ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ: «ΠΛΗΡΟΦΟΡΙΚΗ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ:

ΥΒΡΙΔΟΠΟΙΗΣΗ ΜΕΤΑΕΥΡΕΤΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ ΠΥΓΟΛΑΜΠΙΔΑΣ ΜΕ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ ΑΡΜΟΝΙΑΣ ΓΙΑ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

ΣΠΟΥΔΑΣΤΗΣ:

ΚΟΥΡΑΒΑΝΑΣ ΔΗΜΗΤΡΙΟΣ

(Α.Μ.: mppl11031)

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

ΑΠΟΣΤΟΛΟΥ ΔΗΜΗΤΡΙΟΣ

ΠΕΙΡΑΙΑΣ

2014

Πνευματικά δικαιώματα

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον Γιώργο και την Ελένη για την πολύτιμη βοήθεια στην υλοποίηση αυτής της εργασίας. Επίσης τους γονείς μου, Μαρία και Ηλία, που με πίστεψαν παρά τις αντιξοότητες των τελευταίων χρόνων, και τέλος την Χριστίνα για την συμπαράσταση της…

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια όλο και περισσότεροι αλγόριθμοι δημιουργούνται γύρω από τον χώρο της βελτιστοποίησης προβλημάτων καλύπτοντας ένα μεγάλο φάσμα επιστήμων κυρίως της μηχανικής. Ουσιαστικά βρισκόμαστε στην εποχή των εφευρετικών αλγορίθμων που ξεπερνούν σε απόδοση τις προκάτοχες ευρετικές μεθόδους. Αρκετοί από αυτούς είναι εμπνευσμένοι από την φύση και μιμούνται χαρακτηριστικά αυτής. Δυο βασικά παραδείγματα είναι ο αλγόριθμος Αναζήτησης Αρμονίας, που μιμείται τον τρόπο που ένας μουσικός «παίζει» μουσική και ο αλγόριθμος Πυγολαμπίδας, που μιμείται την κίνηση των πυγολαμπίδων στην φύση. Ωστόσο αρκετές φορές οι μεταευρετικοί αλγόριθμοι θα πρέπει να συνδυαστούν με κάποιον τρόπο με σκοπό να αποδώσουν καλύτερες λύσεις σε ένα πρόβλημα βελτιστοποίησης, δηλαδή να παράγουν έναν υβριδικό αλγόριθμο. Στην παρούσα εργασία θα δούμε την ικανότητα των αλγορίθμων Αναζήτησης Αρμονίας και Πυγολαμπίδας να συνδυάζονται σε έναν υβριδικό αλγόριθμο, που παράγει εξαιρετικές λύσεις σε σχέση με τους μεμονωμένους αλγορίθμους.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ACO	Βελτιστοποίηση αποικίας μυρμηγκιών
PSO	Βελτιστοποίηση σμήνους σωματιδίων
EA	Εξελικτικοί αλγόριθμοι
SA	Προσομοιωμένης ανόπτησης
HSA ή HS	Αλγόριθμος αναζήτησης αρμονίας
HM	Αρμονική μνήμη
HMCR	Δείκτης αποδοχής αρμονικής μνήμης
PAR	Δείκτης προσαρμογής του τονικού
HIS	Βελτιωμένος αλγόριθμος αναζήτησης αρμονίας
GHS	Συνολικά βέλτιστος αλγόριθμος αρμονίας
FA	Αλγόριθμος πυγολαμπίδας
B	Ελκυστικότητα μιας πυγολαμπίδας
$I_{(r)}$	Ένταση της φωτεινότητας
Γ	Συντελεστής απορρόφησης
A	Συνάρτηση πιθανότητας
HAT	Υβριδικές αλγοριθμικές τεχνικές
HAST	Υβριδικοί αλγόριθμοι αναζήτησης τεχνικών
DE	Αλγόριθμος διαφορικής εξέλιξης
HSFA	Υβριδικός αλγόριθμος πυγολαμπίδας με χρήση αλγορίθμου

αναζήτησης αρμονίας

ΕΙΣΑΓΩΓΗ

Ο σκοπός της παρούσας εργασίας είναι η υλοποίηση ενός υβριδικού μεταευρετικού αλγορίθμου με την χρήση των αλγορίθμων της Αρμονικής Αναζήτησης (Harmony Search Algorithm) και Πυγολαμπίδας (Firefly Algorithm), και η δοκιμή του πάνω σε γνωστές συναρτήσεις βελτιστοποίησης, που θα στοχεύει στην επίλυση πραγματικών προβλημάτων βελτιστοποίησης. Δυο εμπνευσμένοι από την φύση αλγόριθμοι, διαφορετικοί ως προς τον τρόπο λειτουργίας τους, θα συνδυαστούν – υβριδοποιηθούν ώστε να ενώσουν τις δυνάμεις τους προς όφελος της βελτιστοποίησης προβλημάτων. Οι συναρτήσεις βελτιστοποίησης ονομάζονται Benchmark και αποτελούν ίσως τις πιο γνωστές συναρτήσεις δοκιμών των μεταευρετικών αλγορίθμων. Αυτές οι συναρτήσεις περιέχουν ένα σύνολο ακρότατων τόσο τοπικών όσο και ολικών , που είναι αρκετά δύσκολα να υπολογιστούν από απλές ευρετικές τεχνικές. Αν και οι μεμονωμένοι αλγόριθμοι, Αναζήτησης Αρμονίας και Πυγολαμπίδας, δίνουν ικανοποιητικές λύσεις, παρουσιάζουν αρκετά προβλήματα, όπως ο εγκλωβισμός σε περιοχές με τοπικά ακρότατα , αφήνοντας ανεύρετες καλύτερες λύσεις άλλων περιοχών . Για αυτόν τον λόγο ο συνδυασμός των αλγορίθμων σε έναν υβριδικό αλγόριθμο, θα μπορούσε να παράγει καλύτερες λύσεις σε μικρότερο υπολογιστικό χρόνο, εκμεταλλευόμενος τις αρετές αυτών. Αυτός είναι ο κύριος σκοπός της παρούσας εργασίας. Τα πειραματικά αποτελέσματα που έδωσε ο υβριδικός αλγόριθμος ήταν πολύ καλύτερα από τα αποτελέσματα των μεμονωμένων αλγορίθμων . Ας δούμε τώρα ποια είναι η δομή της εργασίας.

Στο πρώτο κεφάλαιο θα ξεκινήσουμε μια ανασκόπηση γύρω από τον χώρο της βελτιστοποίησης προβλημάτων, ώστε να γίνουν σαφείς κάποιες βασικές έννοιες, και θα εισαχθούμε στον τομέα των ευρετικών και κυρίως μεταευρετικών αλγορίθμων. Η βελτιστοποίηση προβλημάτων αποτελεί ένα σημαντικό ζήτημα σε πολλές επιστήμες όπως τα μαθηματικά, η μηχανική, η οικονομία, η πληροφορική κλπ. Ο κλάδος της βελτιστοποίησης χωρίζεται σε διαφορετικά είδη και η αναφορά όλων δεν αποτελεί στόχο της παρούσας εργασίας. Στόχος αυτής είναι να γίνει μια αναφορά των δυο σημαντικών ειδών βελτιστοποίησης· της συνδυαστικής βελτιστοποίησης και της πολυκριτηριακής βελτιστοποίησης στόχων. Στην συνέχεια θα ακολουθήσει μια παρουσίαση των ευρετικών αλγορίθμων και κυρίως των τεχνικών που χρησιμοποιούν για την επίλυση προβλημάτων. Εξέλιξη των ευρετικών αλγορίθμων αποτελούν οι μεταευρετικοί. Σε αυτή την κατηγορία ανήκουν και οι αλγόριθμοι Αναζήτησης Αρμονίας και Πυγολαμπίδας. Γι’ αυτό το λόγο θα κάνουμε μια ανασκόπιση γύρω από τους μεταευρετικούς αλγορίθμους, τον τρόπο λειτουργίας τους, τα χαρακτηριστικά τους και το πώς συμπεριφέρονται σε προβλήματα βελτιστοποίησης.

Στο δεύτερο και τρίτο κεφάλαιο αντίστοιχα, θα ακολουθήσει παρουσίαση δυο εμπνευσμένων από την φύση μεταευρετικών αλγορίθμων που αποτελούν τα κύρια δομικά στοιχεία της εργασίας. Αυτοί θα είναι ο αλγόριθμος Αναζήτησης Αρμονίας και ο αλγόριθμος Πυγολαμπίδας. Θα παρουσιαστούν τα χαρακτηριστικά τους, οι ιδιότητες τους, ο τρόπος λειτουργίας τους και άλλα χρήσιμα στοιχεία. Ωστόσο αυτοί οι αλγόριθμοι δεν μοιάζουν μεταξύ τους και έχουν εντελώς διαφορετική λογική στην επίλυση προβλημάτων. Το κρίσιμο σημείο θα είναι το «πώς» μπορούν αυτοί οι δυο αλγόριθμοι να συνδυαστούν με κάποιον τρόπο ώστε να δημιουργήσουν έναν υβριδικό αλγόριθμο για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης.

Στο τέταρτο κεφάλαιο θα αναφερθούμε σε ένα πολύ σημαντικό ζήτημα, την υβριδοποίηση μεταευρετικών αλγορίθμων. Οι τεχνικές υβριδοποίησης αποτελούν το μείζον θέμα της εργασίας και η απόλυτη κατανόηση τους είναι σημαντική για τον αναγνώστη. Ως εκ τούτου θα παρουσιαστούν οι υβριδικές τεχνικές που εφαρμόζονται για την κατασκευή των μεταευρετικών αλγορίθμων. Τέλος και αφού έχουμε σχηματίσει μια πιο ξεκάθαρη εικόνα γύρω από τους αλγορίθμους Αναζήτησης Αρμονίας και Πυγολαμπίδας, θα δούμε παραδείγματα υβριδοποίησης αυτών με χρήση άλλων μεταευρετικών αλγορίθμων και θα συγκρίνουμε τα αποτελέσματα που παράγουν σε σχέση με τους μεμονωμένους μεταευρετικούς αλγορίθμους.

Στο πέμπτο κεφάλαιο, που είναι το τελευταίο και κύριο κομμάτι της παρούσας εργασίας, θα γίνει αναφορά στην υβριδοποίηση των μεταευρετικών αλγορίθμων και η υλοποίηση του πειραματικού μοντέλου που θα συνδυάζει τους αλγορίθμους Αναζήτησης Αρμονίας και Πυγολαμπίδας. Ο πειραματικός αλγόριθμος θα υλοποιηθεί στην προγραμματιστική γλώσσα Matlab και θα παρουσιαστούν χρήσιμα συμπεράσματα γύρω από την λειτουργία του.

Στο σημείο αυτό θα παρουσιαστούν κάποιες βασικές εισαγωγικές έννοιες σχετικά με την βελτιστοποίηση και τους μεταευρετικούς αλγορίθμους.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

1.1 ΑΛΓΟΡΙΘΜΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Οι θετικές επιστήμες και επί το πλείστον η επιστήμη των μαθηματικών ανέπτυξε τον τομέα της βελτιστοποίησης ο οποίος παρουσιάζει υψηλό ενδιαφέρον σε πολλά επιστημονικά πεδία, ένα από αυτά είναι και η βελτιστοποίηση αλγορίθμων. Πολλά δύσκολα προς λύση προβλήματα στους τομείς της πληροφορικής, της ιατρικής, της οικονομίας κλπ χρησιμοποιούν αλγορίθμους βελτιστοποίησης. Κάποιοι από αυτούς, που εξελίχθηκαν τα τελευταία χρόνια, ονομάζονται μεταευρετικοί και μιμούνται φαινόμενα που υπάρχουν στην φύση. Στην συνέχεια της εργασίας θα δούμε δύο παραδείγματα τέτοιων αλγορίθμων, τον αλγόριθμο της Αναζήτησης Αρμονίας και τον αλγόριθμο της Πυγολαμπίδας. Ας δούμε πρώτα την έννοια της βελτιστοποίησης.

1.1.1 ΚΛΑΣΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Η βελτιστοποίηση ουσιαστικά αποσκοπεί, ανάλογα ως προς το επίλυση πρόβλημα, στην εκμετάλλευση όλων των δυνατοτήτων των δεδομένων του προβλήματος, στην μέγιστη αποδοτικότητα της βέλτιστης λύσης και στον βέλτιστο συνδυασμό αντικρουόμενων απαιτήσεων. Στην πληροφορική η έννοια της βελτιστοποίησης αφορά την εύρεση της βέλτιστης λύσης από ένα σύνολο εφικτών λύσεων σε ένα μαθηματικά ορισμένο πρόβλημα. Το βασικό ζητούμενο είναι η μεγιστοποίηση ή ελαχιστοποίηση συναρτήσεων του προς εξέτασιν προβλήματος. Αυτή η διαδικασία κινείται κυρίως ως προς την εύρεση των τιμών των παραμέτρων ελέγχου που καθορίζουν την λειτουργία ενός συστήματος. *Με βάση αυτό, μπορούμε να πούμε ότι βελτιστοποίηση είναι η διαδικασία εύρεσης της καλύτερης αποδοτικά λύσης σε συγκεκριμένα προβλήματα που περιγράφονται με μαθηματικό τρόπο.* Ο όρος της βελτιστοποίησης ανταποκρίνεται στην ελαχιστοποίηση ή μεγιστοποίηση μιας συνάρτησης ενός προβλήματος. Εν ολίγης είναι τα προβλήματα όπου αναζητούμε το ολικό ελάχιστο ή μέγιστο μιας πολύπλοκης συνάρτησης με πληθώρα ακρότατων, μέσα σε έναν διάστημα.

Πολλές φόρες οι λύσεις των προβλημάτων βελτιστοποίησης πρέπει να υπακούουν σε περιορισμούς και συνθήκες. Η μορφή της συνάρτησης παίζει καθοριστικό ρολό στην επίλυση του προβλήματος. Προβλήματα μη συνεχή ή μη παραγωγίσιμα δυσκολεύουν την εύρεση λύσης. Ουσιαστικά αυτές οι περιπτώσεις είναι ισοδύναμες διότι η ελαχιστοποίηση της συνάρτησης f αποτελεί μεγιστοποίηση της συνάρτησης -f αντίστοιχα. Συμπεραίνουμε ότι με μια αλλαγή πρόσημου μεταφερόμαστε από ένα πρόβλημα μεγιστοποίησης σε ένα πρόβλημα ελαχιστοποίησης και αντίθετα. Άρα χωρίς βλάβη της γενικότητας, το πρόβλημα ελαχιστοποίησης (minimization problem) ορίζεται ως:

Δοθείσης μια συνάρτησης f: S→R,

Βρες ένα x^* ϵ S τέτοιο ώστε

1. *(*αν πρόκειται για πρόβλημα ελαχιστοποίησης*)*

- ✓ $f(x^*) \leq f(x), \forall x \in S$

2. (αν πρόκειται για πρόβλημα μεγιστοποίησης)

- ✓ $f(x^*) \geq f(x), \forall x \in S$

όπου S είναι ο χώρος αναζήτησης και ένα υποσύνολο του R^n.

Αν $S=R^n$ θα έχουμε ένα πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης χωρίς περιορισμούς (unconstrained minimization problem). Ωστόσο σε πολλά προβλήματα οι μεταβλητές πρέπει να πληρούν ορισμένες συνθήκες τότε ονομάζουμε αυτά τα προβλήματα ως προβλήματα βελτιστοποίησης με περιορισμούς (constrained minimization problems), και προφανώς παρουσιάζουν μεγαλύτερη δυσκολία από τα προβλήματα χωρίς περιορισμούς.

Η συνάρτηση f ονομάζεται **αντικειμενική συνάρτηση ή συνάρτηση κόστους**. Η συνάρτηση θα ορίζεται από ένα σύνολο ανεξάρτητων μεταβλητών και θα κρίνει την καταλληλότητα των λύσεων, δηλαδή αν μια λύση, είναι καλή ή όχι. Η ελάχιστη ή μέγιστη τιμή της αντικειμενικής συνάρτησης ενός προβλήματος, ανάλογα αν πρόκειται για πρόβλημα μεγιστοποίησης ή ελαχιστοποίησης, αντιπροσωπεύει την βέλτιστη λύση του προβλήματος. Το σημείο x^* ονομάζεται **ολικός ελαχιστοποιητής** (global minimum) της f. Αν για ένα σημείο $x^* \in S$ ισχύει ότι $f(x^*) \leq f(x), \forall x \in B$, όπου $B=\{ x \in S : d(x, x^*) \leq \varepsilon\}$, με d μια παραμετρική απόσταση και ε μια θετική τιμή τότε το x καλείται **τοπικός ελαχιστοποιητής** (local minimizer) της f, και η τιμή του $f' = f(x')$.

Οι αλγόριθμοι επίλυσης προβλημάτων βελτιστοποίησης χωρίζονται σε δυο κατηγορίες, τους τοπικούς και ολικούς αλγόριθμους. Οι τοπικοί αλγόριθμοι εγγυώνται την εύρεση λύσης ενός τοπικού ελαχίστου της αντικειμενικής συνάρτησης, εκκινώντας από ένα δοθέν αρχικό σημείο ενώ οι ολικοί αλγόριθμοι εγγυώνται την εύρεση ενός ολικού ελαχιστοποιητή μιας αντικειμενικής συνάρτησης, χωρίς την χρήση αρχικού σημείου. Ωστόσο πολλοί ολικοί αλγόριθμοι μπορούν να εντοπίσουν ένα τοπικό ελάχιστο χωρίς την χρήση αρχικού σημείου. Για αυτόν τον λόγο οι ολικοί αλγόριθμοι καλούνται και αλγόριθμοι ευρείας σύγκλισης (globally convergent), και απέδειξαν ότι με την χρήση κατάλληλου αρχικού σημείου, μπορούν να προσδιορίσουν ολικά ελάχιστα, χωρίς να υπάρχει κάποιος γνωστός κανόνας.

Η βελτιστοποίηση είναι ένας συνδυασμός θεωρητικής ανάλυσης με πρακτική εφαρμογή, διότι χρησιμοποιούμε παράλληλα μαθηματική θεωρία και ανάλυση, αλλά και κατασκευή, υλοποίηση, έλεγχο και επιβεβαίωση πρακτικών μεθόδων και αλγορίθμων για την αντιμετώπιση συγκεκριμένων προβλημάτων. (Dennis J. E.,Schnabel., R. B.1983[1])

1.1.2 ΣΥΝΔΥΑΣΤΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Η Συνδυαστική Βελτιστοποίηση αποτελεί ένα κλάδο της Βελτιστοποίησης στα Εφαρμοσμένα Μαθηματικά και στην Επιστήμη των Υπολογιστών, ο οποίος συνενώνει πολλά άλλα επιστημονικά πεδία, όπως της Επιχειρησιακής Έρευνας, της Αλγοριθμικής Θεωρίας, της Υπολογιστικής Πολυπλοκότητας αλλά και της Τεχνητής Νοημοσύνης. Αλγόριθμοι Συνδυαστικής Βελτιστοποίησης εφαρμόζονται για την επίλυση προβλημάτων που θεωρούνται αρκετά δύσκολα, λόγω του μεγάλου πλήθους δυνατών συνδυασμών–λύσεων. Η επίλυση των προβλημάτων Συνδυαστικής Βελτιστοποίησης επιτυγχάνεται με μεθόδους που επιτρέπουν τη μείωση του συνόλου των εφικτών λύσεων, αλλά και με μια περισσότερο αποτελεσματική αναζήτηση.

Οι αλγόριθμοί συνδυαστικής βελτιστοποίησης χρησιμοποιούνται κυρίως σε NP-HARD προβλήματα. Αυτά τα προβλήματα είναι από τα πιο δύσκολα της κατηγορίας τους ως προς την επίλυση και ένα από τα δυσκολότερα θέματα από υπολογιστικής άποψης. Ωστόσο,

---

[1] σελ.1 -373

κάποια «στιγμιότυπα» μικρού μεγέθους αυτών των προβλημάτων μπορούν να επιλυθούν σε πραγματικό χρόνο.

Πολλά προβλήματα βελτιστοποίησης ασχολούνται με την εύρεση της καλύτερης σύνθεσης του συνόλου μεταβλητών ώστε να πετύχουν έναν στόχο. Αυτά τα προβλήματα χωρίζονται σε δυο κατηγορίες. Αυτά που έχουν πραγματικές τιμές μεταβλητών και αυτά που έχουν διακριτές τιμές μεταβλητών. Η τελευταία κατηγορία προβλημάτων ονομάζεται **προβλήματα συνδυαστικής βελτιστοποίησης** (Combinatorial Optimization (CO) Problems). Τα προβλήματα αυτά αναζητούν λύσεις σε ένα πεπερασμένο ή, ενδεχομένως, αριθμήσιμα άπειρο σύνολο.

Ένα πρόβλημα P=(S, f) συνδυαστικής βελτιστοποίησης μπορεί να οριστεί ως:

- ένα σύνολο μεταβλητών $X=\{x_1,x_2,\ldots..,xn\}$
- μεταβλητών διαστημάτων $D_1,D_2,\ldots Dn$
- μια αντικειμενική συνάρτηση προς ελαχιστοποίηση f, όπου

✓ $f: D1 \times D2\times\ldots.\times D$

✓ Το σύνολο όλων των εφικτών αναθέσεων είναι

✓ $S = \{s=\{(x_1,v_1),\ldots.,(x_n,v_n)\}\,|\, v_i \in D_i, s$ και **ικανοποιεί όλους τους περιορισμούς.**

Το S συνήθως καλείται αναζήτηση λύσεων ή χώρων, αφού το κάθε στοιχείο του συνόλου μπορεί να θεωρηθεί ως μια καθορισμένη λύση.

Για την επίλυση ενός προβλήματος συνδυαστικής βελτιστοποίησης θα πρέπει να βρεθεί μια λύση, ένα στιγμιότυπο ή ένα ζεύγος *<S, f>*, όπου *S* είναι ένα πεπερασμένο σύνολο λύσεων και $f : S \rightarrow R$ μια συνάρτηση κόστους των λύσεων (που πρέπει να μεγιστοποιηθεί ή να ελαχιστοποιηθεί). Ζητάμε μια λύση $s* \in S$ τέτοια ώστε

$$f(s^*) \leq f(s),\ \forall s \in S$$

αν πρόκειται για πρόβλημα ελαχιστοποίησης ή

$$f(s^*) \geq f(s),\ \forall s \in S$$

αν πρόκειται για πρόβλημα μεγιστοποίησης.

Οι αλγόριθμοι που χρησιμοποιούνται στην συνδυαστική βελτιστοποίηση ανήκουν στις κατηγορίες των πλήρων ή κατά προσέγγιση αλγορίθμων. Οι πλήρεις προσπαθούν να βρουν μια βέλτιστη λύση σε ένα πεπερασμένο σύνολο λύσεων σε καθορισμένο χρόνο. Ωστόσο για NP – hard προβλήματα δεν υπάρχει πολυωνυμικός χρόνος για την εύρεση λύσης. Επομένως, οι πλήρεις μέθοδοι μπορεί να χρειάζονται εκθετικό χρόνο υπολογισμού στη χειρότερη περίπτωση. Αυτό συχνά οδηγεί σε χρόνους υπολογισμού πολύ υψηλούς. Στους κατά προσέγγιση αλγορίθμους δεν υπάρχει εγγύηση εύρεσης βέλτιστης λύσης, αλλά εγγυώνται την εύρεση της βέλτιστης λύσης σε μικρό χρόνο αναζήτησης. Μεταξύ των βασικών μεθόδων ανάλυσης κατά προσέγγιση συνήθως γίνεται διάκριση μεταξύ δημιουργικών μεθόδων και μεθόδων τοπικής αναζήτησης. Οι δημιουργικοί αλγόριθμοι παράγουν λύσεις από το μηδέν και τις προσθέτουν σε ένα αρχικά άδειο σύνολο, μέχρις ότου μια λύση θα είναι πλήρης. Αυτή είναι συνήθως μια πιο γρήγορη προσέγγιση, όμως συχνά επιστρέφονται λύσεις κατώτερης ποιότητας σε σύγκριση με τους αλγόριθμους τοπικής αναζήτησης.

Οι αλγόριθμοι τοπικής αναζήτησης ξεκινούν από κάποια αρχική λύση και επαναληπτικά προσπαθούν να αντικαταστήσουν την τρέχουσα λύση με μια καλύτερη λύση σε μια κατάλληλα ορισμένη δομή γειτονιάς της τρέχουσας λύσης, όπου η γειτονιά επισήμως ορίζεται ως εξής:

Μια **δομή γειτονιάς** είναι η συνάρτηση N: S → 2*S που αντιστοιχεί κάθε s ∈ S ένα σύνολο γειτονιών N(s) που ανήκουν στο S. Το N(s) καλείται γειτονιά του s.

Η καθιέρωση μιας δομής γειτονιάς μας επιτρέπει να ορίσουμε την έννοια της τοπικής ελάχιστης λύσης.

