αποψή*

Αν και καποία είδη νυχτερίδας έχουν εξαιρετική ακοή και όσφρηση η ικανότητα καποίων είδων για ηχοεντοπισμό ενέπνευσε τον Xin-She Yang το 2010 στην δημίουργια του bat algorithm.

Σε αυτό το κεφάλαιο λοίπον θα δώσουμε ένα μαθηματίκο υπόβαθρο και περιγραφή στον αλγόριθμο αυτό. Εξαιδανικεύοντας κάποια χαρακτηριστίκα των νυχτερίδων μπόρουμε να φτίαξουμε διάφορους αlγόριθμους bat ή αλγόριθμους εμπνευσμένους από τον αλγόριθμο bat. Σε γενικές γραμμές στην βιβλιογραφία ακολούθουνται οι εξής κανόνες :

1. Ολές οι νυχτερίδες χρησιμοποίουν ηχοεντοπισμό για να υπολογίσουν την απόσταση, και επίσης καταλαβαίνουν την διαφορά μεταξύ της λείας τους και των εμπόδιων στο φυσικό τους περιβάλλον (τοίχωματα στην σπηλία,πέτρες κτλ)
2. Οι νυχτερίδες πετούν τυχαία με ταχύτητα υi θέση xi με μία σταθερή συχνότητα fmin (ή μήκος κύματος λ),μεταβαλλόμενο μήκος κύματος λ (ή συχνότητα f ) και ηχηρότητα Α0 για να αναζητήσουν την λεία τους. Μπορούν κατά βούληση να προσαρμόσουν το μήκος κύματος (ή την συχνότητα) των εκπεμπόμενων παλμών τους και ρυθμίζουν τον ρυθμό των σφιγμών τους ρ∈ [0,1], εξαρτάται από το πόσο κοντά είναι ο στόχος τους.

Επι πρόσθετα αυτών των παραδοχών κάνουμε και κάποιες προσεγγίσεις στις μεταβλητές μας για απλότητα. Γενικά η συχνότητα f παίρνει τιμές στο σύνολο [fmin,fmax] αντιστοιχεί σε μήκος κύματος που παίρνει τιμές στο σύνολο [λmin,λmax]. Εδώ να θυμίσουμε έναν τύπο της Φυσικής ** , Εξίσωση 1.**

Επομένως μια συχνότητα διαστήματος [20 kHz,500kHz] αντιστοιχεί σε κυμαινόμενο μήκος κύματος [0,7mm,17mm].

Για να λύσουμε ένα πρόβλημα μπορούμε να χρησιμοποιήσουμε οποίο μήκος κύματος μας βοηθάει στην εφαρμογή της τιμής του. Μπορούμε να αλλάξουμε και το range πειραματιζόμενοι με τις τιμές των συχνοτήτων ή μήκων κύματος. Ωστόσο γνωρίζοντας ότι η ποσότητα λf είναι σταθερή μπορούμε να αγνοήσουμε τελείως το μήκος κύματος πειράζωντας την συχνότητα και βάση αυτής ορίζουμε το μήκος κύματος για να ισχύουν τα παραπάνω.

Για απλότητα μπορούμε επίσης να υποθέσουμε ότι : f ∈ [0,fmax]. Γνωρίζουμε επίσης οτί υψηλότερες συχνότητες αντιστοιχούν σε μικρότερα μήκη κύματος και ταξιδεύουν μικρότερες αποστάσεις. Ενώ χαμηλότερες συχνότητες αντιστοιχούν σε μεγαλύτερα μήκη κύματος και ταξιδεύουν μεγαλύτερες αποστάσεις. Για τις νυχτερίδες συνήθως η αποστασή είναι μερικά μέτρα. Ο ρυθμός των παλμών που εκπέμπει η νυχτερίδα μπορεί απλά να είναι [0,1] ,όπου 0 σήμαινει ότι δεν έχουμε καθόλου παλμούς και 1 τον μέγιστο ρυθμό εκπομπής παλμών.

*1.4.3.1 Κίνηση των «ψηφιακών» νυχτερίδων*

Στις προσομοιώσεις χρησιμοποιούνται ψηφιακές νυχτερίδες. Καπώς πρέπει να ορίσουμε πως αλλάζουν η ταχύτητα υi και η θέση xi σε ένα χώρο αναζήτησης n-διαστάσεων. Μετα από ερεύνα στην βιβλιογραφία βρήκα ότι υπαρχούν εξισώσεις που μας δίνουν την νέα ταχύτητα υit και νέα θέση xit οπου t είναι κάποια χρονική στιγμή.

Είναι :

 =  + ( - ) β **,Εξίσωση 2**

** =  + ( -** **)** **,Εξίσωση 3**

** =  +  ,Εξίσωση 4**

οπου το β ∈ [0,1] είναι ένα τυχαίο διάνυσμα που προέκυψε από μία ομοίομορφη κατανομή, **x\***  είναι η τρέχουσα καλύτερη θέση (λύση του προβλήματος), η οποία υπολογίζεται αφου συγκρίθουν όλες οι λύσεις που δίνουν οι n νυχτερίδες που έχουμε εξετάσει. Καθώς το γινόμενο λifi είναι η ταχύτητα (που μένει σταθερή) μπορούμε να χρησιμοποιήσουμε το fi ( ή λi ) για να διορθώσουμε το λi ( ή fi αντίστοιχα).

Εφόσον επιλεγεί μια λύση ανάμεσα από τις τρέχοντες καλύτερες λύσεις δημιουργείται μια καινούργια λύση για κάθε νυχτερίδα με την χρήση της μαθηματικής μεθόδου random walk.

xnew = xold + ∈ At **,Εξίσωση 5**

Οπού ∈ **∈** [-1,1] είναι ενας τυχαίος αριθμός ,ενώ Αt = <Ait> είναι η μέση ηχηρότητα για την συγκεκριμένη χρονική στιγμή t.

*1.4.3.2 O bat algorithm με ψευδοκώδικα*

Αρχικοποίησε τον πληθυσμό των νυχτερίδων με τις θέσεις και ταχυτητές τους xi και υi οπού ( i = 1, 2,…,n ).

Αρχικοποίησε τις συχνότητες fi , ρυθμούς εκπομπής παλμών ri και την ηχηρότητα Ai.

**while** ( t < μεγιστού αριθμού επαναλήψεων)

Δημιούργησε νέες λύσεις ρυθμίζοντας την συχνότητα και αλλάζωντας αναλόγα τις τιμές των ταχυτήτων και των θέσεων/λύσεων (Χρηση εξισώσεων 2 , 3 και 4 ).

**if** (rand > ri)

Διάλεξε μια λύση αναμέσα στις καλύτερες λύσεις

Δημιούργησε μια τοπική λύση γύρω από την επιλεγμένη καλύτερη λύση

**end\_if**

Δημιούργησε μια νέα λύση με τυχαίο πέταγμα

**if** (rand < Ai & f(xi) < f(x\*) )

Δέξου τις νέες λύσεις

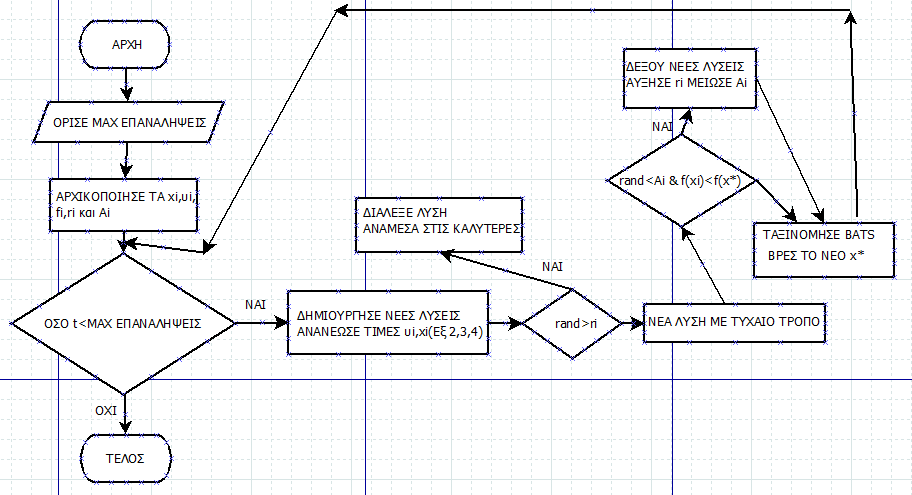
Αύξησε το ri και μείωσε το Αi

**end\_if**

Ταξινόμησε τις νυχτερίδες και βρές την ισχύουσα καλύτερη λύση x\*

**end\_while**

*1.4.4 Διαγραμμά ροης του ψευδοκώδικα*



*1.4.5 Ηχηρότητα και ρυθμός εκπομπής παλμών*

Δεν πρέπει να παραλείψουμε την ανάγκη ενημέρωσης των μεταβλήτων της ηχηρότητας Αi (loudness) και του ρυθμού εκπομπής ηχητικών παλμών ri (rate of pulse emission) , οσον αφορά την τιμή τους καθώς οι επαναλήψεις του αλγορίθμου προχωρούν. Αν και το loudness Ai συνήθως μειώνεται οταν μία νυχτερίδα βρίσκει την λεία της,ενώ το pulse emission αυξάνει,ειτρέπεται να λαβούμε τιμή για το loudness που να είναι εύκολη στους υπολογισμούς μας. Για παράδειγμα ,έστω Α0=150 και Αmin= 1 ,μπορούμε να υποθέσουμε οτι Α0=1 και Αmin=0, ακολουθώντας μια δυαδική προσέγγιση στο πρόβλημα μας ,διότι είναι εύκολα κατανοητό οτι Αmin=0 σημαίνει οτι μια νυχτερίδα μόλις βρήκε την λεία και προσωρινά σταματάει να εκπέμπει ηχχοπαλμούς.

Είναι :

 **Εξίσωση 6**

,οπού τα α και γ είναι σταθερές.Αν και σε απλές περιπτώσεις μπορούμε απλά να υποθέσουμε οτι α=γ συνήθως για τις σταθερές αυτές ,για 0<α<1 και γ>0 είναι :

,αν **Εξίσωση 7**

***2. Εφαρμογές του αλγορίθμου bat***

*2.1 Το πρόβλημα της Συνδυασμένης Κατανομής Οικονομικού Κόστους και Εκπομπής Ρυπών (Combined Economic and Emission Dispatch)*

Ένα πρόβλημα οικονομικής φύσης είναι η εύρεση τρόπων ελαχιστοποίηση του κόστου παραγωγής της ενέργειας. Με την πιο παραδοσιακή ας πουμε οικονομική προσέγγιση στις μελέτες ,το λειτουργικό κόστος μειώνεται με την κατάλληλη τοπολογία της ποσότηταςς της παραχθείσα ενέργειας απο τις διάφορες μονάδες παραγωγής. Ωστόσο αυτή η προσέγγιση δεν μας εξασφαλίζει οτι θα ικανοποιεί και περιβαντολογικές ανάγκες παράλληλα.

Τα τελευταία χρόνια εχουμέ ακούσει πολλά για την μόλυνση του περιβάλλοντος και τις απείλες που αντιμετωπίζουν τόσο η χλωρίδα όσο και η πανίδα. Πράγματι πολλές χώρες ,σε όλο τον κόσμο, κάνουν προσπάθειες που επικεντρώνονται στην μείωση εκπομπή ρυπογόνων ουσιών απο μονάδες παραγωγής που χρησιμοποιούν ορυκτά καύσιμα για την λειτουργία τους.

Τετοίοι ρύποι ,εκτός των στερεών (σε μορφή σωματιδίων) ρύπων,υπαρχούν τρία είδη αέριων τύπων : τα οξείδια του θείου () και τα οξείδια του αζώτου () ,αλλά και το διοξείδιο του άνθρακα (),που βλάπτουν σημαντικά το περιβάλλον αλλά δεν είναι το θέμα της εργασίας να αναλυθούν περαιτέρω αυτά τα θέματα.

Εχωντάς υποψήν λοιπόν τα παραπάνω γίνεται κατανοητό οτι το πρόβλημα αυτό δέχεται λύσεις απο τελείως διαφορετική σκοπιά.Απο την δεκαετία του 70’ πλέον οι στατηγικές κινούνται προς α) την ελαχιστοποίηση της μόλυνσης και β) την ελαχιστοποίηση του κόστου παραγωγής τηρουμένων των περιορισμών εκπομπής ρυπών.

Λύνωντας το πρόβλημα της **οικονομική κατανομής φορτίου(Economic Load Dispatch)** των θερμικών μονάδων παραγωγής ,συμβάλλουμε στην ελαχιστοποίηση του κόστους παραγωγής ενέργειας.Στόχος είναι να οριστεί η οικονομική κατανιομή φόρτιου κάθε γεννήτριας ,ωστέ να αντιμετωπίζει επαρκώς την ζήτηση,χώρις να παραβιάζονται τα λειτουργικά όρια κάθε σταθμού.Αρκετές μέθοδοι έχουν προταθεί για την λύση του προβλήματος ελαχιστοποίησης του κόστους παραγωγής ενέργειας ,που βασίζονται στη προσέγγιση του πολλαπλασιαστή του LaGrange.