Μια τοπική ελάχιστη λύση στην δομή γειτονιάς N είναι μια ελάχιστη λύση s τέτοια ώστε για κάθε $s \in N(s^*)$: $f(s^*) \le f(s)$. Το s* καλείται* **αυστηρά τοπικά ελάχιστη λύση** *αν $f(s^*) < f(s)$ για κάθε $s \in N(s^*)$.*

Τα πρακτικά προβλήματα που συναντάμε είναι προβλήματα της κλάσης NP-complete. Η κλάση αυτή περιλαμβάνει προβλήματα με την ιδιότητα ότι αν υπάρχει αλγόριθμος πολυωνυμικού χρόνου για κάποιο από αυτά, τότε υπάρχει τέτοιος αλγόριθμος και για όλα τα υπόλοιπα. Μέχρι στιγμής δεν έχει βρεθεί πολυωνυμικός αλγόριθμος για κανένα πρόβλημα και εικάζεται ότι κάτι τέτοιο δεν πρόκειται να συμβεί. Λόγω όμως της κρισιμότητας των εφαρμογών στις οποίες εμφανίζονται, πρέπει να είμαστε σε θέση να βρίσκουμε λύσεις για αυτά, παρόλη την δυσκολία επίλυσής τους. Αυτό μπορούμε να το κάνουμε με διάφορους τρόπους. Ένας από αυτούς είναι να επιταχύνουμε τους υπάρχοντες ακριβείς αλγόριθμους. Για τα περισσότερα από τα προβλήματα που μας ενδιαφέρουν υπάρχουν ακριβείς αλγόριθμοι εξαντλητικής αναζήτησης. Μπορούμε να τους επιταχύνουμε με τις εξής τεχνικές:

- Διακλάδωσε και Φράξε (branch and bound), που επιτρέπει να αποκόψουμε τμήμα του δέντρου αναζήτησης υπολογίζοντας φράγματα για τη βέλτιστη τιμή σε κάθε κόμβο,

- Δυναμικό Προγραμματισμό (dynamic programming), που κάποιες φορές δίνει ψευδοπολυωνυμικούς αλγόριθμους.

- Να περιορίσουμε τις εισόδους. Στις αποδείξεις για την NP-πληρότητα (NP-completeness) λαμβάνουμε υπόψη τη χειρότερη περίπτωση των εισόδων ενός προβλήματος. Συχνά όμως οι είσοδοι που εμφανίζονται στην πράξη είναι πιο καλές, επιτρέποντάς μας να λύσουμε το πρόβλημα πιο εύκολα. Για παράδειγμα, παρόλο που το πρόβλημα SAT είναι δύσκολο προς επίλυση, για το 2SAT, που είναι μια εξειδίκευση του προηγούμενου, υπάρχει κάποιος απλός αλγόριθμος πολυωνυμικού χρόνου που το επιλύει.

- Να βρούμε προσεγγιστικές λύσεις. Συχνά συναντάμε NP-complete προβλήματα επειδή θέλουμε ακριβείς λύσεις. Αν όμως από κάποιο πρόβλημα ζητάμε απλώς μια καλή λύση και τίποτα παραπάνω, ενδέχεται να μπορούμε να το λύσουμε με κάποιον αλγόριθμο πολυωνυμικού χρόνου. Για παράδειγμα, υπάρχουν οι αλγόριθμοι Απληστίας (Greedy) που δίνουν μια προσεγγιστική λύση στο πρόβλημα SET COVER.

- Να χρησιμοποιήσουμε ευριστικές τεχνικές. Κάποιες φορές δε μπορούμε να κάνουμε απόλυτες προσεγγίσεις, μπορούμε όμως να σχεδιάσουμε αλγόριθμους που φαίνεται να δουλεύουν καλά στην πράξη και να δικαιολογήσουμε το γιατί δουλεύουν καλά. Οι πιο συνηθισμένες ευριστικές μέθοδοι κάνουν τοπική αναζήτηση στο χώρο των λύσεων του προβλήματος, καταλήγοντας σε μια τοπικά βέλτιστη λύση. Το μειονέκτημά τους είναι ότι δε μπορούν εύκολα να συγκριθούν. Για το λόγο αυτό έχουν προταθεί διάφορα είδη ανάλυσης όπως η εμπειρική, η ανάλυση στη μέση και στη χειρότερη περίπτωση. Ο αλγόριθμος γραμμικού προγραμματισμού Simplex για παράδειγμα είναι εκθετικός στη χειρότερη περίπτωση, στην πράξη όμως δουλεύει καλά και έχει επικρατήσει.

- Να χρησιμοποιήσουμε τυχαιότητα. Το πρόβλημα με τους ακριβείς αλγόριθμους για NP-complete προβλήματα είναι ότι πρέπει κάθε φορά να εξετάσουν εκθετικό (ως προς το μέγεθος της εισόδου τους) πλήθος περιπτώσεων πριν καταλήξουν στη λύση, κάτι που οδηγεί σε εκθετική πολυπλοκότητα. Αυτό μπορούμε να το αποφύγουμε αν τους επιτρέψουμε να κάνουν κάποια πράγματα τυχαία. Οι αλγόριθμοι που προκύπτουν λέγονται πιθανοτικοί (randomized ή probabilistic). Μέχρι στιγμής δεν έχουμε καταφέρει να λύσουμε NP-complete προβλήματα χρησιμοποιώντας τυχαιότητα και αυτό μάλλον δεν πρόκειται να γίνει ποτέ. Όμως η τυχαιότητα είναι χρήσιμη όταν σχεδιάζουμε προσεγγιστικούς και ευριστικούς αλγόριθμους. Επίσης, αν υποθέσουμε ότι η είσοδος ενός προβλήματος προέρχεται από κατάλληλη τυχαία κατανομή, μπορούμε να σχεδιάσουμε αλγόριθμο που να δουλεύει καλά στη μέση περίπτωση. (Β. Ζησιμόπουλος, 2007[2])

1.1.3 ΠΟΛΥΚΡΙΤΗΡΙΑΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Τα πολυκριτηριακα προβλήματα βελτιστοποίησης περιέχουν αντικρουόμενους στόχους, οι οποίοι εμποδίζουν την ταυτόχρονη βελτιστοποίηση τους. Πολλά προβλήματα της μηχανικής στην πραγματικότητα είναι πολυκριτηριακά, δηλαδή μεγιστοποιούν ή ελαχιστοποιούν το κόστος τους, την απόδοση τους κλπ.

Υπάρχουν δυο γενικές προσεγγίσεις πολυκριτηριακής βελτιστοποίησης. Η μια αναφέρεται στον συνδυασμό της αντικειμενικής συνάρτησης σε μια ενιαία σύνθετη συνάρτηση, αλλά μόνο ένα κριτήριο-στόχος θα ανήκει στο σύνολο περιορισμών. Ο καθορισμός ενός ενιαίου κριτηρίου αποτελεί μια απαραίτητη ενέργεια. Ωστόσο αυτό στην πράξη αποτελεί ένα δύσκολο εγχείρημα, λόγο της δυσκολίας εύρεσης ακρίβειας. Η κλιμάκωση μεταξύ των κριτηρίων είναι αναγκαία και κάποιες διαταραχές των τιμών μερικές φορές μπορεί να οδηγήσουν σε διαφορετικές λύσεις. Τελικά το πρόβλημα είναι η συσχέτιση των κριτηρίων στο σύνολο των περιορισμών, οι τιμές των περιορισμών θα πρέπει να καθοριστούν για το καθένα από τα αρχικά κριτήρια, ακόμα και αυθαίρετα. Σε όλες τις περιπτώσεις η μέθοδος βελτιστοποίησης θα πρέπει να παράγει μια λύση και όχι ένα σύνολο λύσεων που θα

[2] σελ. 3 - 5

εξετάζεται για την καταλληλότητα του. Γι' αυτόν τον λόγο συχνά προτιμάται μια σειρά από καλές λύσεις.

Η δεύτερη προσέγγιση παρέχει ένα σύνολο βέλτιστων λύσεων Pareto, ή ένα αντιπροσωπευτικό υποσύνολο του. Ένα σύνολο βέλτιστων λύσεων Pareto είναι μια σειρά από τις καλύτερες λύσεις σε σχέση με τις υπόλοιπες, ενώ η μετακίνηση από μια λύση Pareto σε μια άλλη περιέχει πάντα μια συνολική απώλεια κριτηρίου(-ων) ώστε να επιτευχθεί ένα συνολικό κέρδος για τις υπόλοιπες. Το σύνολο βέλτιστων λύσεων Pareto προτιμάται συχνά σε μεμονωμένες λύσεις, διότι μπορούν να είναι πρακτικές στα πραγματικά προβλήματα, εφόσον η τελική τιμή της απόφασης τερματισμού δεν θα είναι η χειρότερη. Το σύνολο Pareto μπορεί να έχει διαφορετικό μέγεθος ανάλογα με την αύξηση του αριθμού κριτηρίων.

Αν για μια τιμή απόφασης τερματισμού σε ένα πολυκριτηριακό πρόβλημα, θέλουμε να βελτιστοποιήσουμε K στόχους (όπου οι στόχοι είναι ασύμμετροι και η λήψη απόφασης δεν δείχνει σαφή προτίμηση κάποιου στόχου σε σχέση με κάποιον άλλον), χωρίς απώλεια της γενικότητας, όλοι οι στόχοι που βρίσκονται στον τύπο ελαχιστοποίησης μπορούν να τροποποιηθούν σε έναν τύπο μεγιστοποίησης, με πολλαπλασιασμό του αρχικού με μείων ένα. Η πολυκριτηριακή βελτιστοποίηση προβλημάτων με K στόχους ορίζεται ως εξής:

Δοθέντος n-διάστατου διανύσματος μεταβλητής απόφασης $X=\{x_1,y,x_n\}$, στο διάστημα λύσης X, βρείτε ένα διάνυσμα x* που ελαχιστοποιεί ένα δεδομένο σύνολο K αντικειμενικών συναρτήσεων $z(x^*) = \{z_1(x^*),y,z_K(x^*)\}$.

Ο χώρος λύσεων X γενικά περιορίζεται από ένα σύνολο περιορισμών, όπως ο $g_j(x^*) = b_j$ για $j=1,\ldots,m$ και από τα όρια σχετικά με την απόφαση των μεταβλητών.

Η βελτιστοποίηση ενός και μόνο στόχου οδηγεί συχνά σε απαράδεκτα αποτελέσματα, σε σχέση με τους άλλους στόχους. Ωστόσο μια τέλεια πολυκριτηριακή λύση είναι ανέφικτη. Μια λογική λύση σε ένα πολυκριτηριακο πρόβλημα είναι να διερευνήσει ένα σύνολο λύσεων, κάθε μια από τις οποίες ικανοποιεί ένα στόχο σε ένα αποδεκτό επίπεδο χωρίς να κυριαρχείται από οποιαδήποτε άλλη λύση.

Αν όλες οι αντικειμενικές συναρτήσεις είναι προς ελαχιστοποίηση, μια εφικτή λύση x λέμε ότι είναι κυρίαρχη από μια άλλη y (x>y), αν και μόνο αν $z_i(x) \leq z_i(y)$ για $i=1,\ldots,K$ και άρα $z_j(x) < z_j(y)$ για τουλάχιστον μια αντικειμενική συνάρτηση j. Μια λύση λέμε ότι θα είναι ένα βέλτιστο Pareto αν δεν κυριαρχείται από άλλες λύσεις στο χώρο λύσεων. Ένα βέλτιστο Pareto δεν μπορεί να βελτιστοποιηθεί σε σχέση με οποιονδήποτε άλλο στόχο, αν δεν είναι χειρότερο από αυτόν. Το σύνολο όλων των εφικτών μη κυριαρχούμενων λύσεων στον χώρο λύσεων X αναφέρεται ως σύνολο βέλτιστων Pareto και για ένα δεδομένο σύνολο βέλτιστων Pareto, οι αντίστοιχες τιμές τις αντικειμενικής συνάρτησης στον χώρο στόχου θα ονομάζεται μέτωπο Pareto. Για πολλά προβλήματα, ο αριθμός των βέλτιστων Pareto είναι τεράστιος, ίσως και άπειρος.

Ο απώτερος σκοπός της πολυκριτηριακής βελτιστοποίησης αλγορίθμων είναι να ερευνά λύσεις σε ένα σύνολο βέλτιστων Pareto. Ωστόσο η αναζήτηση σε ολόκληρο το σύνολο, για πολλά πολυκριτηριακά προβλήματα είναι πρακτικά αδύνατη, λόγο του μεγέθους του. Σε αντίθεση με πολλά προβλήματα συνδυαστικής βελτιστοποίησης, η απόδειξη της βέλτιστης λύσης είναι υπολογιστικά ανέφικτη. Για αυτό μια πρακτική προσέγγιση για πολυκριτηριακή βελτιστοποίηση είναι η αναζήτηση μιας σειράς λύσεων που αντιπροσωπεύει όσο το δυνατόν καλύτερα το βέλτιστο Pareto (καλά γνωστό σύνολο Pareto). Σχετικά με αυτά τα σημαντικά ζητήματα, μια πολυκριτηριακή μέθοδος βελτιστοποίησης θα πρέπει να πετύχει τους εξής τρείς αντικρουόμενους στόχους:

1. Το καλά γνωστό μέτωπο Pareto θα πρέπει να είναι όσο το δυνατόν πλησιέστερα στο πραγματικό μέτωπο Pareto. Στην ιδανική περίπτωση το πιο γνωστό σύνολο Pareto θα πρέπει να είναι ένα υποσύνολο του βέλτιστου συνόλου Pareto.

2. Οι λύσεις στο πιο γνωστό σύνολο Pareto θα πρέπει να είναι ομοιόμορφα κατανεμημένες και διαφοροποιημένες από το μέτωπο Pareto ώστε να παρέχεται μια λήψη απόφασης που θα είναι μια πραγματική εικόνα των βέλτιστων λύσεων.

3. Το γνωστό μέτωπο Pareto θα πρέπει να καλύπτει όλο το φάσμα του μετώπου Pareto. Αυτό απαιτεί αναζήτηση λύσεων στα άκρα του χώρου της αντικειμενικής συνάρτησης.

Για ένα δεδομένο όριο υπολογιστικού χρόνου, ο πρώτος στόχος είναι η καλύτερη εξυπηρέτηση με την εντατικοποίηση στην έρευνα μιας συγκεκριμένης περιοχής του μετώπου Pareto. Αντίθετα, ο δεύτερος στόχος απαιτεί η προσπάθεια αναζήτησης να είναι ομοιόμορφα κατανεμημένη μέσα στο μέτωπο Pareto . Ο τρίτος στόχος αποσκοπεί στην επέκταση του μετώπου Pareto στα δυο άκρα, ώστε να γίνει διερεύνηση για νέες ακραίες λύσεις. (Konaka et. al.[3] (2006))

1.1.4 ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Ένα στιγμιότυπο ενός προβλήματος συνδυαστικής βελτιστοποίησης είναι ένα ζεύγος $<S, f>$, όπου S είναι ένα πεπερασμένο σύνολο λύσεων και $f : S \rightarrow R$ μια συνάρτηση κόστους των λύσεων (που πρέπει να μεγιστοποιηθεί ή να ελαχιστοποιηθεί).

Ζητάμε μια λύση $s^* \in S$ τέτοια ώστε $f(s^*) \leq f(s)$, $\forall s \in S$ αν πρόκειται για πρόβλημα ελαχιστοποίησης ή $f(s^*) \geq f(s)$, $\forall s \in S$ αν πρόκειται για πρόβλημα μεγιστοποίησης.

Για την επίλυση των NP-πλήρων (NP-complete) προβλημάτων έχουμε στη διάθεσή μας γενικές μεθόδους, που μπορούν να εφαρμοστούν σε οποιοδήποτε πρόβλημα, αλλά και ειδικές, που εξαρτώνται άμεσα από το πρόβλημα για του οποίου την επίλυση σχεδιάζονται. Οι γενικές μέθοδοι συμπεριλαμβάνουν τους ακριβείς (exact) αλγόριθμους και την τοπική αναζήτηση (local search) με τις επεκτάσεις και τα μεταευριστικά της (meta-heuristics). Στις ειδικές περιλαμβάνονται τα ευριστικά (heuristics) και οι προσεγγιστικοί (approximation) αλγόριθμοι.

Στην κλάση NP ανήκουν πολλά θεμελιώδη προβλήματα της Συνδυαστικής Βελτιστοποίησης. Τα γραφοθεωρητικά προβλήματα βελτιστοποίησης, τα προβλήματα σχετικά με τη βέλτιστη τοποθέτηση εργασιών σε παράλληλες αρχιτεκτονικές, το πρόβλημα της τετραγωνικής ανάθεσης (quadratic assignment) και τα προβλήματα που έχουν να κάνουν με δίκτυα διασύνδεσης. Οι ακριβείς αλγόριθμοι μπορούν να χρησιμοποιηθούν για την επίλυση στιγμιότυπων περιορισμένου μεγέθους (που εξαρτάται από το πρόβλημα). Ο χρόνος εκτέλεσής τους ενδέχεται να είναι πολύ μεγάλος. Γενικά δεν είναι εύκολο να παραλληλοποιηθούν.

Τα ευριστικά έχουν αποδεκτό χρόνο εκτέλεσης για στιγμιότυπα μεγαλύτερου μεγέθους. Δίνουν μια προσέγγιση της ζητούμενης λύσης και είναι συγκεκριμένα για το πρόβλημα για το οποίο φτιάχνονται. Και αυτά δεν είναι εύκολο να παραλληλοποιηθούν. Οι προσεγγιστικοί

---

[3] σελ. 992-993

αλγόριθμοι δίνουν μια εγγύηση της ποιότητας των λύσεων που βρίσκουν. Μπορούμε να ορίσουμε και πλήρως πολυωνυμικά προσεγγιστικά σχήματα (fully polynomial-time approximation schemes), τα οποία βρίσκουν μια ε-προσεγγιστική λύση σε χρόνο πολυωνυμικό ως προς το μέγεθος της εισόδου και το 1/ε. Οι μέθοδοι τοπικής αναζήτησης είναι γενικές. Το βέλτιστο που βρίσκουν είναι τοπικό (και ενδέχεται να απέχει πολύ από το ολικό). Προκειμένου να τις εφαρμόσουμε χρειάζεται να έχουμε στη διάθεσή μας μια αρχική εφικτή λύση, που δεν είναι πάντα το ίδιο εύκολο να υπολογιστεί. Η δομή της γειτονιάς καθώς και ο τρόπος με τον οποίο γίνεται η αναζήτηση εξαρτώνται από το πρόβλημα. Η αρχική λύση επηρεάζει την ποιότητα του βέλτιστου.

Η προσομοιωμένη ανόπτηση (simulated annealing) είναι επίσης μια γενική τεχνική. Συγκρινόμενη με την τοπική αναζήτηση πλεονεκτεί στο ότι έχει τη δυνατότητα να μην παγιδεύεται σε τοπικά βέλτιστα. Δίνει λύσεις καλής ποιότητας και έχει μικρό χρόνο εκτέλεσης. Είναι ασυμπτωτικά βέλτιστη, δηλαδή αν την αφήσουμε να εκτελείται επ' άπειρον θα βρει τη βέλτιστη λύση. Κατά συνέπεια η προσέγγιση που θα πετύχουμε εξαρτάται και από το χρόνο εκτέλεσης.

1.2 ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Ο όρος ευρετικός χρησιμοποιείται για τους αλγόριθμους που βρίσκουν λύσεις μεταξύ όλων των πιθανών λύσεων ενός προβλήματος, αλλά δεν εγγυώνται ότι η καλύτερη λύση θα βρεθεί, επιλέγουν την πιο κοντινή λύση στην ακριβή αλλά όχι την ακριβή. Συνήθως βρίσκουν μια λύση κοντά στην καλύτερη , γρήγορα και εύκολα. Μερικές φορές μπορεί να είναι ακριβείς, δηλαδή θα βρουν την πραγματικά καλύτερη λύση, αλλά ο αλγόριθμος εξακολουθεί να ψάχνει για άλλες καλές λύσεις μέχρι να αποδειχθεί ότι βρήκε την καλύτερη όλων.

<u>ΟΡΙΣΜΟΣ</u>

Αλγόριθμοι που είτε δίνουν σχεδόν τη σωστή λύση ή παρέχουν μια λύση όχι για όλες τις περιπτώσεις του προβλήματος ονομάζονται ευρετικοί αλγόριθμοι.

Τα σύγχρονα προβλήματα τείνουν να είναι πολύ περίπλοκα και σχετίζονται με την ανάλυση μεγάλων συνόλων δεδομένων. Ακόμη και αν ένας αλγόριθμος μπορεί να αναπτυχθεί με ακρίβεια, η πολυπλοκότητα χρόνου ή χώρου μπορεί να αποδειχθεί απαράδεκτη. Στην πραγματικότητα αρκεί συχνά να βρεθεί μια προσέγγιση ή μια μερική λύση. Μια τέτοια παραδοχή επεκτείνει το σύνολο των τεχνικών για να αντιμετωπίσουν το πρόβλημα. Οι ευρετικοί αλγόριθμοι παρέχουν κάποιες προσεγγίσεις για την επίλυση των προβλημάτων βελτιστοποίησης. Σε τέτοια προβλήματα ο στόχος είναι να βρεθεί η βέλτιστη όλων των πιθανών λύσεων, που είναι αυτή που ελαχιστοποιεί ή μεγιστοποιεί την αντικειμενική συνάρτηση. Η αντικειμενική συνάρτηση είναι μια συνάρτηση που χρησιμοποιείται για την αξιολόγηση της ποιότητας του παραγόμενου συνόλου λύσεων. Η συλλογή όλων των πιθανών λύσεων για ένα δεδομένο πρόβλημα μπορεί να θεωρηθεί ως ένας χώρος αναζήτησης, και οι αλγόριθμοι βελτιστοποίησης, με τη σειρά τους, συχνά αναφέρονται ως αλγόριθμοι αναζήτησης.

Οι αλγοριθμικές προσεγγίσεις παράγουν ενδιαφέροντα θέματα σχετικά με την εκτίμηση της ποιότητας των λύσεων που βρίσκουν. Λαμβάνοντας υπόψη το γεγονός ότι συνήθως η βέλτιστη λύση είναι άγνωστη, το πρόβλημα αυτό μπορεί να είναι μια πραγματική πρόκληση που περιέχει ισχυρές μαθηματικές αναλύσεις. Σε σχέση με την ποιότητα του ευρετικού αλγορίθμου, ο στόχος είναι να βρεθεί όσο το δυνατόν η καλύτερη λύση για όλες τις περιπτώσεις του προβλήματος.

Οι βασικές αρχές που ακολουθούν οι παραδοσιακοί αλγόριθμοι αναζήτησης είναι οι:

- **Εξαντλητική Αναζήτηση** (Exhaustive search**)**. Η απλούστερη των αλγορίθμων αναζήτησης είναι η εξαντλητική αναζήτηση που προσπαθεί να βρει κάθε δυνατή λύση από ένα προκαθορισμένο σύνολο λύσεων και στη συνέχεια επιλέγει την καλύτερη αυτών.

- **Τοπική αναζήτηση** (Local search). Η Τοπική αναζήτηση είναι μια διεξοδική αναζήτηση που επικεντρώνεται μόνο σε μια περιορισμένη περιοχή του χώρου αναζήτησης. Η τοπική αναζήτηση μπορεί να οργανωθεί με διαφορετικούς τρόπους. Οι τεχνικές hill-climbing, που θα αναφέρουμε πιο κάτω, ανήκουν σε αυτή την κατηγορία. Τέτοιοι αλγόριθμοι αντικαθιστούν σταθερά την τρέχουσα λύση με μια καλύτερη από τις γειτονικές, στην περίπτωση που κάποια από αυτές είναι καλύτερη από την τρέχουσα.

- **Διαίρει και Βασίλευε** (Divide and Conquer). Οι Διαίρει και Βασίλευε αλγόριθμοι προσπαθούν να χωρίσουν ένα πρόβλημα σε μικρότερα προβλήματα που είναι πιο εύκολα να επιλυθούν . Οι λύσεις των μικρών προβλημάτων θα πρέπει να συνδυάζονται σε μια λύση για το αρχικό πρόβλημα . Οι τεχνικές αυτές είναι αποτελεσματικές, αλλά η χρήση τους περιορίζεται επειδή δεν υπάρχει μεγάλος αριθμός προβλημάτων που να μπορούν εύκολα να κατανέμονται και να συνδυάζονται με τέτοιο τρόπο.

- **Διαμερισμός και Φραγή** (Branch-and-bound). Οι αλγόριθμοι αυτής της κατηγορίας κάνουν μια κρίσιμη απαρίθμηση του χώρου αναζήτησης. Απαριθμούν, αλλά παράλληλα προσπαθούν συνεχώς να αποκλείσουν τμήματα του χώρου αναζήτησης που δεν μπορούν να περιέχουν την καλύτερη λύση.

- **Δυναμικός Προγραμματισμός (**Dynamic programming). Οι αλγόριθμοι δυναμικού προγραμματισμού κάνουν μια διεξοδική έρευνα που αποφεύγει εκ νέου υπολογισμούς αποθηκεύοντας τις λύσεις των υποπροβλημάτων. Το βασικό σημείο για την χρησιμοποίηση αυτής της τεχνικής είναι ότι διαμορφώνει τη λύση μέσα από μια διαδικασία αναδρομής.

- **Άπληστοι αλγόριθμοι** (Greedy Algorithms). Μια δημοφιλής τεχνική είναι η άπληστη τεχνική, η οποία προσπαθεί διαδοχικά να κατασκευάσει ένα χώρο λύσεων. Βασίζεται στην αρχή της προφανής λήψης της καλύτερης (τοπικά) επιλογής σε κάθε στάδιο του αλγορίθμου, προκειμένου να βρει το ολικό βέλτιστο της αντικειμενικής συνάρτησης. Οι αλγόριθμοι απληστίας προσπαθούν να οδηγήσουν σε μια εφικτή λύση του προβλήματος, αλλά πολλές φορές χρειάζονται πάρα πολύ μεγάλο χρόνο, διότι λαμβάνουν υπόψη μόνο τις νέες λύσεις.