Είναι προφανές οτι το πρόβλημα γίνεται πιο περίπλοκο οταν πρέπει να υπολογιστεί και το συνολικό αντίκτυπο των εκπομπών βλαβερών ρυπών στο περιβάλλον.Εδώ ορίζουμε το πρόβλημα **περιβαντολογικά περιοριζόμενης οικονομικής κατανομής (Environmentally Constrained Economic Dispatch , ECED**). Το πρόβλημα αυτό συνίσταται στον καθορισμό των αποδεκτών επιπέδων μόλυνσης που δεν πρέπει σε καμμια των περιπτώσεων να υπερβαίνονται,δηλαδή με την εύρεση της βέλτιστης λύσης του προβλήματος τη οικονομικής μελέτης πρέπει να τηρούνται τα οριακά επίπεδα εκπομπών ρυπών ,καθότι αποτελούν λειτουργικούς περιορισμούς που πρέπει να τηρούνται.Επιπλέον τα ανόμοια χαρακτηριστικά των εκπομπών διαφορετικών ρύπων κάνουν ακόμα πιο περίπλοκο το πρόβλημα ,καθώς συχνά η μαθηματική τους μοντελοποίηση είναι μη γραμμική.

Για όλους του παραπάνω λόγους, προσεγγίσεις για την λύση του ECED βασισμένες στην τεχνική του πολλαπλασιαστή του LaGrange αποτυγχάνουν να εντοπίσουν μια συνολικά βέλτιστη λύση.Αλλά το ίδιο ισχύει και για τις μη γραμμικές μεθόδους που βασίζονται στην τεχνική της μεθόδους του Newton.

Ετσί λοιπόν οι K.Srikrishna και C.Palanichamy πρότειναν μια μέθοδο για την **συνδυασμένη κατανομή οικονομικού κόστους και εκπομπής ρυπών(Combined Economic and Emission Dispatch , CEED)**,εισάγωντας τη χρήση του παράγοντα ποινής κόστους(penalty factor).Αλλοί μελετητές ,χωρίς επιτυχία, αποπειράθηκαν με τεχνικές που συνδυάζουν την ασαφή λογική(fuzzy logic) με δοκιμασμένες μεθόδους βελτιστοποίησης να επιλύσουν το ECED ή το E&ED(Economic and Emission Dispatch,κατανομή οικονομικού κόστου και εκπομπής ρυπών),ωστόσο δεν επιτύγχαναν στην εύρεση της συνολικά βέλτιστης λύσης.

*2.2 Περιγραφή του Προβλήματος*

*2.2.1 Οικονομική κατανομή*

Το πρόβλημα της μελέτης οικονομικής κατανομής φορτίου (Economic Load Dispatch,ELD) είναι απο τα κυρίαρχα όσον αφορά τα συστήματα λειτουργίας και σχεδιασμού δικτύου μονάδων παραγωγής ενέργειας.Ετσί το πρόβλημα ELD μπορεί να περιγραφεί ως ,υπο περιορισμούς,ευρέση ελάχιστου κόστους παραγωγής. Η καμπύλη του κόστους καυσίμων προσεγγίεται με μια δευτεροβάθμια εξίσωση της ενεργής παραγωγής ενέργειας απο τις μονάδες παραγωγής.

Το ELD πρόβλημα μπορούμε να το εκφράσουμε ως το ακόλουθο πρόβλημα βελτιστοποίσης :

 **,Εξίσωση 8**

Όπου:

Fcost : Συνολικό κόστος καυσίμων στο σύστημα ($/hr)

αi, βi ,γi : Συντελεστές κόστους καυσίμων της i-οστής μονάδας παραγωγής

n : Αριθμός θερμικών μονάδων παραγωγής

Η παραπάνω συνάρτηση κόστους ακολουθεί του παρακάτω περιορισμούς :

1. Περιορισμός ζήτησης :

 **, Εξίσωση 9**

1. Όρια δυνατότητας παραγωγής :

 **, Εξίσωση 10**

Όπου:

PD : Συνολική ζήτηση δικτύου (MW)

PL : Συνολική απώλεια δικτύου μέσω του μέσου μεταφοράς(MW)

Pimin : Ελάχιστο ορίο παραγωγής i-οστή μονάδας παραγωγής(MW)

Pimax : Μέγιστο ορίο παραγωγής i-οστή μονάδας παραγωγής(MW)

Το PL υπολογίζεται απο τον τύπο:

 **,Εξίσωση 11**

Όπου Bij είναι τα στοιχεία του πινάκα συντελεστών απώλειας B,δηλαδή η απώλεια που θα υπάρχει από το μέσο μεταφοράς μεταξύ των σταθμών i και j.

*2.2.2 Κατανομή Εκπομπής Ρύπων*

Με την επίλυση του προβλήματος ELD θα βρούμε το ποσό εκείνο της ενέργειας που θα παραχθεί από διάφορες μονάδες παραγωγής ενός συστήματος για ένα ελάχιστο συνολικό κόστος καυσίμων.Αλλά ο περιορισμός της εκπομπής ρυπών δεν υπολογίζεται απο αυτό το πρόβλημα.Μπορούμε να περιγράψουμε ως κόστος τις ρυπογόνες ουσίες που εκπέμπονται στο περιβάλλον,καθώς υποβαθμίζουν το περιβάλλον με διάφορους τρόπους.

Η κατανομή εκπομπής ρύπων (Emission Dispatch,ED) στοχεύει στο να ελαχιστοποιηθεί η συνολική υποβάθμιση του περιββάλοντος ή της ελαχιστοποίησης των συνολικών ρύπων που παράγονται κατα την καύση των καυσίμων για την παραγωγή ενέργειας που να ανταποκρίνεται στη ζήτηση για ενέργεια. Η συνάρτηση εκπομπή ρύπων μπορεί να εφραστεί ως το άθροισμα όλων των ειδών της εκπομπής ,όπως CO2,(-SmOn), (-NmOn),στερέα σωματίδια και θερμική ακτινοβολία , καθέ μορφή ρύπου έχει και ανάλογο κόστος.

Το πρόβλημα του ED μπορεί να περιγραφεί απο το παρακάτω πρόβλημα ελαχιστοποίησης.

 **,Εξίσωση 12**

Όπου :

Εcost : Συνολική εκπομπή ρύπων στο περιβάλλον (kg/hr)

ai,bi,ci : Συντελεστές βαρύτητας του i-οστού ρύπου

n : Ο αριθμός των διαφορετικών ειδών ρύπων που εκπέμπονται.

*2.2.3 Συνδυασμένη Κατανομή Οικονομικού Κοστους και Εκπομπής Ρυπών με μαθηματική περιγραφη*

Μελετώντας χωριστά τοσο το πρόβλημα κατανομής οικονομικού κόστους (ELD) ,όσο και το πρόβλημα εκπομπής ρυπών (ED) καταλήγουμε οτι το πρόβλημα της Συνδυασμένης Κατανομής Οικονομικού Κοστους και Εκπομπής Ρυπών(CEED) μπορεί να παρουσιαστεί ως εξής :

minimize f(Fcost,Ecost) **,Εξίσωση 13**

Φυσικά το πρόβλημα υποτάσεται στους περιορισμους που περιγράφηκαν πιο πάνω απο τις εξισώσεις 9 και 10.