1.2.1 ΕΥΡΕΤΙΚΕΣ ΤΕΧΝΙΚΕΣ

Οι τεχνικές Διαμερισμού και Φραγής και Δυναμικού προγραμματισμού είναι αρκετά αποτελεσματικές, αλλά η χρονική τους πολυπλοκότητα είναι συχνά πολύ υψηλή και μη αποδεκτή για NP-complete προβλήματα. Ο αλγόριθμος Hill-climbing είναι αποτελεσματικός, αλλά έχει ένα σημαντικό μειονέκτημα που ονομάζεται πρόωρη σύγκλιση. Ενώ η τεχνική Απληστίας, βρίσκει πάντα το πλησιέστερο τοπικό βέλτιστο, που όμως είναι χαμηλής ποιότητας λύσης. Ο στόχος των σύγχρονων ευρετικών αλγορίθμων είναι να ξεπεραστούν αυτά τα μειονεκτήματα, γι' αυτόν τον λόγο τα τελευταία χρόνια παρουσιάστηκαν νέοι αλγόριθμοι που χρησιμοποιούν την ευρετική λογική. Οι βασικότεροι από αυτούς είναι:

- Η τεχνική **Αναρρίχηση Λόφου** (Hill-Climbing) είναι μια τοπική τεχνική αναζήτησης,η οποία κινείται από την μια λύση στην άλλη, σε μια γειτονιά λύσεων. Εάν η ποιότητα της νέας λύσης είναι καλύτερη από την προηγούμενη, αυτή η κίνηση γίνεται αποδεκτή και η αναζήτηση συνεχίζεται από την νέα λύση. Αν μια γειτονιά δεν οδηγεί σε μια βελτίωση λύση, η κίνηση αυτή θα απορριφτεί και η αναζήτηση συνεχίζεται από την τρέχουσα κατάσταση. Το κύριο μειονέκτημα αυτής της μεθόδου είναι ότι η διαδικασία αναζήτησης ενδέχεται να παγιδευτεί σε ένα τοπικό ελάχιστο, το οποίο δεν είναι το ολικό, δηλαδή η βέλτιστη λύση του προβλήματος. Μια χρήσιμη παραλλαγή για απλή αναρρίχηση λόφων θεωρεί μια σειρά κινήσεων από την τρέχουσα κατάσταση και επιλέγει τη καλύτερη λύση ως την επόμενη γειτονιά. Αυτή η μέθοδος είναι γνωστή ως αναζήτηση κλίσης ή απότομη ανάβαση λόφου-αναρρίχηση.

- Ο αλγόριθμος **Προσομοιωμένης Ανόπτησης** (Simulated annealing), εφευρέθηκε το 1983 και χρησιμοποιεί μια προσέγγιση παρόμοια με τον αλγόριθμο Αναρρίχησης Λόφου, αλλά περιστασιακά δέχεται λύσεις που είναι χειρότερες απ' ότι η τρέχουσα. Η πιθανότητα μιας τέτοιας αποδοχής μειώνεται με το χρόνο.

- Ο αλγόριθμος **Αναζήτησης Tabu** (Tabu search) επεκτείνει την ιδέα να αποφευχθούν τα τοπικά βέλτιστα με τη χρήση δομών μνήμης. Το πρόβλημα της Προσομοιωμένης Ανόπτησης είναι ότι ο αλγόριθμος επαναλαμβάνει απλά το δικό του κομμάτι καθώς μεταπηδά από την μια γειτονιά στην άλλη. Η Αναζήτηση Tabu απαγορεύει την επανάληψη των κινήσεων που έχουν γίνει πρόσφατα. Αυτό για παράδειγμα, μπορεί να το καταφέρει με ένα κριτήριο τερματισμού, χρησιμοποιώντας έναν αριθμό επαναλήψεων ή έναν καθορισμένο αριθμό επαναλήψεων που δεν θα επηρεάζουν την καλύτερη τιμή της αντικειμενικής συνάρτησης. Επίσης σταματά σε οποιαδήποτε επανάληψη όταν δεν υπάρχουν επόμενες κινήσεις στην τοπική γειτονιά.

- Ο αλγόριθμος **Νοημοσύνης Σμήνους** (Swarm intelligence), εισήχθη το 1989. Πρόκειται για μια τεχνική τεχνητής νοημοσύνη, με βάση τη μελέτη της συλλογικής συμπεριφοράς των αποκεντρωμένων, αυτο-οργανωμένων συστημάτων. Δύο από τις πιο επιτυχημένες μορφές αυτής της προσέγγισης είναι η **Βελτιστοποίηση Αποικίας Μυρμηγκιών** (Ant Colony Optimization ACO) και η **Βελτιστοποίηση Σμήνους Σωματιδίων** (Particle Swarm Optimization PSO). Στην ACO, τεχνητά μυρμήγκια χτίζουν λύσεις με την κίνηση τους στο γράφημα ενός προβλήματος και οι αλλαγές γίνονται με τέτοιο τρόπο ώστε μελλοντικά τα μυρμήγκια να μπορούν να δημιουργήσουν καλύτερες λύσεις. Η PSO ασχολείται με προβλήματα στα οποία μια καλύτερη λύση μπορεί να αναπαρασταθεί ως ένα σημείο ή μια επιφάνεια σε ένα ν-διάστατο χώρο. Το κύριο πλεονέκτημα της είναι ότι οι πληροφορίες είναι εντυπωσιακά ανθεκτικές στα τοπικά βέλτιστα ενός προβλήματος.
- Οι **Εξελικτικοί Αλγόριθμοι** (Evolutionary Algorithms) παρουσιάζουν επιτυχία στην αντιμετώπιση της πρόωρης σύγκλισης λαμβάνοντας υπόψη μια σειρά από λύσεις ταυτόχρονα . (N. Maaroju &[4] D. Garg (2008))

1.3 ΜΕΤΑΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Ο όρος μετευρετικός αλγόριθμος χρησιμοποιήθηκε πρώτη φορά από τον Glover το 1986. Οι μετευρετικοί αλγόριθμοι εφαρμόζουν γενικές στρατηγικές αναζήτησης, οι οποίες συνήθως είναι εμπνευσμένες από τις φυσικές διαδικασίες. Οι αλγόριθμοι αυτοί καθοδηγούν κατάλληλα ευρετικές μεθόδους με σκοπό να διερευνήσουν το ευρύτερο τμήμα του χώρου λύσεων του προβλήματος. Οι μετευρετικοί αλγόριθμου που αναπτύχθηκαν τα τελευταία χρόνια, διερευνούν λύσεις σε μη εφικτές περιοχές ή ακόμη και σε περιοχές όπου η αντικειμενική συνάρτηση έχει τις χειρότερες λύσεις. Σκοπός αυτών των κινήσεων είναι η αποφυγή του εγκλωβισμού του αλγορίθμου σε τοπικά ελάχιστα, επιτυγχάνοντας λύσεις οι οποίες είναι πλησιέστερα στις βέλτιστες.

---

[4] σελ. 537 -547

Μια Μεταευρετική Τεχνική επίσημα ορίζεται ως μια επαναληπτική διαδικασία που συνδυάζει έξυπνα διαφορετικές έννοιες για την εξερεύνηση και την εκμετάλλευση του χώρου αναζήτησης. Μαθαίνοντας στρατηγικές που χρησιμοποιούνται για την δομή των πληροφοριών, μπορούν να βρουν σχεδόν αποτελεσματικά βέλτιστες λύσεις. Η βασική επαναληπτική διαδικασία που καθοδηγεί τον αλγόριθμο, τροποποιεί τις ευρετικές τεχνικές για την αποτελεσματική παραγωγή υψηλής ποιότητας λύσεων. Αυτή μπορεί να χειριστεί ένα πλήρες (ή μη πλήρες) σύνολο ή μια συλλογή λύσεων σε κάθε επανάληψη. Οι δευτερεύουσες ευρετικές τεχνικές που χρησιμοποιούνται μπορεί να είναι υψηλού ή χαμηλού επίπεδου, για παράδειγμα μια απλή τοπική αναζήτηση, ή απλά μια μέθοδος κατασκευής.

Ο κύριος στόχος των μεταευρετικών είναι να αποφευχθούν τα μειονεκτήματα της επαναληπτικής βελτίωσης και ειδικότερα να μην επιτρέπει στην τοπική αναζήτηση να ξεφεύγει από τα τοπικά βέλτιστα. Αυτό επιτυγχάνεται είτε επιτρέποντας κινήσεις από τον ένα χώρο αναζήτησης στον άλλο, είτε δημιουργώντας νέες αρχικές λύσεις για την τοπική αναζήτηση με έναν πιο έξυπνο τρόπο απ' το να παράγει τυχαίες αρχικές λύσεις. Πολλές από τις μεθόδους μπορεί να εισάγουν μια αρχική λύση που θεωρούμε ότι θα είναι κοντά στην βέλτιστη, έτσι ώστε οι λύσεις υψηλής ποιότητας να παράγονται γρήγορα. Αρκετές μεταευρετικές προσεγγίσεις βασίζονται σε πιθανολογικές αποφάσεις που λαμβάνονται κατά τη διάρκεια της αναζήτησης. Όμως, η βασική διαφορά στην καθαρή δειγματοληπτική έρευνα είναι ότι οι μεταευρετικοί αλγόριθμοι χρησιμοποιούν την τυχαιότητα, όχι τυφλά αλλά με μια έξυπνη προκατειλημμένη μορφή. Ένας μεταευρετικός αλγόριθμος είναι ένα σύνολο εννοιών που μπορούν να χρησιμοποιούνται για να καθορίσουν μια ευρετική μέθοδο και μπορούν να εφαρμοστούν σε ένα ευρύ σύνολο διαφορετικών προβλημάτων. Με άλλα λόγια, ένας μεταευρετικός αλγόριθμος μπορεί να θεωρηθεί, σε γενικό πλαίσιο, ως ένας αλγόριθμος που μπορεί να εφαρμοστεί σε διαφορετικά προβλήματα βελτιστοποίησης, με σχετικά λίγες τροποποιήσεις, ώστε να προσαρμοσθεί σε ένα συγκεκριμένο πρόβλημα.

ΙΔΙΟΤΗΤΕΣ

1. Είναι στρατηγικές που κατευθύνουν τη διαδικασία αναζήτησης.
2. Ο στόχος είναι να διερευνηθεί αποτελεσματικά ο χώρο αναζήτησης για να βρεθούν οι βέλτιστες λύσεις.

3. Τεχνικές που αποτελούν μεταευρετικούς αλγόριθμούς μπορεί να είναι από απλές τοπικές διαδικασίες αναζήτησης μέχρι και πολύπλοκες μαθηματικές διαδικασίες.
4. Οι μεταευρετικοί αλγόριθμοι είναι κατά προσέγγιση συνήθως Μη-Ντετερμινιστικοί.
5. Μπορούν να ενσωματώσουν μηχανισμούς για να αποφεύγεται η παγίδευση σε κλειστούς χώρους του χώρου αναζήτησης.
6. Οι βασικές έννοιες τους επιτρέπουν την περιγραφή ενός αφηρημένου επίπεδου.
7. Οι μεταευρετικοί αλγόριθμοι δεν επιλύουν ειδικά προβλήματα.
8. Μπορούν να κάνουν χρήση των συγκεκριμένων τομέων γνώσης με τη μορφή των ευρετικών τεχνικών που ελέγχονται από το ανώτερο επίπεδο στρατηγικής.
9. Οι μεταευρετικές τεχνικές χρησιμοποιούν την εμπειρία αναζήτησης (που ενσωματώνεται με κάποια μορφή μνήμης) για την καθοδήγηση της αναζήτησης.

Μεγάλη σημασία έχει ότι η δυναμική ισορροπία δίνεται μεταξύ της ***διαφοροποίησης*** και της ***εντατικοποίησης***. Ο όρος διαφοροποίηση γενικά αναφέρεται στην εξερεύνηση του χώρου αναζήτησης, ενώ ο όρος της εντατικοποίησης αναφέρεται στην εκμετάλλευση της συσσωρευμένης εμπειρίας αναζήτησης. Στην πραγματικότητα, οι έννοιες της διαφοροποίησης και της εντατικοποίηση συχνά αναφέρονται σε μάλλον βραχυπρόθεσμες στρατηγικές που συνδέονται με την τυχαιότητα, λαμβάνοντας υπόψη επίσης ότι η εντατικοποίηση και η διαφοροποίηση αναφέρονται σε μεσοπρόθεσμες και μακροπρόθεσμες στρατηγικές που βασίζονται στην χρήση της μνήμης. Η ισορροπία μεταξύ της διαφοροποίησης και εντατικοποίησης, όπως αναφέρθηκε ανωτέρω, είναι σημαντική διότι από τη μία πλευρά εντοπίζονται γρήγορα οι περιφέρειες στο χώρο αναζήτησης με υψηλή ποιότητα λύσεων και από την άλλη δεν χάνεται πολύς χρόνος σε περιοχές του χώρο αναζήτησης που είτε έχουν ήδη διερευνηθεί είτε δεν παρέχουν υψηλής ποιότητας λύσεις.

Υπάρχουν πολλές εμφανείς διαφορετικές φιλοσοφίες στους μεταευρετικούς αλγορίθμους. Μερικές από αυτές μπορούμε να τις δούμε ως έξυπνες επεκτάσεις της τοπικής αναζήτησης αλγορίθμων. Ο στόχος αυτού του είδους των μεταευρετικών αλγορίθμων είναι να ξεφύγουν από τα τοπικά ελάχιστα προκειμένου να προχωρήσουν στην εξερεύνηση του χώρου αναζήτησης και να ανακαλύψουν ολικά βέλτιστα. Τέτοια παραδείγματα είναι οι Αναζητήσεις Tabu, καθώς και η Προσομοιωμένη Ανόπτηση.

Οι μεταευρετικές τεχνικές (ονομάζονται επίσης μέθοδοι τροχιών) εργάζονται σε μια ή περισσότερες δομές γειτονιάς που επιβάλλουν τα μέλη (λύσεις) τους σε αναζήτησεις στον χώρο. Μπορούμε να βρούμε μια διαφορετική φιλοσοφία αλγόριθμων όπως οι αλγόριθμοι Βελτιστοποίησης Αποικίας Μυρμηγκιών (Ant Colony Optimization) και οι Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms). Αυτοί προσπαθούν να βρουν συσχετίσεις μεταξύ των μεταβλητών απόφασης για τον εντοπισμό περιοχών υψηλής ποιότητας λύσεων στο χώρο αναζήτησης. Κατά μία έννοια, εκτελούν μια προκατειλημμένη δειγματοληψία του χώρου αναζήτησης. Για παράδειγμα, στους Εξελικτικούς Αλγορίθμους αυτό επιτυγχάνεται με ανασυνδυασμό των λύσεων και στους αλγόριθμους Βελτιστοποίησης Αποικίας Μυρμηγκιών με δειγματοληψία στο χώρο αναζήτησης σε κάθε επανάληψη σύμφωνα με μια κατανομή πιθανοτήτων.

1.3.1 ΚΑΤΑΤΑΞΗ ΤΩΝ ΜΕΤΑΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Υπάρχουν διάφοροι τρόποι για την ταξινόμηση και περιγραφή των μεταευρετικών αλγορίθμων. Διαφοροποιούνται μεταξύ τους ανάλογα με τα χαρακτηριστικά τους. Είναι δυνατές αρκετές ταξινομήσεις, η κάθε μια από αυτές είναι αποτέλεσμα μιας συγκεκριμένης άποψης. Οι πιο διαδεδομένες κατηγορίες μεταευρετικών αλγορίθμων είναι οι εξείς:

- ✓ **Εμπνευσμένοι από τη φύση σε σχέση με μη - εμπνευσμένους από τη φύση**. Ο τρόπος ταξινόμησης βασιζόμενος στην κύρια προέλευση του αλγορίθμου. Υπάρχουν εμπνευσμένοι από τη φύση αλγόριθμοι, όπως οι Γενετικοί Αλγόριθμοι, και μη εμπνευσμένοι από τη φύση, όπως η Αναζήτηση Tabu. Η ταξινόμηση αυτή είναι κατά κάποιον τρόπο δυσνόητη για τους εξής λόγους. Πρώτον, πολλοί υβριδικοί αλγόριθμοι δεν ταιριάζουν σε καμία κατηγορία. Δεύτερον, μερικές φορές είναι δύσκολο να αποδοθεί σαφώς ένας αλγόριθμος σε μία από τις δύο κατηγορίες, διότι τα χαρακτηριστικά του μπορεί να μην είναι ξεκάθαρα εμπνευσμένα από την φύση. Για παράδειγμα, κάποιος μπορεί να πει ότι η χρήση μνήμης στην Αναζήτηση Tabu δεν είναι εμπνευσμένη από τη φύση.

- ✓ **Βάση πληθυσμού σε σχέση με αναζήτηση μοναδικού σημείου.** Ένα άλλο χαρακτηριστικό που μπορεί να χρησιμοποιηθεί για την κατάταξη, είναι ο αριθμός των λύσεων που χρησιμοποιούνται την ίδια στιγμή. Οι αλγόριθμοι που εργάζονται με ενιαίες λύσεις ονομάζονται μέθοδοι τροχιάς και περιλαμβάνουν τοπική αναζήτηση με βασικές μεταευρετικές τεχνικές, όπως η Αναζήτηση Tabu. Αυτοί οι αλγόριθμοι μοιράζονται μια τροχιά του χώρου αναζήτησης κατά τη διάρκεια της διαδικασία αναζήτησης. Αντίθετα, οι αλγόριθμοι με βάση τον πληθυσμό εκτελούν διαδικασίες αναζήτησης που περιγράφουν την εξέλιξη ενός συνόλου σημείων στο χώρο αναζήτησης.

- ✓ **Δυναμική λειτουργία στόχου σε σχέση με στατική λειτουργία στόχου.** Αυτοί οι αλγόριθμοι μπορούν να ταξινομηθούν με βάση τον τρόπο χρήσης της αντικειμενική συνάρτηση. Ενώ μερικοί αλγόριθμοι κρατούν την αντικειμενική συνάρτηση δίνοντας στο πρόβλημα μια εκπροσώπηση, κάποιοι άλλοι, όπως ο αλγόριθμος Guided Local Search (GLS), τροποποιούνται κατά τη διάρκεια της αναζήτησης. Η ιδέα αυτή βασίζεται στο ότι πρέπει να ξεφύγουν από τα τοπικά ελάχιστα τροποποιώντας το τοπίο αναζήτησης. Γι’ αυτόν τον λόγο, κατά τη διάρκεια της έρευνας, η αντικειμενική συνάρτηση μεταβάλλεται προσπαθώντας να ενσωματώσει πληροφορίες που συλλέγονται κατά τη διάρκεια της διαδικασία αναζήτησης.

- ✓ **Μια δομή γειτονιάς σε σχέση με πολλές δομές γειτονιών.** Οι περισσότεροι αλγόριθμοι λειτουργούν σε μία ενιαία δομή γειτονιάς. Με άλλα λόγια, η τοπολογική κατάσταση του φυσικού τοπίου δεν επηρεάζεται κατά τη διάρκεια εκτέλεσης του αλγορίθμου. Άλλοι μεταευρετικοί, όπως η Μεταβλητή Αναζήτηση Γειτονιάς (VNS), χρησιμοποιούν ένα σύνολο δομών γειτονιάς, το οποίο δίνει τη δυνατότητα να διαφοροποιείται η αναζήτηση από την εναλλαγή διαφορετικών ταιριαστών τοπίων.

- ✓ **Η χρήση μνήμης σε σχέση με μεθόδους χρήσης λιγότερης μνήμης.** Ένα πολύ σημαντικό χαρακτηριστικό για την ταξινόμηση είναι η χρήση του ιστορικού αναζήτησης, με την χρήση μνήμης ή όχι. Οι αλγόριθμοι που χρησιμοποιούν ελάχιστη μνήμη εκτελούν μια διαδικασία Markov, καθώς οι πληροφορίες που χρησιμοποιούν αποκλειστικά για τον προσδιορισμό της επόμενης ενέργειας είναι η τρέχουσα κατάσταση της διαδικασίας αναζήτησης. Υπάρχουν πολλοί διαφορετικοί τρόποι για να γίνει χρήση της μνήμης. Συνήθως υπάρχει η διάκριση μεταξύ της χρήσης της βραχυπρόθεσμης και μακροπρόθεσμης μνήμη. Η πρώτη συνήθως παρακολουθεί τις πρόσφατες κινήσεις που εκτελέσθηκαν, επισκέπτεται τις λύσεις ή, εν γένει, τις αποφάσεις που λαμβάνονται. Η δεύτερη συνήθως είναι μια συσσώρευση συνθετικών παραμέτρων για την αναζήτηση. Η χρήση της μνήμης στις μέρες μας αναγνωρίζεται ως ένα από τα θεμελιώδη στοιχεία ενός ισχυρού μεταευρετικού αλγορίθμου.

1.3.2 ΕΝΤΑΤΙΚΟΠΟΙΗΣΗ ΚΑΙ ΔΙΑΦΟΡΟΠΟΙΗΣΗ

Οι έννοιες της εντατικοποίησης και της διαφοροποίησης είναι οι ισχυρές κινητήριες δυνάμεις των μεταευρετικών αλγορίθμων που τους παρέχουν υψηλή απόδοση. Πίσω από ένα μεταευρετικό αλγόριθμο, η εντατικοποίηση και η διαφοροποίηση επιτυγχάνονται με διαφορετικούς τρόπους. Παρόλα αυτά, προτείνεται μια ένωση μεταξύ αυτών. Επιπλέον, θα μπορούσαν να οδηγήσουν στην ανάπτυξη υβριδικών αλγόριθμων καθώς συνδυάζουν έννοιες που προέρχονται από διαφορετικούς αλγορίθμους.

Κάθε προσέγγιση θα πρέπει να έχει σχεδιαστεί με στόχο την αποτελεσματική διερεύνηση ενός χώρου αναζήτησης. Η αναζήτηση η οποία πραγματοποιείται από μια μεταευρετική προσέγγιση θα πρέπει να είναι αρκετά έξυπνη για την εντατική διερεύνηση στους τομείς του χώρου αναζήτησης με υψηλής ποιότητας λύσεις και μετακίνηση σε ανεξερεύνητες περιοχές του, όταν είναι απαραίτητο. Οι έννοιες για την επίτευξη αυτών των στόχων ονομάζονται εντατικοποίηση και διαφοροποίηση.

Άλλοι αλγόριθμοι, όπως οι εξε **Πηγή: Wang & L. Guo, (2013), σελ. 2**

λικτικοί, σχετίζονται με έννοιες όπως η εκμετάλλευση (εντατικοποίηση) και η εξερεύνηση (διαφοροποίηση). Ωστόσο, οι όροι εκμετάλλευση και εξερεύνηση έχουν κάπως πιο περιορισμένο νόημα. Στην πραγματικότητα, οι έννοιες της εκμετάλλευσης και εξερεύνησης συχνά αναφέρονται σε μάλλον βραχυπρόθεσμες στρατηγικές που συνδέονται με την τυχαιότητα, ενώ η εντατικοποίηση και η διαφοροποίηση αναφέρονται σε μακροπρόθεσμες στρατηγικές που συνδέονται κυρίως στη χρήση μνήμης. Δεδομένου ότι οι διαφορετικοί τρόποι χρήσης της μνήμης γίνονται ολοένα και πιο σημαντικοί σε όλους τους τομείς των μεταευρετικών αλγορίθμων, οι όροι της εντατικοποίησης και της διαφοροποίησης όλο και περισσότερο υιοθετούνται.

Μια έμμεση αναφορά στην έννοια της «τοποθεσίας» συχνά εισάγεται όταν η εντατικοποίηση και η διαφοροποίηση εμπλέκονται. Η έννοια της «περιοχής» ή «γειτονιάς» του χώρου αναζήτησης και της «θέσης» δεν μπορεί παρά να εκφράζονται με έναν ασαφή τρόπο, όπου πάντα θα εξαρτάται από τα χαρακτηριστικά της αναζήτησης, καθώς και από τον ορισμό των μετρικών χώρων αναζήτησης (αποστάσεις μεταξύ λύσεων).

Δύο πολύ σημαντικά στοιχεία της Αναζήτηση Tabu είναι η εντατικοποίηση και οι στρατηγικές διαφοροποίησης. Επιβάλλονται στρατηγικές που βασίζονται στην τροποποίηση επιλογής κανόνων ώστε να ενθαρρύνεται ο συνδυασμός κινήσεων και τα χαρακτηριστικά μνήμης για την εύρεση μιας καλής λύσης. Μπορεί επίσης να ξεκινήσει μια κίνηση επιστροφής στις ελκυστικές περιοχές για την αναζήτηση πιο επιμελών λύσεων. Οι καλύτερες λύσεις πρέπει να καταγράφονται προκειμένου να εξεταστούν άμεσα οι γειτονιές τους. Η μνήμη συνδέεται στενά με την εφαρμογή των στρατηγικών εντατικοποίησης. Η κύρια διαφορά μεταξύ της εντατικοποίησης και της διαφοροποίησης είναι ότι κατά τη διάρκεια της αναζήτησης, ένα στάδιο της εντατικοποίησης επικεντρώνεται στην εξέταση λύσεων γύρω από τις κορυφαίες. Το στάδιο της διαφοροποίησης από την άλλη πλευρά ενθαρρύνει την διαδικασία αναζήτησης για να εξετάσει τις λύσεις που δεν έχουν επισκεφθεί στις περιφέρειες και να δημιουργήσουν λύσεις που διαφέρουν με διάφορους σημαντικούς τρόπους από εκείνες που παρατηρήθηκαν στο παρελθόν.