Μπορούμε να απλοποιήσουμε ωστόσο το πρόβλημα αυτό βελτιστοποίησης πολλαπλών στόχων κάνωντας το πρόβλημα βελτιστοποίησης ενος στόχου.Αυτό επιτυγχάνεται εισάγωντας ενα παράγοντα ποινής (penalty factor,Pf),ως εξής:

Minimize φ = Fcost + PfEcost $/hr , **Εξίσωση 14**

Ο παράγοντας ποινής Pf (penalty factor) συνδυάζει το κόστος εκπομπής με το κανονικό κόστος των καυσίμων.Μετα την εισαγωγή του στο συνολικό λειτουργικό κόστος του συστήματος είναι το κόστος των καυσίμων συν το επίκτητο κόστος εκπομπής ρύπων.Ο παράγοντας αυτός οδηγεί στην αποφύγη της χρήσης δύο διαφορετικών κατανομών.

Για τον υπολογισμό του ακολουθείται η παρακάτω διαδικασία:

1. Το κόστος των καυσίμων της κάθε μονάδας παραγωγής εκτιμάται κατά τη μέγιστη απόδοση της μονάδας,

$/hr **,Εξίσωση 15**

1. Η εμπομή ρύπων κάθε μονάδας παραγωγή εκτιμάται κατά τη μέγιστη απόδοση της μονάδας,

 Kg/hr **,Εξίσωση 16**

1. Υπολογίζεται το Pfi για κάθε μονάδα παραγωγής :

$/kg , i=1,2,…,n , **Εξίσωση 17**

1. Τα Pfi ταξινομούνται κατά αύξουσα σειρά.
2. Η μέγιστη απόδοση της κάθε μονάδας παραγωγής(Pmax)προστίθεται μία τη φορά,από τη ονάδα του μικρότερου Pfi έως ότου  **, Εξίσωση 18**

***3.Επίλυση του προβλήματος***

*3.1 Παρουσίαση του case study*

Έστω έχουμε δικτυο παραγωγής ενέργειας που αποτελείται απο 6 μηχανές τις Μ1,Μ2,Μ3,Μ4,Μ5,Μ6. Συμφώνα με το δοσμένο μας case study ισχύει

Εξισώσεις Κόστους Καυσίμων Πινακάς 1



Οπού  με i=1,2,3,4,5,6 είναι η εξισώση περιγραφής του συνολικού κόστους καυσίμου για την λειτουργία της i-μηχανής.Τα αi,βi,γi είναι μοναδικοί συντελεστές κόστους καυσίμου για την i-μηχανή που απο τις εξισώσεις είναι ήδη γνωστοί.Ωστόσο τα δεδομένα μας δεν σταματούν εκεί.

Εξισώσεις εκπομπής ρύπων Πίνακας 2



Οπού  με i=1,2,3,4,5,6 είναι η εξισώση περιγραφής της συνολικής εκπομπής ρύπων κατα την λειτουργία της i-μηχανής.Τα ai,bi,ci είναι μοναδικοί συντελεστές κόστους των διαφόρων ρύπων που εκπέμπει η i-μηχανή και απο τις εξισώσεις είναι ήδη γνωστοί.Παρατηρούμε οτι οι μηχανές Μ1 και Μ2,οι μηχανές Μ3 και Μ4 και οι μηχανές Μ5 και Μ6 εκπέμπουν ίδιους μεταξύ τους ρύπους ,μιά και a1,b1,c1=a2,b2,c2 , a3,b3,c3=a4,b4,c4 και a5,b5,c5=a6,b6,c6.

Επίσης ορίζουμε ως Li το κατώτερο ορίο ισχύους λειτουργίας της i-μηχανής ,δηλαδή Pimin=Li. Ως Ui ορίζεται το ανώτερο ορίο ισχύους λειτουγίας της i-μηχανής ,δηλαδή Pimax=Ui.

Είναι:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Πίνακας 3** | Μηχανή | Μ1 | Μ2 | Μ3 | Μ4 | Μ5 | Μ6 |
| Li (σε ΜW) | | 10 | 10 | 35 | 35 | 130 | 125 |
| Ui(σε MW) | | 125 | 150 | 225 | 210 | 325 | 315 |

Επίσης μας δίνεται ο πίνακας Βij που τα στοιχεία του αντιστοιχούν στους συντελεστές απώλειας Β,δήλαδη την απώλεια που θα υπάρχει απο το μέσω μεταφοράς μεταξύ των σταθμών i και j.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Πινάκας4** | i/j | M1 | M2 | M3 | M4 | M5 | M6 |
| M1 | | 1,40 | 0,17 | 0,15 | 0,19 | 0,26 | 0,22 |
| M2 | | 0,17 | 0,60 | 0,13 | 0,16 | 0,15 | 0,20 |
| M3 | | 0,15 | 0,13 | 0,65 | 0,17 | 0,24 | 0,19 |
| M4 | | 0,19 | 0,16 | 0,17 | 0,71 | 0,30 | 0,25 |
| M5 | | 0,26 | 0,15 | 0,24 | 0,30 | 0,69 | 0,32 |
| M6 | | 0,22 | 0,20 | 0,19 | 0,25 | 0,32 | 0,85 |

Οπώς μας έχει ζητηθεί θα τρέξουμε τον αλγόριθμο bat που θα φτιάξουμε τρείς φορές.Μία για Pd=500MW για 700MW και για 900MW.Όπου θυμίζουμε οτι Pd είναι η ζητούμενη ανάγκη ενέργειας που θέλουμε να παραχθεί απο το δίκτυο παραγωγής ενέργειας.

*3.2 Συνδέση Bat algorithm και του προβλήματος CEED*

Μπορούμε να θεωρήσουμε τις 6 μηχανές Μ1,Μ2,Μ3,Μ4,Μ5,Μ6 ως νυχτερίδες και να επιλύσουμε το πρόβλημα CEED με μια bat algorithm προσέγγιση.