Η στρατηγική ταλάντωση συνδέεται στενά με την προέλευση της αναζήτησης Tabu και παρέχει ένα τρόπο ώστε να επιτευχθεί μια αποτελεσματική αλληλεπίδραση μεταξύ εντατικοποίησης και διαφοροποίησης. Μετά από μια τοπική ελαχιστοποίηση συναντάμε όλα τα σημεία στη λεκάνη έλξης να μην έχουν την δυνατότητα βελτιστοποίησης. Η αναζήτηση θα πρέπει να αποφεύγει την σπατάλη υπερβολικών υπολογισμών κάθε φορά σε μια ενιαία λεκάνη και η διαφοροποίηση θα πρέπει να ενεργοποιείται. Από την άλλη πλευρά, στις παραδοχές ότι οι γειτονικές περιοχές συσχετίζουν τις τιμές της αντικειμενικής συνάρτησης, κάποια προσπάθεια θα πρέπει να δαπανηθεί στην αναζήτηση των καλύτερων σημείων που βρίσκονται κοντά στα πιο πρόσφατα σημεία που βρέθηκαν από τα τοπικά ελάχιστα (εντατικοποίηση). Οι δύο απαιτήσεις είναι αντικρουόμενες για την εξεύρεση της κατάλληλης ισορροπίας της διαφοροποίησης και της εντατικοποίησης και αποτελούν ένα κρίσιμο ζήτημα στους μεταευρετικούς αλγορίθμους.

Μια μεταευρετική μέθοδος είναι επιτυχής σε ένα δεδομένο πρόβλημα βελτιστοποίησης, εάν μπορεί να παρέχει μια ισορροπία μεταξύ της εκμετάλλευσης, δηλαδή των συσσωρευμένων αναζητήσεων, και της εξερεύνησης του χώρου αναζήτησης για τον εντοπισμό περιοχών με υψηλής ποιότητας λύσεις σε ένα πρόβλημα με βέλτιστο τρόπο. Η εντατικοποίηση ψάχνει προσεκτικά και εντατικά γύρω από τις καλές λύσεις που έχουν βρεθεί στο παρελθόν από την διαδικασία αναζήτησης. Η διαφοροποίηση αντιθέτως, καθοδηγεί την αναζήτηση σε περιοχές που δεν έχουν επισκεφθεί. Συνοψίζοντας, διαφορετικές ιδέες μεταευρετικών αλγορίθμων θα πρέπει να λαμβάνουν υπόψη τους αυτές τις δυο έννοιες και οι μεταευρετικοί αλγόριθμοι θα πρέπει να είναι σχεδιασμένοι έτσι ώστε η εντατικοποίηση και η διαφοροποίηση να παίζουν ισορροπημένους ρόλους.

1.3.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕΤΑΕΥΡΕΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Για να ξεπεραστεί το μειονέκτημα των αλγόριθμων τοπικής αναζήτησης, δηλαδή η αδυναμία να ξεφεύγουν από τα τοπικά ελάχιστα, δημιουργήθηκαν οι μεταευρετικοί αλγόριθμοι, όπου είναι πιο εξελιγμένες στρατηγικές αναζήτησης με ευρετική σχεδίαση. Αυτό συνεπάγεται την προσωρινή αποδοχή μιας τεχνικής κατάστασης χαμηλότερης ποιότητας.

Παρακάτω παραθέτονται κάποιοι βασικοί μεταευρετικοί αλγόριθμοι:

1. **Γενετικοί Αλγόριθμοι** (Evolution Algorithm). Οι Γενετικοί Αλγόριθμοι αναζήτησης και βελτιστοποίησης λειτουργούν παρόμοια με τις εξελικτικές διαδικασίες που παρατηρούνται στη φύση. Η έρευνα τους κατευθύνθηκε προς τη βελτιστοποίηση που χρησιμοποιεί την «αρχή της επιβίωση του ισχυρότερου». Αυτό επιτυγχάνεται με την επιλογή των πιο επιθυμητών χαρακτηριστικών από μια γενιά λύσεων και ο συνδυασμός τους για να σχηματίσουν την επόμενη γενιά. Η ποιότητα της κάθε λύσης αξιολογείται από τη «φυσική κατάσταση», του κάθε άτομου που έχει επιλεγεί για τη διαδικασία της αναπαραγωγής. Στην συνέχεια αυτής της διαδικασίας, μέσω ενός αριθμού γενεών θα οδηγήθούμε σε βέλτιστες ή σε λύσεις κοντά στην βέλτιστη. Η κύρια διαφορά μεταξύ των γενετικών αλγορίθμων και άλλων μετα-ευρετικών προσεγγίσεων, όπως η προσομοιωμένη ανόπτηση και η αναζήτηση Tabu, είναι ότι ασχολούνται με πληθυσμούς των λύσεων και όχι με μια ενιαία λύση και ως εκ τούτου εξερευνούν τη γειτονιά του συνόλου του πληθυσμού. Φορείς όπως η επιλογή, η διασταύρωση και η μετάλλαξη χρησιμοποιούνται για να εξερευνήσουν την περιοχή και να δημιουργήσουν μια νέα γενιά.

2. **Αφελής Γενετικός Αλγόριθμος** (Naïve Evolution Algorithm) . Η βασική ιδέα είναι η ίδια όπως και στους γενετικούς αλγόριθμους. Ωστόσο, κανένας φορέας δεν

εφαρμόζεται για να ελέγξει τον χώρο αναζήτησης. Μόνον ο φορέας εκμετάλλευσης (μετάλλαξη) χρησιμοποιείται για την παραγωγή του επόμενου πληθυσμού.

3. **Προσομοιωμένης Ανόπτησης** (Simulated Annealing). O Eglese το 1990 ερεύνησε την εφαρμογή της Προσομοιωμένης Ανόπτησης ως εργαλείο για την Επιχειρησιακή Έρευνα. Αυτή εισήχθη ως εργαλείο βελτιστοποίησης τη δεκαετία του 1980, όταν η έννοια της φυσικής πρώτης αναδιάταξης εφαρμόστηκε σε συνδυαστική βελτιστοποίηση. Η μεταφορά αυτού του μοντέλου σε συνδυαστικά προβλήματα, όπου τα ενεργειακά επίπεδα ενός συστήματος αντιστοιχούν στις διάφορες εφικτές λύσεις για ένα πρόβλημα και η ενέργεια του συστήματος για τη λειτουργία του κόστους μιας συνάρτησης, πρέπει να ελαχιστοποιηθούν. Η Προσομοιωμένη Ανόπτηση μπορεί να θεωρηθεί ως μια παραλλαγή της μεθόδου αναρρίχηση λόφων η οποία προσπαθεί να αποφύγει την παγίδευση σε τοπικά ελάχιστα. Αντί μόνο να αποδέχεται γειτονικές λύσεις που έχουν ως αποτέλεσμα μία βελτίωση, δέχεται και κάποιες λύσεις που είναι χειρότερες, οι οποίες μπορεί να γίνουν αποδεκτές τυχαία με μια ορισμένη πιθανότητα. Αυτή η πιθανότητα εξαρτάται από την αύξηση του κόστους και μίας παραμέτρου ελέγχου, δηλαδή της θερμοκρασίας ανόπτησης στη φυσική. Όσο μικρότερη είναι η αύξηση του κόστους και υψηλότερη η θερμοκρασία τόσο πιο πιθανές βέλτιστες λύσεις γίνονται δεκτές. Κατά τη διάρκεια της διαδικασίας ανόπτησης, η θερμοκρασία σταδιακά μειώνεται σύμφωνα με το χρονοδιάγραμμα ψύξης. Αυτό σημαίνει ότι ο αλγόριθμος γίνεται όλο και πιο επιλεκτικός στην αποδοχή νέων λύσεων. Στο τέλος της διαδικασίας μόνο κινήσεις, οι οποίες στην πράξη οδηγούν σε μια βελτίωση γίνονται αποδεκτές. Η διεργασία αναζήτησης τερματίζεται όταν φτάνει σε ένα κατώτερο όριο η θερμοκρασία ή το κόστος.

1.4 ΥΒΡΙΔΙΚΟΙ ΜΕΤΑΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Πολλές από τις επιτυχημένες εφαρμογές που έχουμε αναφέρει στις προηγούμενες παραγράφους είναι υβριδικές. Ο στόχος της έρευνας για την υβριδοποίηση των μεταευρυτικών αλγορίθμων έχει διπλό σκοπό.

1. Οι μεταευρετικοί αλγόριθμοι θα πρέπει να επανασχεδιαστούν ώστε να καθιστούν κατάλληλοι για παράλληλη εφαρμογή προκειμένου να αξιοποιούν τον εγγενή παραλληλισμό.

2. Ένας αποτελεσματικός συνδυασμός των μεταευρετικών πρέπει να βρεθεί, ώστε να συνδυάζουν διαφορετικά χαρακτηριστικά και δυνάμεις γιανα σχεδιάσουν μια αποτελεσματική επικοινωνία μηχανισμών.

Υπάρχουν τρεις μορφές υβριδοποίησης. Η πρώτη μορφή αποτελείται από στοιχεία ενός αλγορίθμου που συμπεριλαμβάνονται σε έναν άλλο μεταευρετικό αλγόριθμο. Η δεύτερη μορφή συστημάτων αναφέρεται μερικές φορές ως συνεταιριστική αναζήτηση και αποτελείται από διάφορους αλγορίθμους που ανταλλάσουν πληροφορίες με κάποιο τρόπο. Η τρίτη μορφή είναι η ενσωμάτωση της προσέγγισης και η συστηματική μέθοδος (ή πλήρης) όπου η ανταλλαγή στοιχείων γίνεται μεταξύ μεταευρετικών αλγορίθμων.

1.4.1 ΥΒΡΙΔΟΠΟΙΗΣΗ ΜΕΘΟΔΩΝ ΤΡΟΧΙΑΣ

Ένας από τους πιο γνωστούς τρόπους υβριδισμού είναι η χρήση της **μεθόδου τροχιάς** με βάση τον πληθυσμό. Οι περισσότερες από τις εφαρμογές των Εξελικτικών Αλγορίθμων και της Βελτιστοποίησης Αποικίας Μυρμηγκιών χρησιμοποιούν τις διαδικασίες τοπικής αναζήτησης. Η δύναμη της μεθόδου με βάση τον πληθυσμό βασίζεται στην έννοια του ανασυνδυασμού λύσεων για την απόκτηση νέων. Στους Εξελικτικούς Αλγορίθμους, συχνά διατηρούνται οι καλύτερες λύσεις που βρέθηκαν στην αρχή της αντίστοιχης εκκίνησης του αλγορίθμου. Αυτό ονομάζεται σταθερή εξέλιξη της γειτονιάς. Σε ορισμένες υλοποιήσεις Βελτιστοποιήσης Αποικίας Μυρμηγκιών μια φερομόνη (χημική ουσία που παράγουν τα μυρμήγκια) ενημερώνει το χρονοδιάγραμμα εφαρμογής κατά την κατάσταση όπου ο αλγόριθμος έχει σχεδόν συγκλίνει σε μια λύση. Μόνον η καλύτερη λύση που βρέθηκε από την έναρξη χρησιμοποιείται για την ενημέρωση των μονοπατιών της φερομόνης. Αυτό αντιστοιχεί σε «αλλαγή κατεύθυνσης» κατευθύνοντας τη διαδικασία αναζήτησης προς μια πολύ καλή λύση και ελπίζοντας να βρει την καλύτερη από αυτές στο μέλλον.

Η δύναμη των μεθόδων τροχιάς δεν έχει μονό στόχο να βρει κάποιο τρόπο και να εξερευνήσει μια πολλά υποσχόμενη περιοχή στο χώρο αναζήτησης, όπως στις μεθόδους τοπικής αναζήτησης, αλλά να οδηγήσει στην λύση. Σε μια πολλά υποσχόμενη περιοχή στον χώρο αναζήτησης, θα γίνει αναζήτηση με έναν πιο δομημένο τρόπο από ότι σε μεθόδους που βασίζονται στον πληθυσμό. Με αυτόν τον τρόπο υπάρχει ο κίνδυνος να είναι κοντά σε καλές λύσεις οι οποίες να μην έχουν υψηλή ακρίβεια εύρεσης. Εν ολίγοις, οι μέθοδοι με βάση τον πληθυσμό είναι καλύτερες για τον εντοπισμό του υποσχόμενου τομέα στον χώρο αναζήτησης, ενώ οι μέθοδοι τροχιάς είναι καλύτερες στην εξερεύνηση του υποσχόμενου τομέα στο χώρο αναζήτησης. Έτσι, οι υβριδικοί μεταευρετικοί αλγόριθμοι καταφέρνουν να συνδυάζουν με ορισμένο τρόπο τα πλεονεκτήματα των αλγορίθμων με βάση τον πληθυσμό με την αντοχή των μεθόδων τροχιάς, και είναι συχνά πολύ επιτυχείς.

1.4.2 ΥΒΡΙΔΟΠΟΙΗΣΗ ΣΥΝΕΤΑΙΡΙΣΤΙΚΩΝ ΑΛΓΟΡΙΘΜΩΝ

Μια όχι τόσο ποιοτική μορφή υβριδοποίησης παρέχεται από την συνεταιριστική αναζήτηση, η οποία αποτελείται από μια αναζήτηση που εκτελείται από πιθανώς διαφορετικούς αλγορίθμους. Έχει να κάνει με την ανταλλαγή πληροφοριών σχετικά με τις γειτονιές, τα μοντέλα, τα υπο-προβλήματα, τις λύσεις ή και άλλα χαρακτηριστικά του χώρου αναζήτησης. Συνήθως, οι **συνεταιριστικοί αλγόριθμοι** αναζήτησης συνίστανται στην παράλληλη εκτέλεση αλγορίθμων αναζήτησης με διαφορετικό επίπεδο επικοινωνίας. Οι αλγόριθμοι μπορεί να είναι διαφορετικοί ή μπορούν να είναι περιπτώσεις του ίδιου αλγόριθμου που εργάζονται σε διαφορετικά μοντέλα ή ρυθμίζουν την εκτέλεση με διαφορετικές παραμέτρους.

1.4.3 ΥΒΡΙΔΟΠΟΙΗΣΗ ΜΕ ΣΥΣΤΗΜΑΤΙΚΕΣ ΤΕΧΝΙΚΕΣ ΑΝΑΖΗΤΗΣΗΣ

Ας κάνουμε μια επισκόπηση της ενσωμάτωσης μεταευρετικών αλγορίθμων και συστηματικών τεχνικών αναζήτησης. Οι τελευταίες περιλαμβάνουν μια σειρά μεταβλητών που περιγράφουν τα υπο-προβλήματα, μειώνοντας έτσι την πολυπλοκότητα της μοντελοποίησης καλώς καθορισμένων τμημάτων του προβλήματος με ενιαίους περιορισμούς. Κάθε περιορισμός συνδέεται με ένα αλγόριθμο φιλτραρίσματος που διαγράφει τις τιμές από μια μεταβλητή περιοχή που δεν συμβάλλουν σε εφικτές λύσεις. Ένα **CP σύστημα** μπορεί να θεωρηθεί ως η αλληλεπίδραση των συστατικών (περιορισμοί) τα οποία επικοινωνούν μέσω κοινών μεταβλητών. Οι περιορισμοί ενεργοποιούνται μόλις ένας τομέας της οποιασδήποτε εμπλεκόμενης μεταβλητής έχει αλλάξει. Στη συνέχεια, εκτελούν μια φάση πολλαπλασιασμών, που ουσιαστικά αποτελεί τον αλγόριθμο φιλτραρίσματος. Αυτή η διαδικασία σταματά μόλις δεν υφίστανται πλέον εφικτές λύσεις που μπορούν να απομακρυνθούν από τους τομείς ή όταν τουλάχιστον ένας τομέας είναι άδειος (δεν υπάρχει εφικτή λύση). Δεδομένου ότι η πολυπλοκότητα της πλήρους διάδοσης περιορισμών είναι συχνά εκθετική, συνήθως δεν γίνεται πλήρες φιλτράρισμα. Συνεπώς κατά το τέλος της φάσης πολλαπλασιασμού, μερικές περιοχές μπορούν να εξακολουθούν να περιέχουν ανέφικτες τιμές.

Υπάρχουν τρεις κύριες προσεγγίσεις για την ένωση των μεταευρετικών αλγορίθμων (ιδιαίτερα σε μεθόδους τροχιάς) και των συστηματικών τεχνικών (CP):

1. Ένας μεταευρετικός αλγόριθμος και μια συστηματική μέθοδος εφαρμόζονται διαδοχικά ή παρεμβάλλονται. Για παράδειγμα, ο αλγόριθμος τρέχει για την παράγωγη κάποιων λύσεων και στην συνέχεια το σύνολο τους χρησιμοποιείται ως ευρετική πληροφορία από τη συστηματική αναζήτηση. Αντιστρόφως, η συστηματική αναζήτηση μπορεί να τρέξει για να δημιουργήσει μια μερική λύση η οποία στη συνέχεια μπορεί να ολοκληρώσει μέχρι και την εκτέλεση του μεταευρυτικού αλγορίθμου. Αυτή η προσέγγιση μπορεί να θεωρηθεί ως ένα παράδειγμα των συνεταιριστικών αναζητήσεων και αντιπροσωπεύει μια μάλλον απλή ενσωμάτωση.

2. Οι μεταευρετικοί αλγόριθμοι χρησιμοποιούν προγραμματισμό με περιορισμούς και (ή) ένα δέντρο αναζήτησης για να διερευνήσουν αποτελεσματικά την γειτονιά, αντί απλώς να απαριθμήσουν τις γείτονες ή να κάνουν μια τυχαία δειγματοληψία της γειτονιάς. Η προσέγγιση συνδυάζει τα πλεονεκτήματα από μια γρήγορη εξερεύνηση του χώρου αναζήτησης μέσω ενός μεταευρετικού αλγορίθμου, με την αποτελεσματική διερεύνηση που πραγματοποιείται σε γειτονιές από μια συστηματική μέθοδο. Αυτές οι προσεγγίσεις είναι αποτελεσματικές, κυρίως όταν η γειτονιά διερεύνησης είναι πολύ μεγάλη. Επιπλέον, πολλά προβλήματα του πραγματικού κόσμου έχουν επιπλέον περιορισμούς (πλάγιοι περιορισμοί), οι οποίοι θα μπορούσαν να καθιστούν ακατάλληλοι για μια συνήθη εξερεύνηση γειτονιάς που εκτελείται από μεταευρετικούς αλγορίθμους, δεδομένου ότι συνήθως μόνο ένα δείγμα της γειτονιάς ή μια απαρίθμηση των λύσεων λαμβάνεται υπόψη. Έτσι, οι τεχνικές φιλτραρίσματος μπορούν να υποστηρίξουν αποτελεσματικά την εξερεύνηση γειτονιάς.

3. Η τρίτη δυνατότητα συνίσταται στην εισαγωγή εννοιών ή στρατηγικών από μια κλάση αλγορίθμων σε μια άλλη. Για παράδειγμα, οι έννοιες της λίστα Tabu και τα κριτήρια που ορίζονται στην Tabu Αναζήτηση μπορούν να χρησιμοποιηθούν για την διαχείριση της λίστας ανοικτών κόμβων (δηλαδή, εκείνοι των οποίων οι κόμβοι παιδιά δεν έχουν ακόμα διερευνηθεί) σε έναν αλγόριθμο αναζήτησης δέντρου. Η

τρίτη προσέγγιση διατηρεί την αναζήτηση του διαστήματος με βάση μια συστηματική αναζήτηση. Ωστόσο αποτελεί πρόβλημα ο εξαντλητικός χαρακτήρας της αναζήτησης. Η υβριδοποίηση επιτυγχάνεται συνήθως με ενσωμάτωση των εννοιών που αναπτύχθηκαν για τους μεταευρετικούς αλγορίθμους (π.χ. πιθανολογικές επιλογές, φιλόδοξα κριτήρια, ευρετική κατασκευή) σε μεθόδους αναζήτησης σε δέντρο. Μια τυπική εφαρμογή αυτής της ολοκλήρωσης είναι η χρήση μίας πιθανοτικής οπισθοδρόμησης, αντί για ντετερμινιστική-χρονολογική οπισθοδρόμηση. Ο κατάλογος των πιθανών κινήσεων οπισθοδρόμησης μπορεί επίσης να ταξινομείται με τη βοήθεια ενός δυναμικού ευρετικού αλγορίθμου. Αυτή η δειγματοληψία μπορεί να πραγματοποιηθεί από ένα μεταευρετικό αλγόριθμο. Το αποτέλεσμα της κάθε πιθανής κίνησης οπισθοδρόμησης επιλέγεται ως το σημείο εκκίνησης για την παραγωγή μιας πλήρης λύσης σε ένα μεταευρετικό αλγόριθμο (περισσότερες από μια λύσεις μπορεί να παραχθούν από κάθε μερική λύση). Στη συνέχεια, μια βαθμολογία σχετικά με την ποιότητα των ολοκληρωμένων λύσεων χρησιμοποιείται για την επιλογή μιας κίνησης υπαναχώρησης. Ένα άλλο σημαντικό στοιχείο για να επιτευχθεί αυτός ο σκοπός είναι η εισαγωγή της τυχαιοποίησης σε συστηματικές τεχνικές. (C. Blum Universite &[5] A.Roli, 2003)

[5] σελ. 268 - 308

ΚΕΦΑΛΑΙΟ 2. ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ ΑΡΜΟΝΙΑΣ

Η πρώτη προσπάθεια κωδικοποίησης της μουσικής σε μαθηματικές σχέσεις ξεκίνησε από τον Πυθαγόρα. Σκοπός ήταν η εύρεση μιας χρυσής τομής, μέσω της μελέτης των αναλογιών, που θα αποδείκνυαν τον αρμονικό συνδυασμό των ήχων των χορδών, στην αρμονική φωτοσκίαση των κιόνων του Παρθενώνα και τελικά στην ένωση σώματος και ψυχής. Ο Πυθαγόρας θεωρούσε ότι μουσική και μαθηματικά εξασκούνται με σκοπό την υπέρβαση των σωματικών περιορισμών και της κάθαρσης της ψυχής, για να μπορέσει η ψυχή να απελευθερωθεί από τον κύκλο της μετενσάρκωσης και να φτάσει τελικά στη θεότητα από την οποία προέρχεται. Με στόχο να ερευνήσει τις σχέσεις μαθηματικών και μουσικής, κατασκεύασε ένα έγχορδο όργανο, το λεγόμενο «μονόχορδο». Με την χρήση του μονόχορδου ανακάλυψε μια σχέση απόλυτα σταθερή ανάμεσα στο μήκος της χορδής του μονόχορδου και των βασικών συγχορδιών (1/2 για το διάστημα ογδόης, 3/2 για της πέμπτης και 4/3 για της τετάρτης). Η αξιοσημείωτη ιδιότητα αυτών των αρμονικών σχέσεων βασίζεται στο ότι περιλάμβαναν τους τέσσερις πρώτους φυσικούς αριθμούς (1, 2, 3, 4), το άθροισμα των οποίων ισούται με το 10, τον ιερό αριθμό των Δελφών (Τετρακτύς). Η ανακάλυψη ότι οι ακριβείς μαθηματικές σχέσεις απεικόνιζαν αρμονικούς ήχους στο μονόχορδο ήταν πραγματικά αξιοπρόσεκτο. Ο χωρισμός ακριβώς στη μέση της χορδής, παρήγαγε ένα ευχάριστο ψυχικό συναίσθημα που απορρέει από έναν αρμονικό ήχο. Αυτή η μικρή εισαγωγή μας κάνει να συμπεραίνουμε ότι η πρώτη μουσική κατάκτηση για τον άνθρωπο είναι ο ρυθμός. Το μουσικό μέτρο είναι απαραίτητο για την παραγωγή μουσικής και παριστάνεται μέσω ενός κλασματικού αριθμού, ο οποίος δίνει τον ρυθμό. Η χρονική αξία της πρώτης και δεύτερης νότας είναι 1/4 και 1/2 αντίστοιχα, ενώ οι υπόλοιπες νότες είναι ενωμένες και έχουν αξία 1/8. Το κλάσμα 4/4 καθορίζει πως κάθε μέτρο, ή διάστημα, το οποίο περιέχει ένα σύνολο νοτών, πρέπει να περιέχει νότες συνολικής αξίας 4/4. Αυτό αποδεικνύεται αν αθροίσουμε τα κλάσματα: 1/4+1/2+1/8+1/8=4/4. Άρα ο αριθμός καθορίζει το ρυθμό και επιτρέπει την εκτέλεση ενός συγχρονισμένου μουσικού κομματιού.

Σήμερα οι επιστήμονες ανακαλύπτουν στα μαθηματικά την παραγωγή μουσικής. Μια πολύ μεγάλη σύνδεση εμφανίστηκε ανάμεσά τους, παραδείγματα της οποίας είναι το "φαινόμενο Mozart" (η μουσική του Mozart βελτιώνει τις μαθηματικές ικανότητες των παιδιών). Το άκουσμα μιας μελωδίας διεγείρει τους νευρώνες του εγκεφάλου, οι οποίοι χρησιμοποιούνται για τη χωρο-χρονική αντίληψη. Η αντίληψη αυτή απαιτείται σε πολλές υψηλού επιπέδου εγκεφαλικές λειτουργίες όπως στη μουσική σύνθεση, το σκάκι, τη μηχανική ακόμη και στα μαθηματικά.