Αν λοιπόν θεωρήσουμε καθέ μηχανή του δικτύου ως νυχτερίδα θα έχουμε τις εξείς παραδοχές: Το fi της κάθε νυχτερίδας-μηχανής ταυτίζεται με το Pi ,την ενέργεια δηλαδή που παράγει κάθε μηχανή.Επιπλέον γνωρίζουμε οτι στον αλγόριθμο bat η συχνότητα f κυμαίνεται σε ένα διαστήμα [fmin,fmax] όπου είναι αντίστοιχα η ελάχιστη δυνατή συχνότητα και η μέγιστη δυνατή,είναι εμφανές εδώ οτι υπάρχει αντιστοιχία με την σχέση  **, Εξίσωση 10** δηλάδη Pimin=fmin και Pimax=fmax ,οπού Pimin η ελάχιστη παραγωγή ενέργειας απο την i-μηχανή-νυχτερίδα(Li οπως το ορίσαμε παραπανω) και Pimax η μέγιστη παραγώγη ενέργειας απο την i-μηχανή-νυχτερίδα (Ui οπως το ορίσαμε παραπάνω) και αυτά τα νούμερα μας είναι ήδη γνωστά.Τα xi καθέ νυχτερίδας αντιστοιχούν στην τιμή της συνάρτησης φ που μας δίνει η i-μηχανή.Οπου η συνάρτηση φ είναι : φ=Fcost+PfEcost η συνάρτηση που θέλουμε να ελαχιστοποιήσουμε στο πρόβλημα CEED. Αφού το xi προκύπτει απο το υi  με τον τύπο ** =  + ** **Εξίσωση 4**και η συνάρτηση φ προκύπτει με την βοήθεια του Pf (penalty factor),τοτέ το υi αντιστοιχεί στο Pfi. .Καταλήγουμε εν τέλει στον παρακάτω ψευδοκώδικα :

*Θεωρώ 6 νυχτερίδες-μηχανές Μ1,Μ2,Μ3,Μ4,Μ5,Μ6*

*Καθέ νυχτερίδα έχει fi ισό με το Pi καθέ μηχανής ,ισχυέι οτι το fi παιρνει τιμές στο [fmin,fmax]=[Pimin,Pimax]=[Li,Ui]*

*To υi αντιστοιχεί στο Pfi κάθε νυχτερίδας-μηχανής*

*Τρέχει ο παρακάτω αλγόριθμος για Pd=500,700,900*

*ΑΡΧΗ*

*Υπολογίζω το PL,* 

Δοκίμασε fi (στα επιτρεπτά όρια)

Υπολόγισε Fcost,Ecost

Υπολόγισε υi (με την διαδικασία που μας δίνει το CEED)

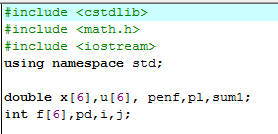
Υπολόγισε τα xi

Τέλος\_επανάληψης

ΤΕΛΟΣ

*3.3 Υλοποίηση του αλγορίθμου σε C++*

Η εργασία πραγματοποιήθηκε στον compiler ,Dev C++ 4.9.9.2.Τα απαραίτητα αρχεία θα τα βρείτε σε CD που θα συνοδεύει το έντυπο αυτό.Το project με το ονόμα Ptyxiakh,καθώς και εφαρμογή του.Το source file με το όνομα main και ενα αρχείο ο με το όνομα main.o,αλλά και ένα Makefile.win ,που είναι αρχείο WIN.Θα βρείτε επίσης διαφόρα screenshots και διαγράμματα σε Dia.Προχωράμε αμέσως σε ανάλυση του κώδικα με screenshots και επεξηγήσεις-αναλύσεις του.



Εδώ συμπεριλαμβάνουμε καποίες βιβλιοθήκες της C++ που κάνωντας την υλοποίηση μας φάνηκαν απαραίτητες.Οπώς η **cstdlib**,για το randomization των **f[i]** στοιχείων,η **math** για να χρησιμοποιήσουμε τον τύπο **pow(a,b)** και η **iostream** για εντολές εισόδου-εξόδου οπώς η **printf** και **scanf**. Η γραμμή **using namespace std;** χρησιμοποιείται σχεδόν πάντα με την βιβλιοθήκη **iostream**.

Υστερά δηλώνουμε καποίες global μεταβλητές,δύο πίνακες που είναι τύπου double και κάποιες αλλές που θα δούμε αργότερα την χρήση τους.Απο τις μεταβλητές τύπου int έχουμε τον πίνακα 6 στοιχείων f,δύο μετρητές i,j και το pd(power demand) που δίνει ο χρήστης.

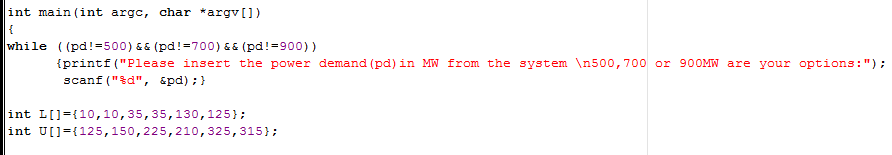
Επείτα έχουμε την συνάρτηση main ,απολύτα απαραίτητη για να «τρέχει » το προγραμμά μας.Επειδή επικαλύπτεται απο αλλά screenshots θα την δείξω αυτούσια εδώ ως κείμενο.

int main(int argc, char \*argv[])

{ return EXIT\_SUCCESS;

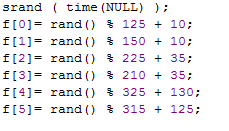
}

Εν ολίγοις παίρνει τα default ορισμάτα argc και \*argv[] τύπου int και char αντίστοιχα και επιστρέφεται η μεταβλητή ΕΧΙΤ\_SUCCESS που δίνει τον ομαλό τερματισμό του προγράμματος.

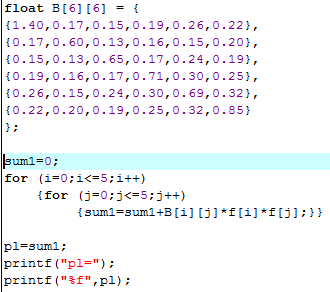


Εδώ ελέγχουμε με ενά βρόχο while οτι δεν θα δοθεί μη επιτρεπτή τιμή στο pd. Με μήνυμα ,μέσω της βοήθειας την εντολής printf ,υπενθυμίζουμε στον χρήστη τις επιτρεπτές τιμές και με την scanf παίρνουμε αυτη την τιμή.Επείτα αρχικοποιούμε τα στοιχεία των int πινάκων L και U ,που όπως είδαμε πιο πάνω αντιστοιχούν στο Pimin και Pimax των μηχανών νυχτερίδων(Πινάκας 3).

Μετέπειτα θέλωντας να αυτοματοποιήσουμε οσο είναι δυνατό την παρέμβαση του χρήστη και για να είμαστε πιο πιστοί στην φιλοσοφία του αλγόριθμου bat χρησιμοποιούμε τις συναρτήσεις srand και rand. Για να δοθούν στα στοιχεία του πίνακα που συμβολίζει την ισχύ που δουλεύει καθέ μηχανή τυχαίες τιμές ,εντός φυσικά των επιτρεπτών ορίων.