Η λέξη αρμονία προέρχεται από το αρχαίο ρήμα «αρμόζω» ή «άρω» (βάση και της λέξεως «Αρετή») σημαίνει αρμογή, μουσική συμφωνία, συναρμογή, σύνδεσμος συνεπώς και εναρμόνιση των πραγμάτων. Ως μουσικός όρος, ειδικά για την αρχαιοελληνική μουσική, αρμονία σημαίνει μέθοδος κατασκευής μουσικών κλιμάκων, μέσω της συναρμογής μουσικών διαστημάτων, άρα κατ' επέκταση ο όρος αρμονία σημαίνει «μουσική κλίμακα». *Η Αρμονία στη μουσική αναφέρεται στο ηχητικό αποτέλεσμα που προκαλούν δύο ή περισσότερα μουσικά όργανα ταυτόχρονα, ή καλύτερα είναι το καλλιτεχνικό αποτέλεσμα δύο ή περισσότερων οργάνων που συνηχούν.* Στην ακουστική, η αρμονία σχετίζεται με τα διαφορετικά ηχητικά κύματα, που παράγονται από τον άνθρωπο και τα μουσικά όργανα, και το πώς αλληλεπιδρούν μεταξύ τους. Και τελικά θα δείξει εάν το τελικό αποτέλεσμα θα είναι ευχάριστο ή όχι. Αν και στη μουσική το όμορφο, το αποδεκτό, είναι κάτι το υποκειμενικό, διότι το ανθρώπινο αυτί αντιλαμβάνεται κάποιες συγκεκριμένες συνηχήσεις ως ευχάριστες-αρμονικές.

Η έννοια της αρμονίας και της σύνθεσης μουσικής δεν έμεινε ανεκμετάλλευτη στις μέρες μας. Πολλοί επιστήμονες θεώρησαν ότι κάτι μεγαλύτερο από την παραγωγή μουσικών αρμονιών, κρύβεται πίσω από την μουσική παραγωγή. Έτσι για την μελέτη της αρμονίας ξεκίνησαν έρευνες γύρω από διάφορες επιστήμες στις αρχές του 2000, ώστε να βρεθεί μια σχέση τους με την μουσική. Πρώτος ο Zong Woo Geem εμπνεύστηκε τη δημιουργία ενός νέου αλγορίθμου βελτιστοποίησης που θα στηριζόταν στη μουσική. Το 2000 παρουσίασε τον **Αλγόριθμο Αναζήτησης Αρμονίας** (Harmony Search Algorithm – HSA) σε διεθνές συνέδριο. (Geem et al., 2009[6])

Ο αλγόριθμος της αρμονικής αναζήτησης ανήκει στην κατηγορία των μεταευρετικών αλγορίθμων και αναζητά την βέλτιστη λύση σε προβλήματα βελτιστοποίησης, τα οποία κυρίως ανήκουν στην κατηγορία NP-Hard. Οι αλγόριθμοι αυτοί είναι εμπνευσμένοι από παραδείγματα της ζωή. Η λειτουργία του αλγορίθμου στηρίζεται στο τρόπο που ένας μουσικός παίζει ένα μουσικό όργανο. Αυτός θεωρείται ως μια μεταβλητή απόφασης, που

[6] σελ . 113

παράγει νότες, δηλαδή υποψήφιες τιμές απόφασης, με στόχο να βρει ένα συνδυασμό που να είναι ελκυστικός στο άκουσμα, δηλαδή με βάση τη μουσική να εμπίπτει στους κανόνες της αρμονίας.

Αλγοριθμικά αυτό που κάνουμε είναι να αποθηκεύουμε σε ένα πίνακα, σύνολα διανυσμάτων που θα είναι ουσιαστικά μεταβλητές απόφασης, δηλαδή νότες. Αυτά τα διανύσματα θα ελέγχονται για το αν θα αποτελούν τις βέλτιστες λύσεις, βάση κάποιου τυπικού σφάλματος. Σε κάθε επανάληψη παράγεται ένα τυχαίο διάνυσμα, κατά τον τρόπο που ένα σύνολο μουσικών κάνει έναν αυτοσχεδιασμό. Αν το νέο διάνυσμα είναι αποδεκτό (καλύτερο από τα προηγούμενα) αποθηκεύεται στη λίστα, δηλαδή τον πίνακα. Υπάρχουν δυο πιθανότητες:

α) η πιθανότητα να χρειαστεί η «μπάντα» να πειραματιστεί με νέες τιμές, και

β) η πιθανότητα οι τιμές του νέου διανύσματος να αποκλείουν κατά ένα βαθμό από το υπό εξέταση διάνυσμα.

Είναι χαρακτηριστικό ότι λόγω της φύσης του αλγορίθμου, δεν εγκλωβίζεται σε τοπικά ελάχιστα. Ας εισχωρήσουμε στην ανάλυση του.

2.1 ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ ΑΡΜΟΝΙΑΣ

Ο Αλγόριθμος Αναζήτησης Αρμονίας εμπνεύστηκε από τον τρόπο που παίζει ένας μουσικός μέσα σε ένα μουσικό σχήμα. Κατά την εκτέλεση ενός μουσικού κομματιού, ο μουσικός έχει τις εξής επιλογές:

1. Να παίξει μια ήδη γνωστή-βασική μελωδία ενός κομματιού. Η συγκεκριμένη μελωδία ονομάζεται «θέμα» και χαρακτηρίζει το κάθε κομμάτι. Προφανώς είναι γνωστή (βρίσκεται δηλαδή στη μνήμη του μουσικού) σε κάθε εκτελεστή.

2. Να παίξει μια παρόμοια με τη βασική μελωδία. Να κάνει παραλλαγές επάνω στο «θέμα» του κομματιού, εμπλουτίζοντας το με νότες που δεν παίχτηκαν μέχρι εκείνη τη στιγμή.

3. Να δημιουργήσει έναν αυτοσχεδιασμό. Να επιλέξει, λοιπόν, νέες ακολουθίες από νότες που θα αποτελέσουν το καινούριο μουσικό υλικό.

Ομοίως, όταν κάθε μεταβλητή απόφασης επιλέγει μία τιμή στον αλγόριθμο HS, ακολουθεί έναν από τις τρεις κανόνες:

(1) την επιλογή καθεμιάς τιμής από τη μνήμη (που ορίζονται ως εκτιμήσεις μνήμης),

(2) την επιλογή μιας παραλλαγμένης τιμής από μία τιμή της μνήμη (που ορίζονται από προσαρμογές βήματος), και

(3) την επιλογή εντελώς τυχαίας τιμής από μια πιθανή τιμή του καθορισμένου εύρους (που ορίζεται ως τυχαιοποίηση).

Κάθε δυνατή λύση αποτελείται από τις τιμές των μεταβλητών της αντικειμενικής συνάρτησης που πρέπει να βελτιστοποιηθεί (μεγιστοποίηση ή ελαχιστοποίηση). Κάθε τέτοιο σύνολο τιμών των μεταβλητών στον Αλγόριθμο ονομάζεται **«Αρμονία»** (Harmony). Κατά την πορεία εύρεσης της βέλτιστης τιμής, ένα *πλήθος «Αρμονιών»* (δηλαδή δυνατών λύσεων) διατηρούνται σε μια θέση στη μνήμη του υπολογιστή. Ο χώρος αυτός ορίζεται ως **«Αρμονική Μνήμη»** (Harmony Memory) και ο αριθμός των «Αρμονιών» που περιέχει σε κάθε στιγμή ορίζεται ως **«Μέγεθος Αρμονικής Μνήμης»** (Harmony Memory Size). Η Αναζήτηση του Αλγορίθμου ολοκληρώνεται όταν πραγματοποιηθεί το σύνολο του αριθμού των επαναλήψεων που έχουν προκαθοριστεί (Maximum Number of Iterations). Οι τρεις κανόνες στον αλγόριθμο λειτουργούν αποτελεσματικά χρησιμοποιώντας δύο παραμέτρους στην αρμονική μνήμη, θεωρώντας **το Δείκτη Αποδοχής Αρμονικής Μνήμης** (HMCR) και αργότερα τον **Δείκτη Προσαρμογής του Τονικού Ύψους** (PAR), όπως αναφέρεται. Και οι δύο παράμετροι είναι ποσοστά.

2.2 ΔΙΑΔΙΚΑΣΙΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Η διαδικασία βελτιστοποίησης του αλγορίθμου της αρμονικής αναζήτησης, αποτελείται από πέντε βήματα τα οποία είναι τα ακόλουθα:

1. Καθορισμός του προβλήματος βελτιστοποίησης και των παραμέτρων του αλγόριθμου.

2. Διαμόρφωση της μνήμης αρμονίας (HM).

3. Αυτοσχεδιασμός μιας νέας αρμονίας από την μνήμη αρμονίας ΗΜ.

4. Ενημέρωση της αρμονικής μνήμης ΗΜ.

5. Επανάληψη των βημάτων 3 και 4 μέχρι το κριτήριο τερματισμού να ικανοποιείται.

Στην συνέχεια ακολουθεί η ανάλυση του κάθε βήματος.

1) Το πρόβλημα βελτιστοποίησης ορίζεται ως εξής: Ελαχιστοποίηση της f(x), με x_i є Xi και i є 1,2,3,...,N όπου f(x) είναι η αντικειμενική συνάρτηση, x είναι το σύνολο της κάθε μεταβλητής σχεδιασμού (x_i), Xi είναι το σύνολο του πιθανού εύρους τιμών για κάθε μεταβλητή σχεδιασμού (συνεχείς μεταβλητές σχεδιασμού), δηλαδή, $_LXi \leq Xi \leq {}_uXi$, και N είναι ο αριθμός των μεταβλητών σχεδιασμού.

Οι παράμετροι του αλγορίθμου που απαιτούνται για την επίλυση του προβλήματος βελτιστοποίησης (δηλαδή τηςf(x)), ορίζονται σε αυτό το βήμα: το μέγεθος της μνήμης αρμονίας (αριθμός των διανυσμάτων λύσης στη μνήμη αρμονίας - HMS), η αρμονία μνήμης θεωρώντας το ποσοστό (HMCR), το ποσοστό τόνου (PAR) και το κριτήριο τερματισμού (μέγιστος αριθμός των αναζητήσεων). Εδώ, οι HMCR και PAR είναι παράμετροι που χρησιμοποιούνται για να βελτιώσουν το διάνυσμα λύσης. Και οι δύο ορίζονται στο Βήμα 3.

2) Αρχικοποίηση της μνήμης αρμονίας ΗΜ (πίνακας ΗΜ), που φαίνεται στην Εικόνα 1. Ο πίνακας θα είναι γεμάτος με τυχαία διανύσματα λύσεων και ταξινομημένος σύμφωνα με τις τιμές της αντικειμενικής συνάρτησης f(x).

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & \cdots & x_n^1 & | & f(\mathbf{x}^1) \\ \vdots & \ddots & \vdots & | & \vdots \\ x_1^{hms} & \cdots & x_n^{hms} & | & f(\mathbf{x}^{hms}) \end{bmatrix}$$

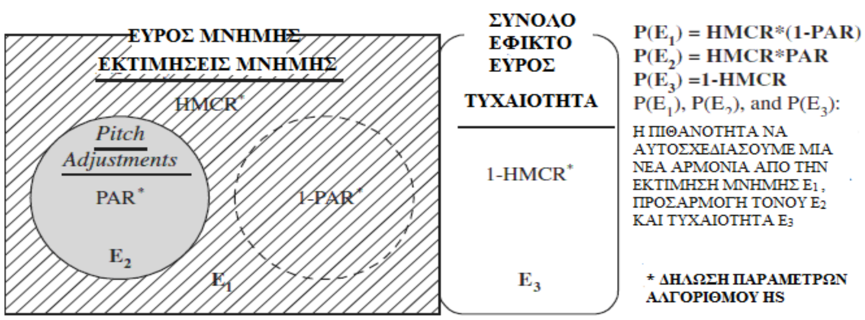
Εικόνα 1: Αρμονική Μνήμη ΗΜ

3) Αυτοσχεδιασμός μια νέας αρμονία από το ΗΜ. Στο βήμα αυτό, ένα νέο διάνυσμα αρμονίας, $x'=(x'_1, x'_2, x'_3, \ldots, x'_N)$ παράγεται από την μνήμη ΗΜ και βασίζεται σε εκτιμήσεις, προσαρμογές τόνου, και τυχαιοποίηση. Για παράδειγμα, η τιμή της πρώτης μεταβλητής σχεδιασμού (x'_1) για το νέο φορέα μπορεί να επιλεγεί από οποιαδήποτε τιμή στο καθορισμένο εύρος (x'_1 - x^{HMS}_1). Οι τιμές των άλλων μεταβλητών σχεδιασμού (x'_1) μπορεί να επιλέγονται με τον ίδιο τρόπο. Εδώ, είναι δυνατόν να επιλέγεται μια νέα τιμή χρησιμοποιώντας την παράμετρο HMCR, η οποία παίρνει τιμές στο διάστημα μεταξύ 0 και 1, ως εξής:

Αν Xi΄ μεταβλητή σχεδιασμού τότε:

- Xi΄ є {Xi1,Xi2, ... , XiHMS} με πιθανότητα = HMCR,
- Xi΄ є Xi με πιθανότητα = (1 – HMCR)

Το HMCR είναι η πιθανότητα επιλογής μίας τιμής από τις ιστορικές τιμές που είναι αποθηκευμένες στην μνήμη HM και (1 - HMCR) είναι η πιθανότητα τυχαίας επιλογής μίας εφικτής τιμής που δεν περιορίζεται σε αυτές που είναι αποθηκευμένες στη μνήμη HM. Για παράδειγμα, μια τιμή HMCR 0.95 δείχνει ότι ο αλγόριθμος HS θα επιλέξει την τιμή μεταβλητής σχεδιασμού από τις ιστορικά αποθηκευμένες τιμές στην HM με πιθανότητα 95% ενω από ολόκληρο το εφικτό εύρος με πιθανότητα 5%. Η τιμή 1.0 του HMCR δεν συνιστάται λόγω της πιθανότητας ότι το διάνυσμα λύσης μπορεί να βελτιωθεί με τις τιμές που δεν αποθηκεύονται στην μνήμη HM. Κάθε στοιχείο του νέου διανύσματος αρμονίας, $x'=(x'_1, x'_2, x'_3 ,…, x'_N)$, εξετάζεται για να διαπιστωθεί αν θα πρέπει να είναι προσαρμοσμένο στον τόνο. Αυτή η διαδικασία χρησιμοποιεί την παράμετρο PAR που προσαρμόζει το τονικό ύψος για το βήμα επιλέγοντας από την μνήμη HM, όπως φαίνεται στην Εικόνα 2, ως ακολούθως:



Εικόνα 2: Διαδικασία εκτίμησης Μνήμης Αρμονίας

Απόφαση Προσαρμογής τόνου για το xi΄:

- ΝΑΙ για πιθανότητα PAR

- ΟΧΙ για πιθανότητα 1 - PAR

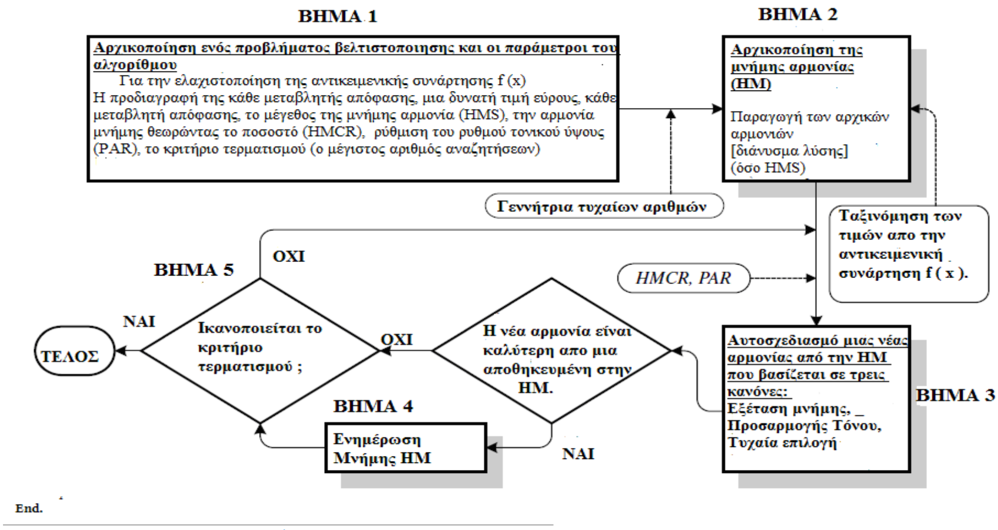
Η διαδικασία ρύθμισης τόνου εκτελείται μόνο μετά από την επιλογή μιας τιμής από την μνήμη HM. Η τιμή (1 - PAR), καθορίζει το ρυθμό διακοπής. Μια παράμετρος PAR ίση με 0,3 υποδεικνύει ότι ο αλγόριθμος θα επιλέξει μια γειτονική τιμή με πιθανότητα (30%*HMCR). Εάν η απόφαση ρύθμισης βήματος για την x_i μεταβλητή είναι ΝΑΙ και η x_i θεωρείται ότι είναι $x_i(k)$, δηλαδή το k-οστό στοιχείο στο Xi, η τιμή προσαρμογής τόνου της $x_i(k)$ είναι: $x_i \leftarrow x_i + a$ όπου a είναι η τιμή του (b_w*u(-1, 1)), όπου b_w είναι ένα αυθαίρετο εύρος απόστασης για τη συνεχή μεταβλητή σχεδιασμού και u(-1, 1) είναι μια ομοιόμορφη κατανομή μεταξύ -1 και 1. Οι παράμετροι HMCR και PAR εισήχθησαν στην αναζήτηση αρμονίας για να βοηθήσουν τον αλγόριθμο να βρει ολικά και τοπικά επίπεδά βέλτιστων λύσεων, αντίστοιχα.

4) Ενημέρωση της μνήμης HM. Στο βήμα αυτό, εάν το νέο διάνυσμα αρμονίας είναι καλύτερο από την χειρότερη αρμονία στην HM με βάση την τιμή της αντικειμενικής συνάρτησης, η νέα αρμονία περιλαμβάνεται στην HM και η υφιστάμενη χειρότερη αρμονία εξαιρείται από την μνήμη HM. Η HM στη συνέχεια ταξινομεί τις τιμές της από την αντικειμενική συνάρτηση.

5) Επανάληψη των βημάτων 3 και 4 μέχρι το κριτήριο τερματισμού να ικανοποιηθεί. (Lee & Geem, 2004[7])

Το διάγραμμα ροής του Αλγορίθμου Αναζήτησης Αρμονίας απεικονίζεται στην Εικόνα 3:

[7] σελ 781-784



Εικόνα 3: Διάγραμμα ροής Αλγορίθμου Αναζήτησης Αρμονίας

Ο ψευδοκώδικας του αλγορίθμου αναζήτησης αρμονίας παραθέτεται στην Εικόνα 4.

```
Harmony Search
begin
    Objective function f(x), x=(x1,x2, ...,xd)^T
    Generate initial harmonics (real number arrays)
    Define pitch adjusting rate (r_pa), pitch limits and bandwidth
    Define harmony memory accepting rate (r_accept)
    while ( t<Max number of iterations )
        Generate new harmonics by accepting best harmonics
        Adjust pitch to get new harmonics (solutions)
        if   (rand>r_accept), choose an existing harmonic randomly
        else if  (rand>r_pa), adjust the pitch randomly within limits
        else     generate new harmonics via randomization
        end if
        Accept the new harmonics (solutions) if better
    end while
    Find the current best solutions
end
```

Εικόνα 4: Ψευδοκώδικας του Αλγορίθμου Αναζήτησης Αρμονίας

2.3 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ ΑΡΜΟΝΙΑΣ

Ας αναλύσουμε τον αλγόριθμο αναζήτησης Αρμονίας στο πλαίσιο των μεγάλων συνιστωσών των μεταευρετικών αλγορίθμων και ας προσπαθήσουμε να τον συγκρίνουμε με άλλους αλγορίθμους. Έτσι μπορούμε να εντοπίσουμε τους τρόπους εντατικοποίησης και διαφοροποίησης του αλγορίθμου και ίσως καταλάβουμε γιατί είναι ένας πολύ επιτυχημένος μεταευρετικός αλγόριθμος.

Στον αλγόριθμο, η διαφοροποίηση ουσιαστικά ελέγχεται από τη ρύθμιση βήματος και την τυχαιοποίηση. Υπάρχουν δύο δευτερεύουσες συνιστώσες για τη διαφοροποίηση, η οποία είναι ένας σημαντικός παράγοντας για την υψηλή αποτελεσματικότητα της μεθόδου. Η πρώτη είναι η σύνθεση «νέας μουσικής» μέσω τυχαιοποίησης και η δημιουργία νέων λύσεων, που θα είναι τουλάχιστον ίδιου επίπεδου απόδοσης, όπως σε άλλους αλγορίθμους που λειτουργούν με τυχαιοποίηση. Ωστόσο, ένα επιπλέον στοιχείο για τη διαφοροποίηση είναι η προσαρμογή του βήματος. Η ρύθμισης βήματος πραγματοποιείται με προσαρμογή του βήματος στο δεδομένο εύρος ζώνης από μια μικρή τυχαία ποσότητα σε σχέση με το υφιστάμενο βήμα ή με μια λύση από τη μνήμη αρμονίας. Ουσιαστικά, το βήμα ρύθμισης είναι μια διαδικασία βελτίωσης των τοπικών λύσεων. Τόσο η μνήμη όσο και η ρύθμιση βήματος διασφαλίζουν την διατήρηση των καλών τοπικών λύσεων, ενώ η τυχαιοποίηση και η μνήμη αρμονίας θα διερευνήσουν τον ολικό χώρο αναζήτησης αποτελεσματικά. Το κύριο σημείο είναι ότι πρόκειται για μια ελεγχόμενη διαφοροποίηση γύρω από τις καλές λύσεις (καλές και αρμονικές θέσεις) και λειτουργεί σχεδόν σαν ένας παράγοντας εντατικοποίησης. Η τυχαιοποίηση εξερευνά το χώρο αναζήτησης πιο αποδοτικά και αποτελεσματικά, ενώ η προσαρμογή βήματος εξασφαλίζει ότι οι λύσεις που δημιουργήθηκαν πρόσφατα είναι αρκετά καλές ή δεν είναι πολύ μακριά από την εύρεση των υπαρχουσών καλών λύσεων.

Η εντατικοποίηση αντιπροσωπεύεται κυρίως στον αλγόριθμο από το ποσοστό αποδοχής μνήμης αρμονίας. Ένα υψηλό ποσοστό αποδοχής της αρμονίας λέει ότι οι καλές αποθηκευμένες λύσεις στην μνήμη είναι πιο πιθανό να επιλεγούν ή να κληρονομηθούν. Αυτό ισοδυναμεί με έναν ορισμένο βαθμό ελιτισμού. Προφανώς, εάν το ποσοστό αποδοχής είναι πολύ χαμηλό, οι λύσεις θα συγκλίνουν πιο αργά. Όπως αναφέρθηκε προηγουμένως, αυτή η εντατικοποίηση ενισχύεται από την ελεγχόμενη προσαρμογή βήματος. Τέτοιες αλληλεπιδράσεις μεταξύ διαφορετικών συστατικών θα μπορούσαν να είναι ένας άλλος σημαντικός παράγοντας για την επιτυχία του αλγόριθμου σε σύγκριση με άλλους.

Επιπλέον, η εφαρμογή του αλγορίθμου αρμονίας είναι επίσης ευκολότερη. Υπάρχουν κάποια στοιχεία που δείχνουν ότι ο αλγόριθμος αρμονίας είναι λιγότερο ευαίσθητος σε επιλεγμένες παραμέτρους, κάτι το οποίο σημαίνει ότι δεν χρειάζεται να έχουμε τελειοποιήσει αυτές τις παραμέτρους για να πάρουμε ποιοτικές λύσεις. Επίσης, ο αλγόριθμος αρμονίας είναι ένας μεταευρετικός αλγόριθμος βασισμένος στον πληθυσμό, που σημαίνει ότι οι πολλαπλές ομάδες αρμονίας μπορούν να χρησιμοποιηθούν παράλληλα. Ο σωστός παραλληλισμός συνήθως οδηγεί σε καλύτερη εύρεση λύσης με υψηλότερη απόδοση.

Ο καλός συνδυασμός του παραλληλισμού με τον ελιτισμό, καθώς και μια λεπτή ισορροπία της εντατικοποίησης και της διαφοροποίησης είναι το κλειδί για την επιτυχία του αλγορίθμου αρμονίας όπως και για την επιτυχία οποιουδήποτε μεταευρετικού αλγορίθμου. Τα πλεονεκτήματα αυτά τον καθιστούν πολύ ευέλικτο στο να συνδυάζεται με άλλους μεταευρετικούς αλγόριθμους, όπως τον PSO, για την παραγωγή υβριδικών μεταευρετικών και να εφαρμόζεται σε διάφορες εφαρμογές.(Yang 2009a[8]))

2.4 ΠΑΡΑΛΛΑΓΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ο Αλγόριθμος Αναζήτησης Αρμονίας τράβηξε την προσοχή πολλών ερευνητών για την επίλυση προβλημάτων βελτιστοποίηση όπως προβλήματα μηχανικής και πληροφορικής. Κατά συνέπεια, το ενδιαφέρον γι' αυτόν τον αλγόριθμο οδήγησε τους ερευνητές να βελτιώσουν και να αναπτύξουν τις επιδόσεις του, σύμφωνα με τις απαιτήσεις των προβλημάτων προς λύση. Οι βελτιώσεις αυτές καλύπτουν κυρίως δύο πτυχές:

(1) την βελτίωση του αλγορίθμου, στην διάρκεια της ρύθμισης των παραμέτρων, και

(2) τις βελτιώσεις της υβριδοποίησης του αλγορίθμου ως προς τα συστατικά σε σχέση με άλλους μεταευρετικούς αλγορίθμους.