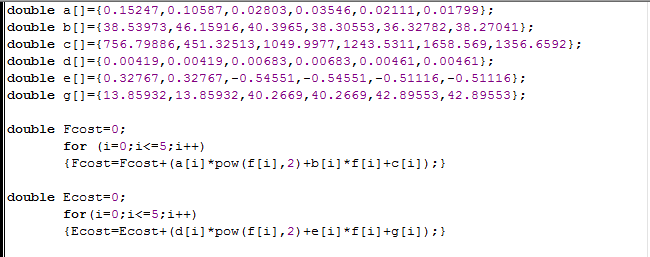
Προχωράμε στο επόμενο κομμάτι κώδικα που είναι το εξής:



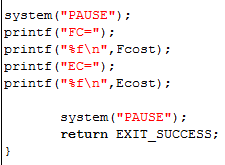
Οπού αρχικοποιούμε τις τιμές του float πίνακα Β και υπολογίζουμε το pl απο τον τύπο της εξίσωσης 11:

Υστέρα εμφανίζουμε το αποτέλεσμα.

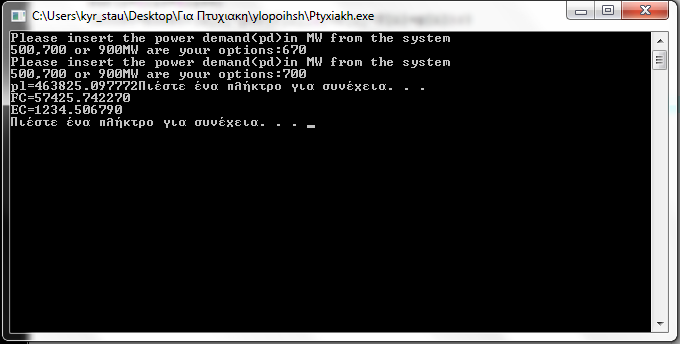
Επείτα αρχικοποιούμε τους double πίνακες a,b,c,d,e,g που αντιστοιχούν στους συντελεστές α,β,γ της Οικονομικής Κατανομής(Economic Dispatch) και a,b,c της Κατανομής Εκπομής Ρύπων(Emission Dispatch).Και υπολογίζουμε τα Fcost και Ecost για να ακολουθήσουν αργότερα συγκρίσεις αυτού του αλγορίθμου με τους αλλούς του case study που μας δόθηκε. Η υψώση στο τετράγωνο επιτυγχάνεται με την εντολή **pow(f[i],2).**



Επείτα εμφανίζουμε τα αποτελέσματά μας ως εξής:



Παράδειγμα επιτυχημένης εκτέλεσης



Παρατηρούμε οτι σην μή δεκτή τιμή 670 το πρόγραμμα μας ζήτησε ξάνα την τιμή του pd.Οταν δώσαμε την επιτρεπτή τιμή 700 τότε υπολογίστηκε το pl και εμφανιστήκε το αποτελεσμά αυτό.Βλέπουμε οτι το σύστημα ζητάει να πατήσουμε ενα πλήκτρο για συνέχεια αυτό επιτεύχθηκε με την εντολή **system(“PAUSE”);** για να προλάβει να δεί ο χρήστης τα αποτελέσματα.

Επίσης υπολογίζουμε τα u[i] (Pfi) ως εξής:

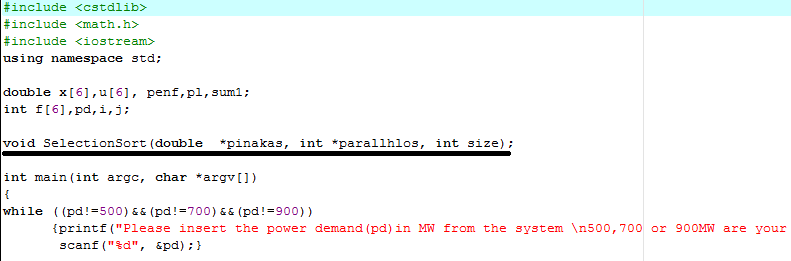


Με την βοήθειας των συντελεστών που έχουμε ήδη αρικοποιήσει και του πίνακα U πού έχουμε επίσης αρχικοποιήσει ,και την εντολή pow(U[i],2),για υψώση δύναμης στο τετράγωνο.

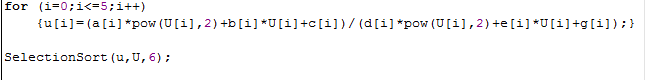
Αργότερα προσθέσαμε μια ακόμα λειτουργία στο προγραμμά μας μεσώ μιας τροποποιημένης συνάρτησης ταξινόμησης με την μέθοδο του Selection Sort(Ταξινόμηση Επιλογής),η οποία ταξινομεί ενά double array ,αλλά παράλληλα μεταθέτει τα στοιχεία του παράλληλου int πίνακα σύμφωνα με την ταξινόμηση του πρώτου πίνακα.

Για την υλοποίηση της εγίναν τα εξής:

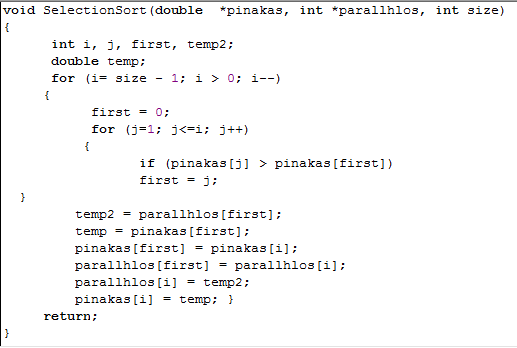
Α)Προσθήκη header στην περιοχή δηλώσεων του προγράμματος με την γραμμή **void SelectionSort(double \*pinakas, int \*parallhlos, int size);**



Β)Υστέρα την καλλούμε για τον πίνακα τύπου double u,τον παράλληλο σε αυτόν και τύπου int ,U με το μεγεθός τους(size)που είναι 6.Με την γραμμή **SelectionSort(u,U,6);**



Γ)Η ιδία η συνάρτηση,που την γράφουμε μετα το τέλος της main(για αυτό αλλώστε χρησιμοποίησα και header) εχεί ως εξής:



Ολές οι μεταβλητές που ορίζονται στην συνάρτηση αυτή είναι τοπικής εμβέλειας,δηλαδή το i,j για παράδειγμα είναι διαφορετικό απο αυτό που χρησιμοποιούμε στο main πρόγραμμα.Η συνάρτηση είναι μια κλασική Selection Sort ,μόνο που με την χρήση μίας ακόμη μεταβλητής για swap στοιχείων των πινάκων καταφέρνουμε να αντιστοιχούμε τα στοιχεία του παράλληλου πίνακα που αντιστοιχούν σε αυτόν που ταξινομούμε.Ετσί λοιπόν ο u double array θα ταξινομηθεί και τον ρόλο του παράλληλου θα τον παίξει ο U.

Ετσί λοιπόν ικανοποιείται το 5ο βήμα για τον υπολογισμό του penalty factor,η μέγιστη απόδοση της κάθε μονάδας παραγωγής (Pimax ή Ui) προστίθεται μία τη φορά,ξεκινώντας από την μονάδα του μικρότερου Pfi(ui) έως ότου ΣPimax>=Pd.Σε C++ αυτό γίνεται ως εξής:



Εφόσον με την κλήση της συνάρτησης πιο πάνω εχούμε «ταξινομήσει» τον U συμφωνά με την νέα διάταξη του u.