Οι ερευνητές προσπάθησαν να δημιουργήσουν παραλλαγές του με διαφορετικές ρυθμίσεις ώστε να βελτιώσουν την απόδοση του και την ταχύτητα σύγκλισης στην βέλτιστη λύση του προβλήματος που μελετούσαν. Ανάμεσα στους ερευνητές ήταν και ο δημιουργός του αλγορίθμου Z. W. Geem, ο οποίος πρότεινε παραλλαγές στον τρόπο λειτουργίας και στην δομή του αλγορίθμου. Στην συνέχεια θα δούμε τις παραλλαγές του αλγορίθμου αναζήτησης αρμονίας.

2.4.1 ΜΕ ΣΥΜΠΡΑΞΗ ΜΟΥΣΙΚΟΥ ΣΥΝΟΛΟΥ

Η πρώτη παραλλαγή του Αλγορίθμου εμφανίστηκε το 2006 και ονομάστηκε Βελτιωμένος Αλγόριθμος Αναζήτησης Αρμονίας με την σύμπραξη μουσικού συνόλου. Η παραλλαγή αυτή

---

[8] σελ. 14 -17

προσθέτει έναν επιπλέον μηχανισμό, που ονομάζεται «Επίδραση του Μουσικού Συνόλου» και η λειτουργία ξεκινά κατόπιν της δημιουργίας της πρώτης λύσης, δηλαδή της νέας «Αρμονίας». Παρεμβάλλεται δηλαδή ανάμεσα στα στάδια 3 και 4 που παρουσιάστηκαν σε προηγούμενη παράγραφο, σαν 4η διαδικασία στο στάδιο 3. (Geem et. al., 2006[9])

Η παραλλαγή εμπνεύστηκε από το γεγονός ότι αρκετά συχνά σε ένα μουσικό σύνολο παρουσιάζεται στενή σύνδεση μεταξύ κάποιων μουσικών συνόλων. Οι παρόμοιες μελωδίες που παίζονται ανάμεσα στα όργανα και ο μουσικός διάλογος είναι συχνό φαινόμενο αρκετών ειδών μουσικής, όπως για παράδειγμα οι δύο ομάδων βιολιών μιας συμφωνικής ορχήστρας. Τα πρώτα βιολιά εναλλάσσουν ή συνοδεύουν τις μελωδίες τους με την ομάδα των δευτέρων βιολιών, η μεταξύ τους σχέση είναι εντονότερη από τη σχέση τους με τα χάλκινα, πνευστά ή κρουστά.

Στον αλγόριθμο τώρα το ίδιο συμβαίνει με κάποιες μεταβλητές απόφασης της αντικειμενικής συνάρτησης, που εμφανίζουν μεγαλύτερη αλληλεξάρτηση από ότι οι υπόλοιπες σε αρκετά προβλήματα. Μόνο οι αποφάσεις της αντικειμενικής συνάρτησης παρουσιάζουν μια μεγαλύτερη αλληλεξάρτηση από ότι οι υπόλοιπες. Η τιμή της μεταβλητής απόφασης της νέας Αρμονίας x_{new}^{j} που παράγει ο Αλγόριθμος μπορεί να στηρίζεται στην τιμή μιας άλλης μεταβλητής x_{new}^{i} όταν πιστοποιηθεί ότι οι μεταβλητές x_i και x_j που βρίσκονται στην Αρμονική Μνήμη παρουσιάζουν στενή σχέση. Η διαδικασία αυτή υλοποιείται με την εξίσωση

$$x_i^{new} \leftarrow f_n(x_j^{new})$$
$$i \neq j,$$

και η νέα τιμή θα καθορίζεται από τη μέγιστη συσχέτιση μεταξύ των μεταβλητών x_i και x_j

$$\max\,[\mathrm{Corr}(x_i, x_j)]^2$$

Η συνάρτηση Corr() είναι η συνάρτηση στατιστικής συσχέτισης μεταξύ των μεταβλητών $x_i=(x^1_i,x^2_i,\ldots,x^{HMS}_i)$ και $x_j=(x^1_j,x^2_j,\ldots,x^{HMS}_j)$ και παράγει το συντελεστή γραμμικής συσχέτισης

---

[9] σελ. 86 -93

r. Η max() είναι η συνάρτηση που βρίσκει την ποιο ισχυρά συσχετισμένη μεταβλητή, μεταξύ x_j και x_i, βασιζόμενη στον όρο r^2. Η $f_n()$ είναι η συνάρτηση που επιλέγει τον όρο x_{new}^i βασισμένη στον όρο x_{new}^j, και βρίσκει την τιμή x_i που εμφανίζεται πιο συχνά ανάμεσα στα ζευγάρια x_i και x_j, όταν το x_j είναι ίσο με το x_{new}^j.

Σε ένα ποσοστό των νέων αρμονιών που δημιουργήθηκαν, εισάγεται μια νέα παράμετρος, που ονομάζεται Δείκτης Επίδρασης του Μουσικού Συνόλου ECR (Ensemble Consideration Rate) . Η παράμετρος αυτή παίρνει τιμές από 0 έως και 1, δηλαδή ($0 \leq ECR \leq 1$) και ορίζει πιο θα είναι το ποσοστό των κομματιών μιας Αρμονίας που θα βασίζονται στην Επίδραση Μουσικού Συνόλου. Μετά τη δημιουργία της νέας Αρμονίας με την χρήση του νέου μηχανισμού,γίνεται έλεγχος για το αν η νέα Αρμονία ικανοποιεί τους περιορισμούς. Τέλος όμοια με τον Κλασικό Αλγόριθμο Αρμονίας, η νέα Αρμονία εισάγεται στην Αρμονική Μνήμη ή απορρίπτεται από αυτήν.

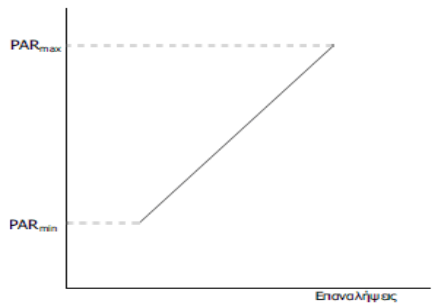
2.4.2 ΒΕΛΤΙΩΜΕΝΟΣ ΑΛΓΟΡΙΘΜΟΣ ΑΡΜΟΝΙΑΣ

Ο Mahdavi 2007, πρότεινε μια Βελτιωμένη εκδοχή του Αλγορίθμου Αρμονίας – Improved Harmony Search. Θεώρησε ότι η παράμετρος HMCR, και γενικά ο μηχανισμός της Αρμονικής Μνήμης, στοχεύει στην ευρεία αναζήτηση βέλτιστων λύσεων στο πεδίο τιμών (global search), καθώς και ότι ο μηχανισμός προσαρμογής του Τονικού Ύψους και ο συντελεστής PAR στοχεύουν στην αναζήτηση λύσεων σε τοπικό επίπεδο (local search). Η παράμετρος PAR καθώς και το εύρος b_w της ρύθμισης του τονικού ύψους είναι παράμετροι που παίζουν κυρίαρχο ρόλο τόσο στην λειτουργία όσο και στην απόδοση του Αλγορίθμου. Ο Mahdavi υποστήριζε ως μειονέκτημα του Κλασικού Αλγόριθμου Αρμονίας ότι αυτές οι δυο παράμετροι αρχικοποιούνται από σταθερές τιμές. Υποστήριξε επίσης ότι μικρές τιμές της παραμέτρου PAR μαζί με μεγάλες τιμές εύρους b_w μπορούν να κάνουν τον Αλγόριθμο να έχει χαμηλή απόδοση και σημαντική αύξηση στις απαιτούμενες επαναλήψεις για τη σύγκλιση στο ολικό ακρότατο. Παράλληλα, παρατήρησε ότι αν και για μικρές τιμές του εύρους b_w στις τελευταίες επαναλήψεις του Αλγορίθμου βελτιώνουν τις επιδόσεις του, στο αρχικό του στάδιο ωστόσο αποτελούν ένα μεγάλο εμπόδιο. Έτσι, συμπέρανε ότι για τις αρχικές επαναλήψεις του Αλγορίθμου η παράμετρος b_w θα προτιμάτε να παίρνει μεγαλύτερες τιμές, ώστε ο Αλγόριθμος να έχει την δυνατότητα ποικιλίας διαθέσιμων λύσεων στην Αρμονική Μνήμη. Τέλος παρατήρησε ότι οι μεγάλες τιμές της παραμέτρου PAR σε συνάρτηση με μικρό εύρος, ενισχύουν σημαντικά την επίδοση του Αλγορίθμου στις τελευταίες επαναλήψεις.

Για τα λάβει τα αποτέλεσμα των παρατηρήσεων αυτών έκανε τις μεταβλητές PAR και b_w να παίρνουν μεταβαλλόμενες τιμές στο πέρασμα του χρόνου. Με τον τρόπο αυτόν κατάφερε η παράμετρος PAR να αυξάνεται γραμμικά με το πέρας των επαναλήψεων και το εύρος b_w να μειώνεται εκθετικά. Αυτές οι μεταβλητές παράμετροι μεταβάλλονται δυναμικά ως εξής:

$$PAR_{gn} = PAR_{min} + \frac{[(PAR_{max} - PAR_{min}) \cdot gn]}{MaxIter}$$

όπου PAR είναι ο Δείκτης Προσαρμογής του Τονικού ύψους, PAR_{max} είναι η μέγιστη τιμή του Δείκτη Προσαρμογής του Τονικού ύψους, PAR_{min} είναι η ελάχιστη τιμή του Δείκτη Προσαρμογής του Τονικού ύψους, MaxIter είναι ο μέγιστος αριθμός επαναλήψεων του Αλγορίθμου και gn είναι ο τρέχων αριθμός επανάληψης.



Εικόνα 5: Η μεταβολή τιμών του Δείκτη Προσαρμογής Τονικού Ύψους

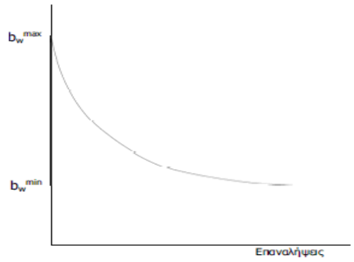
Η παράμετρος b_w μεταβάλλεται ως εξής:

$$b_w\,(gn) = b^{max}{}_w * \exp(c*gn)$$

όπου b_w είναι το εύρος προσαρμογής τονικού ύψους σε κάθε επανάληψη, $b_{max}w$, $b_{min}w$ είναι η ελάχιστη/μέγιστη τιμή εύρους

$$c = \frac{\ln(b_w^{max} / b_w^{min})}{MaxIter}$$

(Mahdavi et. al., 2007[10])



Εικόνα 6: Η μεταβολή τιμών του εύρους προσαρμογής τονικού ύψους

2.4.3 ΣΥΝΟΛΙΚΑ ΒΕΛΤΙΣΤΟΣ ΑΡΓΟΡΙΘΜΟΣ ΑΡΜΟΝΙΑΣ

Ο Omran και ο Mahdavi το 2008, έφτιαξαν μια παραλλαγή του Κλασικού Αλγόριθμου Αρμονίας, και την ονόμασαν Συνολικά Βέλτιστο Αλγόριθμο Αρμονίας (Global-best Harmony Search). Η συγκεκριμένη παραλλαγή αποτελείται από στοιχεία της μεθόδου Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization). Αυτός ο Αλγόριθμος χρησιμοποιεί την μίμηση και αναφέρεται στο ότι η θέση που θα λάβει ένα μέλος του σμήνους εξαρτάται τόσο από τις θέσεις που έχει ήδη λάβει, όσο και από τις θέσεις του καλύτερου μέλους του σμήνους. (Mahdavi & Omran, 2008[11])

Οι Omran και Mahdavi προτείνουν παρόμοια τροποποίηση του τρόπου λειτουργίας του μηχανισμού προσαρμογής τονικού ύψους. Αυτό έγινε με την εισαγωγή κυμαινόμενης τιμής στον συντελεστή PAR και αντικαθιστώντας το εύρος προσαρμογής b_w. Κάθε φορά που ο GB–HSA επιλέγει να πραγματοποιήσει προσαρμογή του τονικού ύψους, αντικαθιστά την τιμή που έχει επιλέξει από την Αρμονική Μνήμη (HM) με μία από τις τιμές των μεταβλητών της καλύτερης «Αρμονίας» που περιέχεται στην HM. Η καλύτερη μέχρι εκείνη τη στιγμή λύση συνεισφέρει την τιμή της εκάστοτε μεταβλητής σε κάθε προσαρμογή του τονικού ύψους. Οι αλλαγές αυτές οδήγησαν σε εξαιρετικές επιδόσεις στις δοκιμές του αλγορίθμου

[10] σελ. 1568 -1571

[11] σελ. 643 - 656

και χρησιμεύουν τόσο στην επίλυση διακριτών όσο και συνεχών προβλημάτων. Όμως η επιλογή τιμής άλλης μεταβλητής χωρίς μάλιστα κάποιο έλεγχο συσχέτισης φαίνεται το κάτι ανεπίτρεπτο. Θεωρήθηκε από πολλούς ερευνητές αδικαιολόγητα αυτονόητη η τόσο προφανή επιλογή τιμής της καλύτερης αρμονίας, χωρίς την συσχέτιση της με διαφορετικές μεταβλητές. Υποστηρίχθηκε η αδυναμία της συγκεκριμένης μεθόδου στην πρόωρη σύγκλιση σε τοπικό ακρότατο. Ως επακόλουθο και παρότι ο Συνολικά-βέλτιστος Αλγόριθμος Αρμονίας στις δοκιμές (σύνθετες μαθηματικές εξισώσεις) παρουσιάζει εξαιρετικές επιδόσεις, στα πραγματικά προβλήματα, δεν παρουσιάζει την ίδια αποδοτικότητα. Ο συνολικά βέλτιστος αλγόριθμος της αρμονίας παρουσιάζεται σε μορφή ψευδοκώδικα στην Εικόνα 7.

```
Συνολικά Βέλτιστος Αλγόριθμος Αρμονίας

Αντικειμενική συνάρτηση     f(x),  x = (x_1, ..., x_p)^T
Επιλογή των n το πλήθος μεταβλητών απόφασης και των περιορισμών
Καθορισμός των παραμέτρων HMSize, HMCR, PAR, MaxIter
while κριτήριο επαναλήψεων i < MaxIter, do
    i = i + 1
    for k = 1 → n do
        if random(0, 1) ≤ HMCR then
            Επέλεξε τιμή από την Αρμονική Μνήμη
            x_i^new = x_i^j,        όπου j ← U(1, ..., HMSize)
            if random(0, 1) ≤ PAR then
                Προσαρμογή του Τονικού Ύψους:
                x_i^new = x_i^best,   best είναι η καλύτερη «Αρμονίας» στην HM.
            end if
        else
            Αυτοσχεδιασμός - τυχαία επιλογή τιμής:     x_i^new = random
        end if
        Επικοινώνησε και αναβάθμισε το βέλτιστο πακέτο λύσεων
    end for
end while
Αποκωδικοποίησε και εξάγαγε τα καλύτερα αποτελέσματα
```

Εικόνα 7: Ο Ψευδοκώδικας του Βελτιωμένου Αλγορίθμου Αναζήτησης Αρμονίας

2.4.4 ΑΥΤΟ-ΠΡΟΣΑΡΜΟΣΤΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΑΡΜΟΝΙΑΣ

Το όνομα της συγκεκριμένης παραλλαγής είναι Self-adaptive harmony search algorithm και παρουσιάστηκε από τους Wang και[12] Huang το 2010. Αυτή η παραλλαγή παρεμβαίνει στη δομή του μηχανισμού προσαρμογής του τονικού ύψους του κλασικού Αλγόριθμου Αρμονίας. Οι δημιουργοί του, συμφώνησαν ότι η ρύθμιση των παραμέτρων της μεθόδου αποτελεί μια αδυναμία του, διότι σε κάποια προβλήματα δεν γνωρίζουμε ούτε καν την τάξη μεγέθους των λύσεων. Σαν συμπέρασμα η ρύθμιση των μηχανισμών της μεθόδου και η τελική εύρεση

---

[12] σελ. 2826–2837

βέλτιστων λύσεων δεν είναι πάντα εφικτή. Επιφυλάχθηκαν να δώσουν κάποια απάντηση σχετικά με τη συνεχή, γραμμική αύξηση του συντελεστή PAR που παρουσιάζει ο βελτιωμένος Αλγόριθμος Αρμονίας. Πρότειναν μάλιστα για την έρευνα του μηχανισμού προσαρμογής τονικού ύψους, ότι θα έπρεπε να υπάρχει εντατική χρήση στα πρώτα στάδια του Αλγορίθμου και βαθμιαία να μειώνεται. Εάν γίνει αυτό, η πρόωρη σύγκλιση του αλγορίθμου και η ταλάντωση γύρω από τοπικά ακρότατα, μπορεί να παραληφθεί. Τα λεγόμενα τους ενισχύει η παρατήρηση πως η χρήση σταθερού συντελεστή PAR στο Συνολικά Βέλτιστο Αλγόριθμο Αρμονίας (Global– best Harmony Search), τον έκανε πιο αποδοτικό σε σχέση με την περίπτωση γραμμικής αύξησής του.

Έτσι, λοιπόν, πρότειναν τη γραμμική μείωση του συντελεστή PAR κατά τη διάρκεια εκτέλεσης από 1 σε 0. Ταυτόχρονα συνέδεσαν το εύρος προσαρμογής με τις εκάστοτε μέγιστες και ελάχιστες τιμές της μεταβλητής που περιλαμβάνονται στην Αρμονική Μνήμη. Έτσι η νέα μεταβλητή xi που επιλέγεται για Προσαρμογή τονικού ύψους από την Αρμονική Μνήμη, θα παραλλάσσεται με δύο τρόπους:

$$x_i + [\max(HMi) - x_i] * \mathrm{random}[0,1)$$

$$x_i - [ximin(HMi)] * \mathrm{random}[0,1)$$

όπου x_i είναι η τιμή που επιλέχτηκε από την Αρμονική Μνήμη για προσαρμογή, max(HMi), min(HMi) είναι η μέγιστη και ελάχιστη τιμή της μεταβλητής στην HM και random[0,1) τυχαίος αριθμός μεταξύ 0 και 1 (δίχως ή και με το 1).

Συνολικά απέδειξαν ότι η συγκεκριμένη παραλλαγή παρουσιάζει καλύτερη απόδοση επιτυγχάνοντας μεγαλύτερη προσέγγιση στο εκάστοτε ολικό ακρότατο.

2.4.5 ΑΝΕΞΑΡΤΗΤΟΣ ΑΠΟ ΠΑΡΑΜΕΤΡΟΥΣ

Λόγου του προβλήματος της ρύθμισης παραμέτρων, ο Geem και[13] Sim το 2010 εμπνεύστηκε μιας παραλλαγή που δε θα απαιτεί ρύθμιση των παραμέτρων **(Parameter–setting–free harmony search algorithm)**. Στην παραλλαγή αυτή ο αλγόριθμος ξεκινά και εκτελείται για μικρό αριθμό επαναλήψεων, για παράδειγμα 3*HMSize πλήθος επαναλήψεων, κατά τον

[13]σελ. 3881-3889.

κλασικό τρόπο. Εκείνη την στιγμή θα δημιουργηθεί ένας νέος πίνακας με πλήθος στοιχείων ίσο με αυτά που περιέχονται στην αρμονική μνήμη. Ο πίνακας αυτός ονομάζεται Μνήμη Λειτουργικού Τύπου και περιέχει την προέλευση των στοιχείων της Αρμονικής Μνήμης. Με αυτόν τον τρόπο ο πίνακας Μνήμης Λειτουργικού Τύπου θα αποθηκεύσει τον τρόπο δημιουργίας ενός στοιχείου της αρμονικής μνήμης που θα έχει προέλθει από τον μηχανισμό αυτοσχεδιασμού. Επίσης γίνεται αυτόματη ρύθμιση των παραμέτρων ανάλογα με το πλήθος των στοιχείων που προήλθαν από διαφορετικούς μηχανισμούς.

Έτσι:

HMCR = Πλήθος στοιχείων που προήλθαν από (HMCR / PAR) ÷ HMSize

PAR= Πλήθος στοιχείων που προήλθαν από PAR ÷ HMSize

Συνεπώς όταν βρίσκουμε μια καλύτερη λύση σε ένα πρόβλημα ανανεώνεται εκτός από την Αρμονική Μνήμη και η Μνήμη Λειτουργικού Τύπου, καθώς την ίδια στιγμή μεταβάλλονται και οι τιμές των παραμέτρων HMCR και PAR.

Στην τελευταία μέθοδο, εκτός από τη συνεχόμενη αλλαγή των παραμέτρων HMCR και PAR κατά την εύρεση της βέλτιστης λύσης, κάθε μεταβλητή της αντικειμενικής συνάρτησης θα αποκτά διαφορετική τιμή για τις παραπάνω μεταβλητές. Αυτό συμβαίνει διότι οι παράμετροι δείχνουν να έχουν μεγάλη εξάρτιση από το πλήθος των στοιχείων της αρμονικής μνήμης που αντιστοιχούν στη συγκεκριμένη μεταβλητή και έχουν συγκεκριμένη προέλευση. Αποτέλεσμα είναι να έχουμε γρήγορη σύγκλιση του αλγορίθμου και άρα χαμηλή αποτελεσματικότητα. Έτσι, ο Geem για να αποφύγει την πρόωρη σύγκλιση του αλγορίθμου, εισήγαγε ακόμα μία ρύθμιση. Σκεπτόμενος ότι αν οι παράμετροι PAR και HMCR λάβουν την τιμή 0 ή 1 αντίστοιχα, δε θα αλλάξουν ποτέ ξανά τιμή κατά την διάρκεια της εκτέλεσης λύσης του προβλήματος. Γι' αυτό ο αλγόριθμος πρέπει να έχει την δυνατότητα να αποφεύγει την πρόωρη σύγκλιση όταν οι παράμετροι λάβουν τιμές κοντά στο 0 και 1.

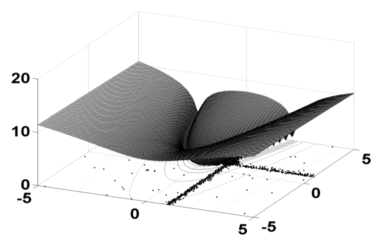
Αν ο HMCR = [HMCR+ Φ_{HMCR} * random (0,1)] δεν είναι μεταξύ [0,1], διατηρείται ο ίδιος HMCR. Και αν ο PAR=[PAR+ Φ_{PAR}* random(0,1)] δεν είναι μεταξύ [0,1], διατηρείται ο ίδιος PAR, όπου οι όροι Φ_{HMCR} και Φ_{PAR} είναι συντελεστές θορύβου με προτεινόμενες τιμές 0.05 και 0.1 αντίστοιχα.

Τα αποτελέσματα που δόθηκαν ήταν ενθαρρυντικά και η σύγκλιση στη βέλτιστη λύση επιτυχής και ταχεία. Σημαντικό πλεονέκτημα της συγκεκριμένης μεθόδου σύμφωνα με τον Geem είναι ότι δεν απαιτεί τον καθορισμό παραμέτρων του Αλγορίθμου Αρμονίας. Μόνο ο καθορισμούς του πλήθους των επαναλήψεων αρκεί για την εκτέλεση του Αλγορίθμου.

2.5 ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ

ΠΑΡΑΔΕΙΓΜΑ 1^{o}

Χρησιμοποιώντας την προγραμματιστική γλώσσα Matlab έχουμε την λογαριθμική συνάρτηση «μπανάνας του Rosenbrock» $f(x,y)=\ln[1+(1+x)^2+100(y-x^2)^2]$ όπου (x, y) ϵ [-10,10] × [-10,10], έχει ένα ολικό ελάχιστο f_{min}=0 στο (1,1). Η καλύτερη λύση εκτιμάται ότι είναι (1.0023,1.0070) και λαμβάνεται μετά από 15.000 επαναλήψεις. Σε έναν υπολογιστή με 3GHz επεξεργαστή, συνήθως η εύρεση της βέλτιστης τιμής διαρκεί περίπου 2 λεπτά. Στην Εικόνα 8 απεικονίζεται η γραφική παράσταση της συνάρτησης «μπανάνας του Resonbrock».



Εικόνα 8: Γραφική Παράσταση της συνάρτησης «μπανάνας του Rosenbrock»

Πηγή:X-S Yang,(2009) σελ.5

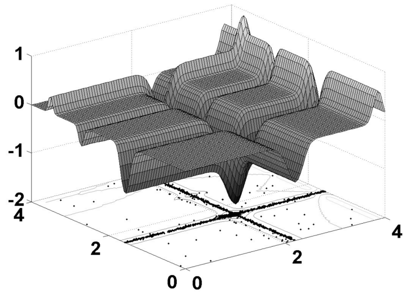
Έχουμε χρησιμοποιήσει 20 αρμονίες, το ποσοστό αποδοχής τόνου είναι r_{accept} = 0,95, και το ποσοστό ρύθμισης τόνου r_{pa} = 0,7. Οι διαδρομές αναζήτησης απεικονίζονται σε συνδυασμό με το τοπίο της f(x, y). Μπορούμε να δούμε ότι η προσαρμογή τόνου είναι πιο εντατική στις τοπικές περιφέρειες (δύο λεπτές λωρίδες), στο κάτω μέρος του σχήματος, αυτός είναι πιθανώς ένας άλλος λόγος για τον οποίο ο αλγόριθμος αναζήτησης αρμονίας είναι πιο αποτελεσματικός από τους γενετικούς αλγόριθμους.

ΠΑΡΑΔΕΙΓΜΑ 2^{o}

Ως ένα άλλο παράδειγμα, έχουμε την συνάρτηση δύο μεταβλητών Michalewicz,

$$f(x,y) =- \sin(x)\sin^{20}(x^2/\pi)-\sin(y)\sin^{20}(2y^2/\pi)$$

η οποία έχει ένα ολικό ελάχιστο fmin ≈-1,801 στο [2.20319,1.57049] στον τομέα 0≤.x≤.π και 0≤.y≤.π. Αυτό το ολικό ελάχιστο μπορεί να βρεθεί μετά από περίπου 23.000 επαναλήψεις τις αντικειμενικής συνάρτησης, και τα αποτελέσματα παρουσιάζονται στην Εικόνα 9.



Εικόνα 9: Γραφική Παράσταση της συνάρτησης δύο μεταβλητών Michalewicz

Πηγή:X-S Yang,(2009) σελ.6

ΚΕΦΑΛΑΙΟ 3. ΑΛΓΟΡΙΘΜΟΣ ΠΥΓΟΛΑΜΠΙΔΑΣ

Οι μεταευρετικοί αλγόριθμοι αποτελούν ένα σημαντικό μέρος της σύγχρονης αλγοριθμικής βελτιστοποίησης, της υπολογιστικής νοημοσύνης και του προγραμματισμού. Αυτοί οι αλγόριθμοι είναι συνήθως εμπνευσμένοι από τη φύση, ένα υποσύνολο των μεταευρετικών που συχνά αναφέρονται ως αλγόριθμοι σμήνους νοημοσύνης (SI) και έχουν αναπτυχθεί μιμούμενοι τα χαρακτηριστικά νοημοσύνη σμήνους των βιολογικών οργανισμών, όπως τα πτηνά, τα ψάρια, οι άνθρωποι και άλλα. Για παράδειγμα, η βελτιστοποίηση σμήνους σωματιδίων βασίστηκε στην εξάπλωση της συμπεριφοράς των πουλιών και των ψαριών, ο αλγόριθμος πυγολαμπίδας στο μοτίβο των τροπικών πυγολαμπίδων που εκπέμπουν φωτεινότητα, ενώ ο αλγόριθμος αναζήτησης κούκου είναι εμπνευσμένος από το πλήθος ορισμένων ειδών κούκων που παρασιτούν. Μεταξύ αυτών των αλγορίθμων, έχει αποδειχθεί ότι ο αλγόριθμος πυγολαμπίδας είναι πολύ αποτελεσματικός στην αντιμετώπιση προβλημάτων βελτιστοποίησης.

Ο αλγόριθμος Πυγολαμπίδας ανήκει στους αλγόριθμους στοχαστικών διαδικασιών, δηλαδή ψάχνει σε ένα σύνολο λύσεων χρησιμοποιώντας ένα είδος τυχαιότητας. Οι ευριστικοί αλγόριθμοι ακολουθούν την λογική «να βρω» ή «να ανακαλύψω τις λύσεις με δοκιμή και σφάλμα». Στην πραγματικότητα, δεν υπάρχει εγγύηση ότι η βέλτιστη λύση θα βρεθεί σε ένα εύλογο χρονικό διάστημα. Από την άλλη πλευρά, οι μεταευρετικές τεχνικές παρέχουν «υψηλότερου επίπεδου» λύσεις. Αυτό σημαίνει ότι κατά την διαδικασία αναζήτησης που χρησιμοποιεί επηρεάζονται ορισμένες αλληλεξαρτήσεις μεταξύ της τυχαίας και της τοπικής αναζήτηση.

Στον αλγόριθμο πυγολαμπίδας, το «χαμηλότερο επίπεδο» λύσεων (ευρετική τεχνική) επικεντρώνεται στην παραγωγή νέων λύσεων μέσα σε ένα χώρο αναζήτησης και επιλέγει την καλύτερη λύση για την επιβίωση. Από την άλλη, επιτρέπει την διαδικασία αναζήτησης με τυχαιοποίηση για να αποφευχθεί η παγίδευση των λύσεων σε τοπικά βέλτιστα. Η τοπική αναζήτηση βελτιώνει μια υποψήφια λύση έως ότου διαπιστωθούν οι βέλτιστες, δηλαδή τοποθετεί την συνάρτηση σε ένα τοπικό βέλτιστο. Κάθε μεταευρετική διαδικασία αναζήτησης εξαρτάται από την εξισορρόπηση μεταξύ δύο βασικών συνιστωσών, της εντατικοποίησης και της διαφοροποίηση;. Και οι δύο όροι επηρεάζονται από τις παραμέτρους ελέγχου του αλγορίθμου.

Ο Αλγόριθμος Πυγολαμπίδας είναι αλγόριθμος με βάσει τον πληθυσμού. Οι αλγόριθμοι με βάση τον πληθυσμό έχουν τα ακόλουθα πλεονεκτήματα σε σύγκριση με τους αλγόριθμους αναζήτηση μοναδικού σημείου:

1. Τα δομικά στοιχεία τοποθετούν μαζί τις διαφορετικές λύσεις μέσω διασταυρώσεων.
2. Εστιάζοντας και πάλι σε μια αναζήτηση που βασίζεται στην διασταύρωση, αν και οι δύο γονείς μοιράζονται την ίδια τιμή μιας μεταβλητής, τότε ο απόγονος θα έχει επίσης την ίδια τιμή αυτής της μεταβλητής.
3. Το χαμηλού επιπέδου φιλτράρισμα αγνοεί περισπασμούς μέσα στο τοπίο αναζήτησης.
4. Αντισταθμίζεται η κακή επιλογή αρχικών θέσεων ή οι αποφάσεις που λαμβάνονται.
5. Η ρύθμιση παραμέτρων είναι η ευκαιρία του αλγορίθμου να μάθει τις καλές τιμές προκειμένου να εξισορροπήσει την εξερεύνηση ενάντια στην εκμετάλλευση. (Fister et. al., 23013[14])

3.1 ΣΥΜΠΕΡΙΦΟΡΑ ΤΩΝ ΠΥΓΟΛΑΜΠΙΔΩΝ

Οι πυγολαμπίδες είναι αρθρόποδα έντομα, που ανήκουν στην τάξη των κολεοπτέρων και κατοικούν στις τροπικές και εύκρατες περιοχές. Είναι γνωστά ως λαμπυρίδες, πυγολαμπίδες ή κωλοφωτιές γιατί έχουν την ικανότητα να παράγουν φως. Ειδικά κύτταρα στο κάτω μέρος της κοιλιάς τους μετατρέπουν μέρος της χημικής ενέργειας που παίρνει το έντομο με το φαγητό του σε φως. Το φως αυτό ανάβει και σβήνει ρυθμικά. Ο ρυθμός φωτεινότητας και ο χρόνος αποτελούν μέρος του συστήματος σήματος που φέρνει και τα δύο φύλα μαζί. Υπάρχουν διαφορετικά είδη πυγολαμπίδων και κάθε ένα από αυτά παράγει διαφορετικό φωτεινό σήμα. Επιπλέον, η φωτεινότητα μπορεί επίσης να χρησιμεύσει ως ένας μηχανισμός προστασίας - προειδοποίησης.

Στα σμήνη των πυγολαμπίδων ισχύει ο κανόνας ότι η ελκυστικότητα είναι ανάλογη με τη φωτεινότητα, και οι δύο με την μείωση τους αυξάνουν την απόσταση. Έτσι, για

[14] σελ. 1- 3

οποιεσδήποτε δυο πυγολαμπίδες που αναβοσβήνουν, η λιγότερο φωτεινή θα κινηθεί προς την φωτεινότερη. Ο συνδυασμός αυτών των παραγόντων κάνει τις πυγολαμπίδες ορατές μόνο σε μια περιορισμένη απόσταση, συνήθως αυτή η απόσταση είναι αρκετά εκατοντάδες μέτρα τη νύχτα, ωστόσο είναι ο μοναδικός τρόπος επικοινωνίας τους.

Το σύστημα εναλλαγής φωτεινότητας \ μπορεί να μορφοποιηθεί με τέτοιο τρόπο ώστε να συνδέεται με μια αντικειμενική συνάρτηση που πρέπει να βελτιστοποιηθεί, η οποία καθιστά δυνατή την διαμόρφωση νέων αλγορίθμων βελτιστοποίησης. (Yang, 2010a[15])

3.1.1 ΚΛΑΣΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΠΥΓΟΛΑΜΠΙΔΑΣ

Ο Αλγόριθμος Πυγολαμπίδας (FA) είναι ένας μεταευρεστικός αλγόριθμος ο οποίος εμπνέεται από την συμπεριφορά των πυγολαμπίδων. Αναπτύχθηκε για πρώτη φορά από τον Xin - She Yang το 2008 στο Πανεπιστήμιο του Cambridge.

Στα σμήνη των πυγολαμπίδων ισχύει ότι η ελκυστικότητα είναι ανάλογη με τη φωτεινότητα, και τα δύο μεγέθη με την μείωση τους αυξάνουν την απόσταση. Έτσι, για δυο τυχαίες πυγολαμπίδες που αναβοσβήνουν, η λιγότερο φωτεινή θα κινηθεί προς την φωτεινότερη. Εάν δεν υπάρχει φωτεινότερη από ένα συγκεκριμένο πλήθος πυγολαμπίδων, θα κινούνται τυχαία. Η ένταση της φωτεινότητας (I) σε μια απόσταση (r) από την πηγή φωτεινότητας είναι αντιστρόφως ανάλογη του τετραγώνου της απόστασης ($1/ r^2$). Ο χώρος στον οποίο κινούνται οι πυγολαμπίδες απορροφά την εκπεμπόμενη φωτεινότητα τους με αποτέλεσμα ένα πλήθος πυγολαμπίδων να είναι αντιληπτές σε μια περιορισμένη απόσταση. Οι δυο παράγοντες: ένταση φωτεινότητας και απορρόφηση, έχουν μεγάλη σημασία στην αλγοριθμική συμπεριφορά των πυγολαμπίδων.

Ο αλγόριθμος της πυγολαμπίδας βασίζεται σε μια φυσική φόρμουλα της έντασης του φωτός το οποίο μειώνεται με την αύξηση του τετραγώνου της απόστασης r^2. Ωστόσο, καθώς η απόσταση από τη φωτεινή πηγή αυξάνει, η απορρόφηση κάνει το φως να γίνεται πιο αδύναμο. Αυτά τα φαινόμενα μπορεί να συνδέονται με την αντικειμενική συνάρτηση που πρέπει να βελτιστοποιηθεί .

---

[15] σελ. 210

Τρία είναι τα κύρια χαρακτηριστικά (Yang , 2010b[16]) του αλγορίθμου (FA).

1) Οι πυγολαμπίδες είναι μονογενείς οντότητες (δεν διαχωρίζονται σε αρσενικές και θηλυκές).

2) Μια πυγολαμπίδα κατευθύνεται προς τις πυγολαμπίδες οι οποίες εκπέμπουν μεγάλη φωτεινότητα. Η ελκυστικότητα μιας πυγολαμπίδας είναι ανάλογη της φωτεινότητας της. Έστω δύο πυγολαμπίδες, η λιγότερο φωτεινή έλκεται από την φωτεινότερη.

3) Η μείωση της φωτεινότητας μιας πυγολαμπίδας ελέγχεται από τον συντελεστή απορρόφησης γ. Αν υπάρχουν πυγολαμπίδες χωρίς να είναι ιδιαίτερα φωτεινές τότε αυτές κινούνται τυχαία.

3.1.2 ΕΝΤΑΣΗ ΦΩΤΕΙΝΟΤΗΤΑΣ ΚΑΙ ΕΛΚΥΣΤΙΚΟΤΗΤΑ

Δύο σημαντικά σημεία του αλγορίθμου (FA) είναι η φωτεινότητα και η ελκυστικότητα μεταξύ των πυγολαμπίδων. Σε ένα απλό πρόβλημα μεγιστοποίησης η ένταση της φωτεινότητας I μιας πυγολαμπίδας για μια συγκεκριμένη θέση x ορίζεται ως:

$$I = a * f(x)$$

όπου f(x) η αντικειμενική συνάρτηση του προβλήματος.

Η ελκυστικότητα β μιας πυγολαμπίδας σχετίζεται με την φωτεινότητα των υπολοίπων πυγολαμπίδων και μεταβάλλεται με την απόσταση r_{ij} των πυγολαμπίδων i και j. Η ένταση της φωτεινότητας μειώνεται με την απόσταση από την πηγή φωτεινότητας και η φωτεινότητα απορροφάται από το μέσο στο οποίο κινούνται οι πυγολαμπίδες, οπότε η ελκυστικότητα μεταβάλλεται ανάλογα με τον βαθμό απορρόφησης της φωτεινότητας.

Η ένταση της φωτεινότητας $I_{(r)}$ δίδεται ως εξής: $I_{(r)} = I_{(s)}/r^2$ **(1)** όπου $I_{(s)}$ είναι η φωτεινότητα της πηγής. Για ένα δεδομένο μέσο με σταθερό συντελεστή απορρόφησης γ, η ένταση της φωτεινότητας δίδεται $I = I_0 \times e^{-\gamma r}$ **(2)** όπου I_0 είναι η αρχική ένταση φωτεινότητας όταν r=0. Στην περίπτωση κατά την οποία r=0 (το πεδίο ορισμού της (1) δεν

[16] σελ. 81 -104

είναι ορισμένο) ο παράγοντας I_s/r^2 και ο συντελεστής απορρόφησης προσεγγιστικά δίδονται με την ακόλουθη Γκαουσιανή μορφή: $I = I_0 \times e^{-\gamma r^2}$ **(3).**

Η ελκυστικότητα των πυγολαμπίδων είναι ανάλογη της έντασης φωτεινότητας των πλησιέστερων πυγολαμπίδων, οπότε ορίζεται ως εξής: $\beta = \beta_0 \times e^{-\gamma r^2}$ **(4)** όπου: β_0 είναι η ελκυστικότητα στην απόσταση r=0. Για γρήγορους υπολογισμούς της ελκυστικότητας μπορεί να χρησιμοποιηθεί η ακόλουθη μαθηματική σχέση της ελκυστικότητας: $\beta = \beta_0/(1 + \gamma \times r^2)$ **(5).** Οι εξισώσεις (4) και (5) ορίζουν μια χαρακτηριστική απόσταση Γ: $\Gamma = 1/\gamma$ **(6)** όπου η ελκυστικότητα μεταβάλλεται σημαντικά από το β_0 στο $\beta_0 * e^{-1}$ για την (4) ή $\beta_0/2$ για την (5). Σε πραγματικό χρόνο, η συνάρτηση της ελκυστικότητας β(r) είναι μια συνάρτηση μονότονα φθίνουσα: $\beta(r) = \beta_0 \times e^{-\gamma r m}$, (m≥1) **(7)**

Για μία σταθερή χαρακτηριστική απόσταση Γ, ισχύει :

$$\Gamma = \lim_{x \to \infty} \gamma^{-\frac{1}{m}} = 1 \ \textbf{(8)}$$

Για ένα χαρακτηριστικό μήκος Γ, σε ένα πρόβλημα βελτιστοποίησης η παράμετρος γ μπορεί να ληφθεί ως μια αρχική τιμή: $\gamma = 1/\Gamma^m$ **(9)**

3.1.3 ΑΠΟΣΤΑΣΗ ΜΕΤΑΞΥ ΠΥΓΟΛΑΜΠΙΔΩΝ

Η συνάρτηση της απόστασης x*x→R όπου x είναι το σύνολο των λύσεων, δίδει πληροφορίες για την ποιότητα μεταξύ δύο λύσεων. Όταν $X \in \mathbb{R}^n$ η συνάρτηση της απόστασης δίδεται

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{n} \left(x_k^i - x_k^j\right)^2} \ \textbf{(10)}$$

όπου x_k^i είναι η k-οστή συνιστώσα της χωρικής συντεταγμένης x_i της i-οστής πυγολαμπίδας. Στον δισδιάστατο χώρο η απόσταση $r_{i,j}$ δίδεται:

$$r_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \ \textbf{(11)}$$

3.1.4 ΕΞΙΣΩΣΗ ΚΙΝΗΣΗΣ

Η εξίσωση κίνησης της πυγολαμπίδας i η οποία έλκεται από μια φωτεινότερη δίνεται ως εξής:

$$x_i = x_0 + \beta_0 \times e^{-y \times r_{i,j}^2} + \alpha \times (rand(\quad) - \tfrac{1}{2}) \ \textbf{(12)}$$

Ο δεύτερος όρος αντιστοιχεί στην ελκυστικότητα μεταξύ των πυγολαμπίδων i και j και ο τρίτος όρος είναι μια πιθανοκρατική παράμετρος. Συνήθως είναι $\beta_0 \in [0,1]$ και $\alpha \in [0,1]$.Ο συντελεστής απορρόφησης γ μεταβάλλεται μεταξύ 0,1 και 10. (Yang , 2010b[17])

3.1.5 ΛΕΙΤΟΥΡΓΙΑ ΑΛΓΟΡΙΘΜΟΥ

Τα κύρια στάδια του αλγόριθμου πυγολαμπίδας ξεκινούν από την αρχικοποίηση ένα σμήνους πυγολαμπίδων, όπου για κάθε πυγολαμπίδα καθορίζεται η ένταση του φωτός της. Κατά τη διάρκεια του βρόχου γίνεται σύγκριση κατά ζεύγη ως προς την ένταση φωτεινότητας της κάθε πυγολαμπίδας και η πυγολαμπίδα με την χαμηλότερη ένταση του φωτός θα κινηθεί προς την πυγολαμπίδα με την υψηλότερη ένταση. Η απόσταση εξαρτάται από την ελκυστικότητα. Μετά τη μετακίνηση, η νέα πυγολαμπίδα αξιολογείται και ενημερώνεται η ένταση της φωτεινότητας της. Κατά τη διάρκεια της σύγκρισης ανά ζεύγη στον βρόχο, συνεχώς θα προκύπτουν καλύτερες λύσεις μέσα από τις επαναληπτικές ενημερώσεις. Η διαδικασία σύγκρισης των ζευγών επαναλαμβάνεται έως ότου τα κριτήρια τερματισμού θα ισχύσουν. Τέλος, η βέλτιστη λύση θα είναι ορατή. (Khadwilard et.al. , 2012[18])

3.1.6 ΕΙΔΙΚΕΣ ΠΕΡΙΠΤΩΣΕΙΣ

Υπάρχουν δύο ειδικές περιπτώσεις του αλγορίθμου (FA) οι οποίες βασίζονται στην συμπεριφορά του συντελεστή απορρόφησης γ:

1. Όταν $\gamma \rightarrow 0$, τότε ο συντελεστής ελκυστικότητας είναι σταθερός $\beta = \beta_0$ και η ένταση φωτεινότητας δεν μειώνεται όταν η απόσταση μεταξύ δύο πυγολαμπίδων αυξάνεται. Αυτή η ειδική περίπτωση αντιστοιχεί στον αλγόριθμο Βελτιστοποίησης Σμήνους Σωματιδίων (PSO).

---

[17] σελ.1 - 6

[18] Khadwilard et.al. , (2012))

2. Όταν $\gamma \rightarrow \infty$, τότε ο συντελεστής ελκυστικότητας παράγεται από την συνάρτηση Dival $\beta(r) \rightarrow \delta(r)$. Σε αυτή την περίπτωση η ελκυστικότητα στην ένταση φωτεινότητας ισούται με μηδέν. Αυτή η περίπτωση αντιστοιχεί σε μια μέθοδο τυχαίας αναζήτησης.

3.1.7 ΡΥΘΜΙΣΗ ΠΑΡΑΜΕΤΡΩΝ

Το α ή α_t ελέγχει ουσιαστικά την τυχαιότητα ή, σε κάποιο βαθμό, την ποικιλία των λύσεων που έχουμε και μπορεί να ρυθμίζει την παράμετρο κατά τη διάρκεια των επαναλήψεων ώστε να μπορεί να ποικίλει ανάλογα με τον μετρητή επαναλήψεων t.

Έτσι, ένας καλός τρόπος για να εκφράσουμε το α_t είναι:

$$\alpha_t = \alpha_0 \times \delta^t, (0 < \delta < 1),$$

όπου α_0 είναι ο αρχικός παράγοντας κλιμάκωσης τυχαιότητας και δ είναι ουσιαστικά ένας παράγοντας ψύξης. Για τις περισσότερες εφαρμογές, μπορούμε να χρησιμοποιήσουμε δ=0,95 έως 0,97. Όσον αφορά την αρχική α_0, οι προσομοιώσεις δείχνουν ότι ο FA θα είναι πιο αποτελεσματικός εάν η α_0 σχετίζεται με την κλιμάκωση των μεταβλητών σχεδιασμού. Έστω L η μέση κλίμακα του προβλήματος που μας ενδιαφέρει, μπορούμε να θέσουμε α_0=0.01L αρχικά. Ο συντελεστής 0,01 προέρχεται από το γεγονός ότι οι τυχαίες βόλτες απαιτούν μια σειρά από μέτρα για την επίτευξη του στόχου, ενώσω η εξισορρόπηση της τοπικής εκμετάλλευσης χωρίς άλματα σε λίγα βήματα κατευθύνεται πολύ μακριά. Η παράμετρος ελέγχου της ελκυστικότητας και οι παραμετρικές μελέτες δείχνουν ότι το β_0=1 μπορεί να χρησιμοποιηθεί για τις περισσότερες εφαρμογές. Ωστόσο, θα πρέπει επίσης να σχετίζονται με την κλιμάκωση L.

Σε γενικές γραμμές, μπορούμε να καθορίσουμε $\gamma = 1/\sqrt{L}$. Εάν οι διακυμάνσεις κλιμάκωσης δεν είναι σημαντικές, τότε μπορούμε να θέσουμε γ=O(1). Για τις περισσότερες εφαρμογές, μπορούμε να χρησιμοποιήσουμε το μέγεθος του πληθυσμού n=15 έως 100, αν και η καλύτερη σειρά είναι n=25 έως 40.

3.1.8 ΨΕΥΔΟΚΩΔΙΚΑΣ ΑΛΓΟΡΙΘΜΟΥ ΠΥΓΟΛΑΜΠΙΔΑΣ

Στο σημείο αυτό θα παρουσιάσουμε τον ψευδοκώδικα του αλγορίθμου πυγολαμπίδας του Yang όπως φαίνεται στην Εικόνα 10.

Algorithm 1 Pseudo code of the base Firefly algorithm

```
t = 0; s* = ∅; γ = 1.0;          // initialize: gen.counter, best solution, attractivene
P^(0) = InitializeFA();           // initialize a population
while (t <MAX_FES) do
    α^(t)= AlphaNew();            // determine a new value of α
    EvaluateFA(P^(t), f(s));      // evaluate s according to f(s)
    OrderFA(P^(t), f(s));         // sort s according to f(s)
    s* = FindTheBestFA(P^(t), f(s));  // determine the best solution
    P^(t+1) = MoveFA(P^(t));      // vary the attractiveness accordingly
    t = t + 1;
end while
```

Εικόνα 10: Ψευδοκώδικας του Αλγόριθμου Πυγολαμπίδας

1. Η λειτουργία του «AlphaNew» είναι αφιερωμένη στο να τροποποιήσει την αρχική τιμή της παραμέτρου. Σημειωτέον ότι αυτό το βήμα είναι προαιρετικό στο σχετικό αλγόριθμο πυγολαμπίδας.

2. Η λειτουργία «EvaluateFA» αξιολογεί την ποιότητα της λύσης. Η εφαρμογή μιας συνάρτησης fittness f(α) εκτελείται μέσα σε αυτή.

3. Η λειτουργία «OrderFA» ταξινομεί τον πληθυσμό των πυγολαμπίδων σύμφωνα με τις τιμές fitness τους.

4. Η λειτουργία «FindTheBestFA» επιλέγει το καλύτερο άτομο από τον πληθυσμό.

5. Τέλος, η λειτουργία «MoveFA» εκτελεί μια κίνηση θέσης στις πυγολαμπίδες στο χώρο αναζήτησης.

Σημειωτέον ότι σχετικά με την μετακίνηση των πυγολαμπίδων προς τα πιο ελκυστικά άτομα, η διαδικασία αναζήτησης ελέγχεται από το μέγιστο αριθμό των fitness με την αξιολόγηση λειτουργία MAX FES.

3.1.9 ΑΛΓΟΡΙΘΜΙΚΗ ΠΟΛΥΠΛΟΚΟΤΗΤΑ

Σχεδόν όλοι οι μεταευρετικοί αλγόριθμοι είναι απλοί όσον αφορά την πολυπλοκότητα τους, και έτσι είναι εύκολο να εφαρμοστούν. Ο αλγόριθμος πυγολαμπίδας έχει δύο εσωτερικούς βρόχους οι οποίοι περνούν μέσα από τον πληθυσμό n, και ένα εξωτερικό βρόχο για την επανάληψη t. Έτσι, η πολυπλοκότητα στην ακραία περίπτωση είναι $O(n^2t)$. Καθώς το n είναι μικρό (τυπικά, n=40), και το t είναι μεγάλο (π.χ. t=5000), το κόστος υπολογισμού είναι σχετικά μικρό, επειδή η πολυπλοκότητα αλγορίθμου είναι γραμμική ως προς το t. Το κύριο υπολογιστικό κόστος θα είναι κατά τις αξιολογήσεις των αντικειμενικών συναρτήσεων, ειδικά για τους εξωτερικούς στόχους που αποτελούν το μαύρο κουτί της μεθόδου. Το τελευταίο είναι μια μεγάλη αλήθεια για όλους τους μεταευρετικούς αλγορίθμους.

Σε τελική ανάλυση, για όλα τα προβλήματα βελτιστοποίησης, το πιο εκτεταμένο υπολογιστικό τμήμα είναι οι αντικειμενικές εκτιμήσεις. Εάν το n είναι σχετικά μεγάλο, είναι δυνατόν να χρησιμοποιηθεί ένας εσωτερικός βρόχος για την κατάταξη της ελκυστικότητας ή της φωτεινότητας όλων των πυγολαμπίδων χρησιμοποιώντας αλγόριθμους ταξινόμησης. Σε αυτήν την περίπτωση, η πολυπολοκότητα του αλγορίθμου πυγολαμπίδας θα είναι O(nlog(n)).