Εν τέλει υπολογίζουμε και τα x[i] στοιχεία που σύμφωνα με τις παραδοχές μας αντιστοιχούν στις τιμές της i-μηχανής-νυχτερίδας για την συνάρτηση φ=Fcost+Pf\*Ecost.



***4.Συγκρίσεις των αποτελεσμάτων των αλγορίθμων***

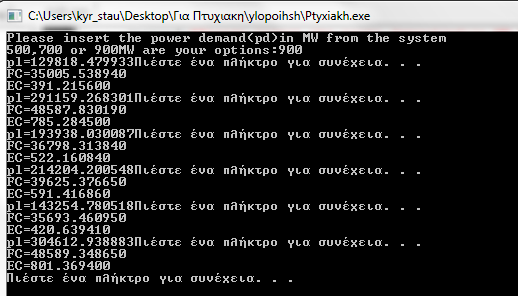
Σε αυτό το κεφάλαιο θα κάνουμε συγκρίσεις του bat αλγόριθμου που δημιουργήσαμε με διαφόρους γενετικούς αλγόριθμους ως προς το πρόβλημα επίλυσης του CEED.Οι αλγόριθμοι αυτοί που είναι μέτρο σύγκρισης και αξιολογήσης του αλγοριθμού μας είναι ο Genetic Algortihm(GRA),Hybrid Genetic Algorithm (Hybrid GA) και μία προτεινόμενη λύση,στο paper με Quadratic programming.

**Πινάκας 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Demand | Perfomance | GRA | Hybrid GA | Quadratic Solution |
| 500 | FC,Rs/hr | 27683,300 | 27695,000 | 27606,470 |
| EC,Rs/hr | 263,472 | 263,370 | 262,400 |
| Pl,MW | 10,172 | 10,135 | 8,932 |
| 700 | FC,Rs/hr | 37640,370 | 37640,400 | 37488,580 |
| EC,Rs/hr | 439,979 | 439,978 | 439,720 |
| Pl,MW | 18,521 | 18,517 | 17,054 |
| 900 | FC,Rs/hr | 48567,750 | 48567,500 | 48330,310 |
| EC,Rs/hr | 694,169 | 694,172 | 693,600 |
| Pl,MW | 29,725 | 29,718 | 28,007 |

Δεν εχούμε λοιπόν παρα να τρέξουμε τον αλγόριθμο μερικές φορές για να δούμε αν τα αποτελέσματα του είναι χειρότερα,ισάξια ή καλύτερα των αλλών αλγορίθμων.Γενικά με τον αλγοριθμό μας δεν ελαχιστοποιείται το PL, που συχνά βγαίνει αρκετά μεγάλο,αλλά ο αλγόριθμος δινεί βάρος στα Fcost(FC) και Ecost(EC).Για δική μας διευκόλυνση τρέξαμε τον κώδικα περιβάλλοντας ολο τον κώδικα εντός του main σε μία loop for(t=0;t<=5;t++) για να έχουμε 5 διαδοχικές εκτελέσεις του αλγορίθμου.

**Εκτέλεση για PD=900**



1η επανάληψη

FC=35005,538940

EC=391,215600

2η επανάληψη

FC=48587,830190

EC=785,284500

3η επανάληψη

FC=36798,313840

EC=522,160840

4η επανάληψη

FC=39625,376650

EC=591,416860

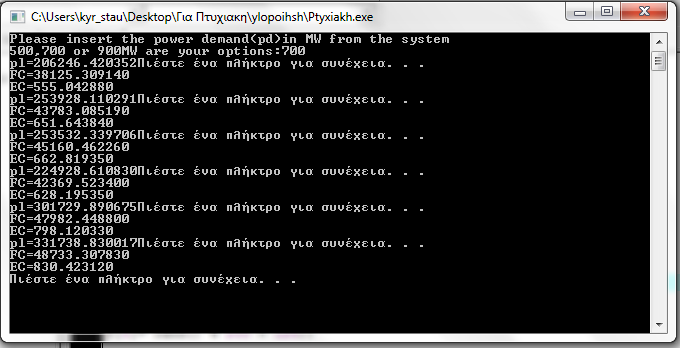
5η επανάληψη

FC=48589,348650

EC=801,369400

Συμπεραίνουμε οτί ,δεδομένου του randomize καποίων διαδικασίων του αλγορίθμου, τις περισσότερες φορές μας δίνει βέλτιστα αποτελέσματα ,ενδείκτικα είχαμε πολύ καλα αποτελέσματα,καλύτερα και απο το Quadratic Programming στις 4 απο τις 5 επαναλήψεις και στην 5η μια λύση παρόμοια των υπολοίπων με παραπάνω απο το δεκτό Εcost ,αλλα οχί σ απαγορευτικά επίπεδα.

**Εκτέλεση για PD=700**



1η επανάληψη

FC=38125,309140

EC=555,042880

2η επανάληψη

FC=43783,085190

EC=651,643840

3η επανάληψη

FC=45160,462260

EC=662,819350

4η επανάληψη

FC=47982,448800

EC=798,120330

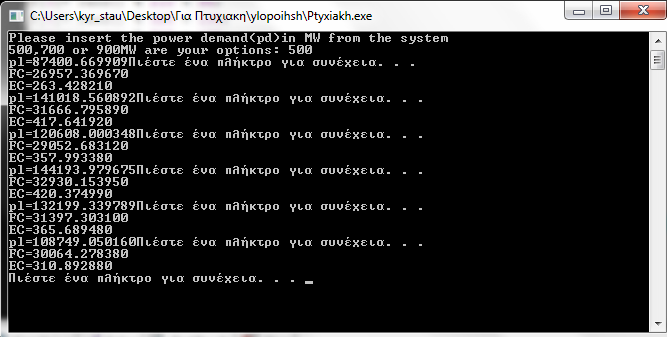
5η επανάληψη

FC=48589,348650

EC=830,423120

Συμπεραίνουμε οτι εχούμε και παλί καλά αποτελέσματα καποιές φορές λιγό χειρότερες απο τις δοσμένες λύσεις με αλλού αλγορίθμους και αλλές φορές λίγο καλύτερες,ωστόσο για μικρότερο PD μας δημιουργείται η υποψία οτι ο αλγόριθμος είναι λιγότερο αποδοτικός.

**Εκτέλεση για PD=500**



1η επανάληψη

FC=26957,369670

EC=263,428210

2η επανάληψη

FC=31666,795890

EC=357,993380

3η επανάληψη

FC=32930,153950

EC=420,374990

4η επανάληψη

FC=31397,303100

EC=365,689480

5η επανάληψη

FC=30064,278380

EC=310,892880

Σε αυτο το σημείο μαλλόν επιβεβαιώνεται οτι η ποιότητα των αποτελεσμάτων του αλγορίθμου είναι μεγαλύτερη οσο μεγαλύτερο power demand (PD),δίνουμε στο σύστημα παραγωγής ενέργειας.

Παραθέτουμε και συνολικά τον κώδικα με σχόλια:

//Program Libraries

#include <cstdlib>

#include <math.h>

#include <iostream>

using namespace std;

//Global Variables

double x[6],u[6],pl,sum1;

int f[6],pd,i,j,penf,t;

//Header

void SelectionSort(double \*pinakas, int \*parallhlos, int size);

//Begin of main

int main(int argc, char \*argv[])

{

for(t=0;t<=5;t++)

{

//pd?

while ((pd!=500)&&(pd!=700)&&(pd!=900))

{printf("Please insert the power demand(pd)in MW from the system \n500,700 or 900MW are your options:");

scanf("%d", &pd);}

//Initialize L amd U arrays

int L[]={10,10,35,35,130,125};

int U[]={125,150,225,210,325,315};

//Randomization of f array

srand ( time(NULL) );

f[0]= rand() % 125 + 10;

f[1]= rand() % 150 + 10;

f[2]= rand() % 225 + 35;

f[3]= rand() % 210 + 35;

f[4]= rand() % 325 + 130;

f[5]= rand() % 315 + 125;

//Initialize B array

float B[6][6] = {

{1.40,0.17,0.15,0.19,0.26,0.22},

{0.17,0.60,0.13,0.16,0.15,0.20},

{0.15,0.13,0.65,0.17,0.24,0.19},

{0.19,0.16,0.17,0.71,0.30,0.25},

{0.26,0.15,0.24,0.30,0.69,0.32},

{0.22,0.20,0.19,0.25,0.32,0.85}

};

//Calculation of pl

sum1=0;

for (i=0;i<=5;i++)

{for (j=0;j<=5;j++)

{sum1=sum1+B[i][j]\*f[i]\*f[j];}}

pl=sum1;

//Print pl

printf("pl=");

printf("%f",pl);

//Initialize a,b,c,d,e,g arrays

double a[]={0.15247,0.10587,0.02803,0.03546,0.02111,0.01799};

double b[]={38.53973,46.15916,40.3965,38.30553,36.32782,38.27041};

double c[]={756.79886,451.32513,1049.9977,1243.5311,1658.569,1356.6592};

double d[]={0.00419,0.00419,0.00683,0.00683,0.00461,0.00461};

double e[]={0.32767,0.32767,-0.54551,-0.54551,-0.51116,-0.51116};

double g[]={13.85932,13.85932,40.2669,40.2669,42.89553,42.89553};

//Calculation of Fcost

double Fcost=0;

for (i=0;i<=5;i++)

{Fcost=Fcost+(a[i]\*pow(f[i],2)+b[i]\*f[i]+c[i]);}

//Calculation of Ecost

double Ecost=0;

for(i=0;i<=5;i++)

{Ecost=Ecost+(d[i]\*pow(f[i],2)+e[i]\*f[i]+g[i]);}

//Calculation of u array

for (i=0;i<=5;i++)

{u[i]=(a[i]\*pow(U[i],2)+b[i]\*U[i]+c[i])/(d[i]\*pow(U[i],2)+e[i]\*U[i]+g[i]);}

//We call the SelectionSort function for u,U arrays of size=6

SelectionSort(u,U,6);

//Calculation of penalty factor

penf=0;

while (penf<pd)

{penf=penf+U[i];}

//Calculation of x array

for (i=0;i<=5;i++)

{x[i]=(a[i]\*pow(f[i],2)+b[i]\*f[i]+c[i])+penf\*(d[i]\*pow(f[i],2)+e[i]\*f[i]+g[i]);}

system("PAUSE");

//Print Fcost,Ecost

printf("FC=");

printf("%f\n",Fcost);

printf("EC=");

printf("%f\n",Ecost);

}

system("PAUSE");

return EXIT\_SUCCESS;

}

//end of main

//SelectionSort function

void SelectionSort(double \*pinakas, int \*parallhlos, int size)

{

int i, j, first, temp2;

double temp;

for (i= size - 1; i > 0; i--)

{

first = 0;

for (j=1; j<=i; j++)

{

if (pinakas[j] > pinakas[first])

first = j;

}

temp2 = parallhlos[first];

temp = pinakas[first];

pinakas[first] = pinakas[i];

parallhlos[first] = parallhlos[i];

parallhlos[i] = temp2;

pinakas[i] = temp; }

return;

}

Σημείωση: Τα σχόλια είναι στα αγγλικά γιατι δεν εμφανίζονται σωστα αλλίως,είναι πρόβλημα του Dev C++ compiler.

***Ευχαριστίες***

Τέλος αισθάνομαι την ανάγκη να ευχαριστήσω θερμά τους καθηγητές μου κυρίους Ε.Φούντα και Α.Βλάχο για την πολύτιμη βοηθεία τους και απο επιστημονική και ψυχολογική άποψη ,στην περαίωση αυτής της εργασίας.Καθότι μου έδωσαν την ευκαιρία να ασχολήθω με ένα τόσο ενδιαφέρον θέμα.

Η εργασία αυτή είναι αποτέσμα μελέτης της Βιβλιογραφίας(παρατίθεται στην επόμενη παράγραφο) που ήταν γνωστικό αντικείμενο αυτής της εργασίας.Η Βιβλιογραφία αυτή αποτελούταν τόσο απο έντυπη μορφή ,όσο και απο papers και εργασίες μελετητών στο Διαδίκτυο.

***Βιβλιογραφία***

* Ε.Χ Φούντας ,Α.Γ Βλάχος Μαθηματικά Αποφάσεων
* Ε.Χ Φούντας,Α.Χ Παναγιωτόπουλος Μαθηματικά Οικονομικών και Διοικητικών Επιστημών
* Π.Γ Ντούρμας «Εφαρμογές Μεταευριστικών Αλγόριθμων»
* Ε.Χ Φούντας Εφαρμογές Συνδυαστικής Ανάλυσης Τόμοι 1 και 2
* Ε.Χ Φούντας Ασκήσεις Μαθηματίκων Τόμοι 1 και 2
* Μ.Β Κούτρα Εισαγώγη στις Πιθανότητες : Θεωρία και Εφαρμογές Β’ Εκδόση
* Quadratic Programming Solution Emission and Economic Dispatch Problems by RMS Danaraj and Dr F Gajendran
* Nature-Inspired Algorithms ,2nd Edition by Xin-She Yang
* A New Metaheuristic Bat-Inspired Algorithm by Xin-She Yang
* An Optimized K-Means Clustering

Technique using Bat Algorithm by G. Komarasamy and Amitabh Wahi