3.2 ΤΑΞΙΝΟΜΗΣΗ ΚΑΙ ΑΝΑΛΥΣΗ ΑΛΓΟΡΙΘΜΩΝ ΠΥΓΟΛΑΜΠΙΔΑΣ

Μια ορισμένη ταξινόμηση είναι απαραίτητη προκειμένου να ταξινομηθεί ένα μεγάλος πλήθος αλγορίθμων πυγολαμπίδας. Ο ευκολότερος τρόπος για την επίτευξη αυτού του σκοπού είναι η διάκριση των αλγορίθμων πυγολαμπίδας σύμφωνα με τις ρυθμίσεις των παραμέτρων τους.

Οι ρυθμίσεις των παραμέτρων αυτών είναι ζωτικής σημασίας για την καλή απόδοση και, συνεπώς, επιλέγονται προσεκτικά από τους προγραμματιστές. Σε γενικές γραμμές, υπάρχουν δύο τρόποι για να αναζητήσουμε τις παραμέτρους του αλγορίθμου σωστά. Από τη μία πλευρά, κατά την ρύθμιση των παραμέτρων, οι καλές τιμές που βρέθηκαν πριν από την εκτέλεση του αλγορίθμου, διορθώθηκαν κατά τις επαναλήψεις. Από την άλλη, με τον έλεγχο των τιμών των παραμέτρων όταν είναι καθορισμένες κατά τις δοκιμές. Επιπλέον, η συμπεριφορά του FA δεν εξαρτάται μόνο από τις κατάλληλες τιμές των παραμέτρων, αλλά και από το τι στοιχεία ή τι χαρακτηριστικά ενσωματώνονται σε αυτόν. Η ενιαία κατάταξη θα πρέπει να είναι σε θέση να ταξινομήσει αλγόριθμους πυγολαμπίδας, σύμφωνα με τα στοιχεία που θα ληφθούν υπόψη. Μας ενδιαφέρει κυρίως:

1. Η τροποποίηση
2. Με ποιο τρόπο έγινε
3. Ποιος είναι ο σκοπός

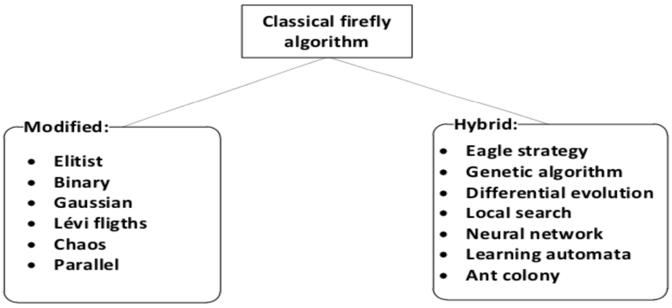
Μια άλλη εκδοχή κατάταξης είναι με βάση τα κατασκευαστικά στοιχεία ή τα χαρακτηριστικά τους. Αυτά είναι:

1. η εκπροσώπηση της πυγολαμπίδας (δυαδικό, real-valued),
2. ο σχεδιασμός του πληθυσμού (σμήνος, multi - swarm),
3. η αξιολόγηση της λειτουργίας fitness (αντικειμενικής συνάρτησης),
4. ο προσδιορισμός της βέλτιστης λύσης (μη - ελιτισμός, ελιτισμός),
5. η κίνηση της πυγολαμπίδας (ομοιόμορφη, Gaussian, Levy flight, διανομή χάους).

Όσον αφορά τη δεύτερη άποψη (έλεγχος παραμέτρων), οι αλγόριθμοι πυγολαμπίδας μπορούν να χωριστούν σε: ντετερμινιστικούς, προσαρμοστικούς και αυτοπροσαρμοστικούς. Τέλος, σύμφωνα με την τελευταία άποψη, η κατάταξη των αλγορίθμων πυγολαμπίδας μπορούν να επηρεαστούν από ένα στοιχείο της πυγολαμπίδας, ολόκληρη τη πυγολαμπίδα, ή το συνολικό πληθυσμό.

Στα αρχικά στάδια, ο FA δρα ως γενικός λύτης προβλημάτων. Δηλαδή, για αρκετά συνεχή προβλήματα βελτιστοποίησης, ο αλγόριθμος βρίσκει τις επιθυμητές λύσεις. Εξαιτίας δυσκολιών, δεν μπορεί να βρει τις κατάλληλες λύσεις για ορισμένα άλλα προβλήματα βελτιστοποίησης, σύμφωνα με το θεώρημα No-Free-Lunch. Η υβριδοποίηση έχει εφαρμοστεί σε αλγορίθμους βελτιστοποίησης για την επίλυση ενός δεδομένου συνόλου προβλημάτων. Οι αλγόριθμοι υβριδοποιούνται με άλλους αλγορίθμους βελτιστοποίησης, τεχνικές μηχανικής μάθησης, ευρετικές τεχνικές, κλπ. Η υβριδοποίηση μπορεί να λάβει χώρα σε σχεδόν κάθε στοιχείο του αλγόριθμου Πυγολαμπίδας, για παράδειγμα στη διαδικασία αρχικοποίησης, στη λειτουργία αξιολόγησης, στη συνάρτηση κίνησης, κλπ.

Οι αλγόριθμοι πυγολαμπίδας αναλύονται σύμφωνα με το σχήμα της Εικόνας 11, όπου οι κλασικοί αλγόριθμοι χωρίζονται σε τροποποιημένους και υβριδικούς. Οι κλασικοί αλγόριθμοι έχουν χρησιμοποιηθεί κυρίως για προβλήματα βελτιστοποίησης προκειμένου να παρέχουν τα βέλτιστα αποτελέσματα στην επίλυση διάφορων κατηγοριών προβλημάτων που έχουν υποστεί αρκετές τροποποιήσεις και υβριδισμούς. Οι κύριες κατηγορίες σχετίζονται με τον σχεδιασμό των τροποποιημένων αλγορίθμων και χωρίζονται σε ελιτίστικους, δυαδικούς αλγόριθμους πυγολαμπίδας, Gaussians, Levy flights και χαοτικούς, και βασίζονται στον αλγόριθμο πυγολαμπίδας και σε παρόμοιούς του. Από την άλλη πλευρά, οι ακόλουθοι υβριδικοί, εφαρμόζονται χρησιμοποιώντας τον κλασικό αλγόριθμο με χρήση στρατηγικής Eagle γενετικούς αλγόριθμους, τοπική αναζήτηση, νευρωνικά δίκτυα κλπ.



Εικόνα 11: Ταξινόμηση Αλγορίθμων Πυγολαμπίδας

3.3 ΠΑΡΑΔΕΙΓΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ ΠΥΓΟΛΑΜΠΙΔΑΣ

Ο αλγόριθμος της πυγολαμπίδας εξαρτάται κυρίως από την μεταβολή της έντασης του φωτός και την διατύπωση της ελκυστικότητας. Και οι δύο παράγοντες του επιτρέπουν σημαντικά περιθώρια για βελτιώσεις.

- Για να σταθεροποιηθεί η κίνηση της πυγολαμπίδας, ο Farahani διατύπωσε ένα νέο FA που αυξάνει την ταχύτητα σύγκλισης, χρησιμοποιώντας Gaussian κατανομή για να μετακινήσει όλες τις νέες πυγολαμπίδες σε ολικά καλύτερες λύσεις σε κάθε επανάληψη. Ωστόσο η τροποποιούμενη παράμετρος τυχαιοποίησης, ήταν τροποποιήσιμη προσαρμοστικά στον προτεινόμενο αλγόριθμο. Ο αλγόριθμος δοκιμάστηκε σε πέντε τυπικές συναρτήσεις. Τα πειραματικά αποτελέσματα έδειξαν καλύτερη απόδοση και μεγαλύτερη ακρίβεια από ό,τι στο κλασικό αλγόριθμο πυγολαμπίδας.
- Ο Yang διαμόρφωσε ένα νέο μεταευρετικό FA χρησιμοποιώντας τη Levy flight κίνηση. Οι Αριθμητικές μελέτες και τα αποτελέσματα δείχνουν ότι η προτεινόμενη μέθοδος είναι ανώτερη από τη βελτιστοποίηση σμήνους σωματιδίων και των γενετικών αλγόριθμων όσον αφορά την απόδοση και το ποσοστό επιτυχίας. Παρά το γεγονός ότι ο Yang προσπάθησε να ανακαλύψει κάποια μαθηματική θεμελίωση για τη συμπεριφορά των μεταευρετικών, κατέληξε στο συμπέρασμα ότι, παρά το γεγονός ότι οι νεοαποκτηθέντες εμπνευσμένοι από τη φύση μεταευρετικοί αλγόριθμοι

λειτουργούν καλά κατά μέσο όρο, η μαθηματική κατανόηση αυτών παραμένει εν μέρει ένα μυστήριο.

- Ο Coelho (Coelho et.al, ,2010[19]) πρότεινε ένα συνδυασμό FA με χαοτικούς χάρτες προκειμένου να βελτιωθεί η σύγκλιση του αλγόριθμου πυγολαμπίδας. Η χρήση των αλληλουχιών χάους δείχθηκε να είναι ιδιαίτερα αποτελεσματική και οδηγεί πιο αργά σε διαφυγή από τα τοπικά βέλτιστα Ο προτεινόμενος αλγόριθμος χρησιμοποιεί αυτούς τους χαοτικούς χάρτες ρυθμίζοντας την τυχαία παράμετρο α και το τελεστή απορρόφησης φωτός γ. Ένα σημείο αναφοράς της αξιοπιστίας της βελτιστοποίησης έχει εξεταστεί, προκειμένου να τονίσει τη δύναμη του προτεινόμενου FA χρησιμοποιώντας χαοτικούς χάρτες. Τα αποτελέσματα της προσομοίωσης του προτεινόμενου FA συγκρίθηκαν με άλλες παρόμοιες τεχνικές βελτιστοποίησης και αποκαλύφθηκε ότι ο προτεινόμενος αλγόριθμος τις ξεπέρασε ως προς τις γνωστές διαθέσιμες λύσεις.
- Ο Διακριτός Αλγόριθμος Πυγολαμπίδας (DFA) (Sayadi et. al, 2010[20]) είναι μια διακριτή εκδοχή του αλγόριθμου πυγολαμπίδας, που προτάθηκε από τον Sayadi, R. Ramezanian και N. Ghaffari-Nasab και μπορεί να λύσει αποτελεσματικά NP-hard προβλήματα. Ο DFA υπερτερεί από τους υπάρχοντες αλγόριθμους, όπως ο αλγόριθμος αποικίας μυρμηγκιών.
- Από την άλλη πλευρά, ο Gandomi εισήγαγε το χάος στο FA προκειμένου να αυξηθεί η γενική κινητικότητα αναζήτησης της ισχυρής ολικής βελτιστοποίησης. Χρησιμοποιώντας χαοτικούς χάρτες, οι ερευνητές συντόνισαν την ελκυστικότητα β_0 και το συντελεστή απορρόφησης του φωτός γ. Ανέλυσαν την επιρροή 12 διαφορετικών χαοτικών χαρτών για τη βελτιστοποίηση των λειτουργιών αναφοράς. Τα αποτελέσματα έδειξαν ότι κάποιοι χαοτικοί FA μπορούν να ξεπεράσουν σαφώς τον κλασικό FA.

[19] σελ. 517-521

[20] σελ. 1-10

- Ο πολυκριτηριακός αλγόριθμος πυγολαμπίδας, μια σημαντική μελέτη που διεξήχθη από τον Αποστολόπουλου και Βλάχο (Apostolopoulos &[21] Vlachos (2011)), η οποία παρέχει ένα λεπτομερές ιστορικό και την ανάλυση σε ένα ευρύ φάσμα προβλημάτων, συμπεριλαμβανομένης της δοκιμής πολυκριτηριακών προβλημάτων αποστολής φορτίου.
- Ένας ενδιαφέρον αλγόριθμος πυγολαμπίδας που προτείνεται για την επίλυση του προβλήματος βελτιστοποίησης στην δέσμευση των μονάδων στα ηλεκτρικά συστήματα από τον Rampriya το 2010 είναι ο Lagrangian FA (Rampriya et. al.[22]).

3.4 ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΑΛΓΟΡΙΘΜΟΥ ΠΥΓΟΛΑΜΠΙΔΑΣ

Στην ενότητα αυτή θα δούμε γιατί ο FA είναι τόσο αποτελεσματικός. Υπάρχουν πολλοί λόγοι για την επιτυχία του. Με την ανάλυση των κύριων χαρακτηριστικών του κλασικού αλγόριθμου, μπορούμε να επισημάνουμε τα εξής τρία σημεία:

1. Ο FA μπορεί να υποδιαιρεί αυτόματα τον πληθυσμό του σε υποομάδες, λόγω του γεγονότος αυτού η τοπική έλξη είναι ισχυρότερη από ό,τι σε μεγάλες αποστάσεις έλξης. Άρα, ο FA μπορεί να ασχοληθεί με εξαιρετικά γραμμικές, πολυτροπικές βελτιστοποιήσεις προβλημάτων με φυσικότητα και επιτυχία. Παρατηρούμε ότι:

    - Ο FA βασίζεται στην έλξη και στην ελκυστικότητα που μειώνονται με την απόσταση. Αυτό οδηγεί στο γεγονός ότι το σύνολο του πληθυσμού μπορεί να υποδιαιρεθεί αυτόματα σε υποομάδες και κάθε ομάδα να συρρέει γύρω από κάθε λειτουργία ή κάποιο τοπικό βέλτιστο. Ανάμεσα σε όλες αυτές τις καταστάσεις, η καλύτερη συνολική λύση μπορεί να βρεθεί.
    - Η υποδιαίρεση επιτρέπει στις πυγολαμπίδες να είναι σε θέση να βρουν όλα τα βέλτιστα ταυτόχρονα, εάν το μέγεθος του πληθυσμού είναι αρκετά μεγαλύτερο από τον αριθμό των λειτουργιών. Από μαθηματικής άποψης, το

[21] σελ. 1 - 23

[22] σελ. 389-393

κλάσμα $1/\sqrt{\gamma}$ ελέγχει την μέση απόσταση μιας ομάδας πυγολαμπίδων που μπορού να την δουν άλλες παρακείμενες ομάδες. Ένας ολόκληρος πληθυσμός μπορεί να υποδιαιρεθεί σε υποομάδες με βάση ένα δεδομένο, την μέση απόσταση. Σε ακραία περίπτωση, όταν γ=0, το σύνολο του πληθυσμού δεν θα υποδιαιρεθεί. Αυτή η αυτόματη ικανότητα υποδιαίρεσης καθίσταται ιδιαίτερα κατάλληλη για εξαιρετικά γραμμικά, πολυτροπικά προβλήματα βελτιστοποίησης.

2. Ο FA δεν χρησιμοποιεί ιστορικά μοναδικά βέλτιστα s_i*, ούτε υπάρχει ρητό ολικό βέλτιστο g*. Έτσι αποφεύγει το μειονέκτημα της πρόωρης σύγκλισης. Επιπλέον, ο FA δεν χρησιμοποιεί ταχύτητες αποφεύγονται προβλήματα όπως στην PSO.
3. Ο FA έχει την ικανότητα να ελέγχει την τροπικότητα και την προσαρμογή στο πρόβλημα ελέγχοντας τις παραμέτρους κλιμακώσεως του. Στην πραγματικότητα, μπορεί, λοιπόν να θεωρηθεί ως μια γενίκευση της βελτιστοποίησης σμήνους σωματιδίων (PSO), της διαφοροποιημένης εξέλιξης (DE) και της προσομοιωμένης ανόπτησης (SA).Μπορούμε να δούμε ότι, όταν το β_0 είναι μηδέν, ο τύπος του αλγορίθμου γίνεται ουσιαστικά μια έκδοση της παράλληλης προσομοιωμένης ανόπτησης, καθώς και το χρονοδιάγραμμα ανόπτησης ελέγχεται από το α. Αντίθετα, αν θέσουμε γ=0 και ρυθμιστεί η β_0=1 (ή β_0 є Unif(0,1)), ο FA γίνεται απλοποιημένη έκδοση της διαφοροποιημένης εξέλιξης χωρίς μετάλλαξη και ο ρυθμός διασταυρώσεων ελέγχεται από την β_0. Επιπλέον, αν θέσουμε γ=0 και αντικαταστήσουμε την s_j από την τρέχουσα ολική βέλτιστη λύση g*, τότε γίνεται μια παραλλαγή της PSO, ή πιο συγκεκριμένα επιταχυνόμενη βελτιστοποίηση σμήνους σωματιδίων. Ο αλγόριθμος πυγολαμπίδας περιλαμβάνει τους αλγορίθμους DE, PSO και SA ως ειδικές περιπτώσεις του. Ως αποτέλεσμα, ο FA μπορεί να έχει όλα τα πλεονεκτήματα αυτών των τριών αλγόριθμων .
4. Ο FA έχει την ικανότητα αντιμετώπισης της πολυτροπικότητας των παραμέτρων του. Οι παράμετροι του FA μπορούν να συντονίζονται για να ελέγχουν την τυχαιότητα ως ενεργές επαναλήψεις, έτσι ώστε η σύγκλιση να μπορεί επίσης να επισπευσθεί με ρύθμιση αυτών των παραμέτρων.

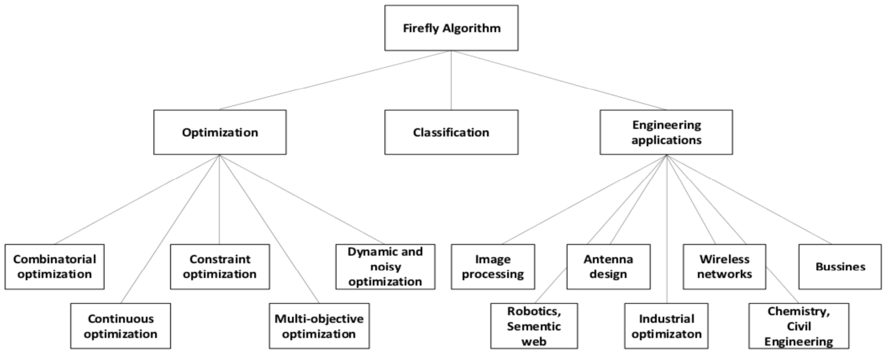
Ο αλγόριθμος της πυγολαμπίδας έχει επεκταθεί και εφαρμόζεται σε πολλά πεδία από την δημιουργία του μέχρι σήμερα. Δεν υπάρχει τομέας που να μην έχει εφαρμοστεί. Επιπλέον, οι περιοχές ανάπτυξης του αλγορίθμου είναι πολύ δυνατές, διότι νέες εφαρμογές εμφανίζονται κάθε μέρα. Ο αλγόριθμος της πυγολαμπίδας:

- Διαθέτει πολυτροπικά χαρακτηριστικά
- Χειρίζεται πολυτροπικά προβλήματα επαρκώς
- Έχει γρήγορο ρυθμό σύγκλισης
- Μπορεί να χρησιμοποιηθεί ως γενικός, ολικός λύτης προβλημάτων τόσο καλά όσο η ευχετική τοπική αναζήτηση
- Εφαρμόζεται σε κάθε τομέα και στα προβλήματα του

Ο αλγόριθμος της πυγολαμπίδας λύνει τα προβλήματα από τους ακόλουθους τομείς: συνεχής βελτιστοποίησης, συνδυαστικής βελτιστοποίησης, βελτιστοποίησης περιορισμών, πολυκριτηριακής βελτιστοποίησης, καθώς και δυναμικών θορυβώδων περιβαλλόντων ακόμη και ταξινομήσεων. Επίσης είναι κατάλληλος για πολυτροπικές βελτιστοποιήσεις, που έχουν γρήγορη σύγκλιση,δίνει καλά αποτελέσματα για τη βελτιστοποίηση των συναρτήσεων, και επίσης είναι κατάλληλος για συνδυαστική βελτιστοποίηση. Η γρήγορη σύγκλιση κάνει τον αλγόριθμο να χάνει την ποικιλομορφία του πληθυσμού. Για να συνεχίσει να χρησιμοποιείται αυτός ο αλγόριθμος για προβλήματα βελτιστοποίησης μεγάλης κλίμακας, πρέπει να εξισορροπηθούν και να καθορισθούν η εξερεύνηση και η εκμετάλλευση. Τα αποτελέσματα του αλγόριθμου εξαρτώνται από την καλύτερη εύρεση λύσης από ένα σμήνος. Γι’ αυτό, η βελτίωση των καλύτερων λύσεων μπορεί να βελτιώσει την δύναμη της αναζήτηση του σμήνους. Πειράματα με τη χρήση δυαδικών αλγόριθμων πυγολαμπίδας μπορούν να εφαρμοσθούν επιτυχώς σε αυτά τα προβλήματα.

3.5 ΕΦΑΡΜΟΓΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΠΥΓΟΛΑΜΠΙΔΑΣ

Σήμερα, ο FA και οι παραλλαγές του έχουν εφαρμοστεί για την επίλυση πολλών προβλημάτων βελτιστοποίησης και ταξινόμησης, καθώς επίσης και σε διάφορα τεχνικά προβλήματα στην πράξη. Η ταξινόμηση των αναπτυγμένων εφαρμογών του αλγόριθμου απεικονίζεται στην Εικόνα 12. Όπως μπορεί να φανεί, ο αλγόριθμος πυγολαμπίδας, εφαρμόζεται στις ακόλουθες κατηγορίες προβλημάτων βελτιστοποίησης: συνεχής, συνδυαστική, με περιορισμούς, πολλαπλών μεταβλητών, δυναμικής και θορυβώδους βελτιστοποίησης. Επιπλέον, έχει χρησιμοποιηθεί για την ταξινόμηση σε προβλήματα βελτιστοποίησης της: μηχανικής μάθησης, εξαγωγής δεδομένων, και των νευρωνικών δικτύων. Τέλος, οι αλγόριθμοι χρησιμοποιούν τον αλγόριθμο της πυγολαμπίδας σε όλους σχεδόν τους κλάδους της μηχανικής.



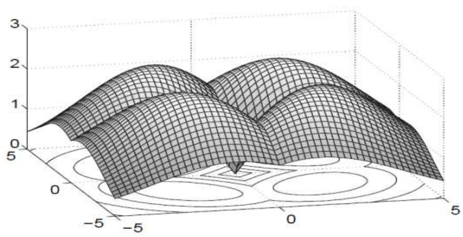
Εικόνα 12: Ταξινόμηση εφαρμογών Αλγορίθμου Πυγολαμπίδας

3.6 ΠΑΡΑΔΕΙΓΜΑΤΑ

Ας δούμε δύο παραδείγματα για να αποδείξουμε το υπολογιστικό κόστος που εξοικονομείται από τον FA.

ΠΑΡΑΔΕΙΓΜΑ 1^{o}

- ✓ Για τη συνάρτηση De Jong, Εικόνα 13, με d=256 διαστάσεις
- ✓ $f(x) = \sum_{i=1}^{256} x_i^2$



Εικόνα 13: Γραφική παράσταση συνάρτησης De Jong

Πηγή:X-S. Yang,(2013),σελ. 5

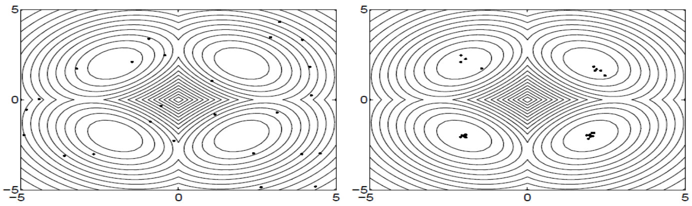
Ο γενετικός αλγόριθμος απαιτεί 25412±1237 υπολογισμούς για απόκτηση ακρίβειας της τάξης 10^{-5} από τις βέλτιστες λύσεις ενώ ο PSO χρειάζεται 17040±1123 υπολογισμούς. Με τον FA πετύχαμε την ίδια ακρίβεια με 17040±1123 υπολογισμούς. Έτσι εξοικονομήσαμε περίπου από 78% έως 67% υπολογιστικό κόστος, σε σύγκριση με τους αλγορίθμους GA και PSO, αντίστοιχα.

ΠΑΡΑΔΕΙΓΜΑ 2^{o}

- ✓ Για την συνάρτηση Yang:
- ✓ $f(x) = \sum_{i=1}^{d} |x_i| \times e(-\sum_{i=1}^{d} \sin(x_i^2)),\ -2\pi \leq x_i\ \leq 2\pi$

ο GA απαιτεί 37079±8920 επαναλήψεις με ποσοστό επιτυχίας 88% για d=16 και ο PSO απαιτεί 19725±3204 με ποσοστό επιτυχίας 98%. Ο FA έχει ποσοστό επιτυχίας 100%, με μόνο 5152±2493 επαναλήψεις. Σε σύγκριση με τους GA και PSO, ο FA αποθηκεύει περίπου το 86% και 74% των συνολικών υπολογιστικών προσπαθειών αντίστοιχα. Ως παράδειγμα για την αυτόματη υποδιαίρεση, μπορούμε πλέον να χρησιμοποιήσουμε τον FA για να βρούμε τα ολικά μέγιστα της συνάρτησης Yang με $-10 \leq x_i \leq 10$ για όλα τα (i=1, 2,..., d) όπου d είναι ο αριθμός των διαστάσεων. Η λειτουργία αυτή έχει πολλαπλά ολικά βέλτιστα. Στην περίπτωση που d=2, έχουμε 4 ίσα μέγιστα. Δηλαδή, με $f = 1/\sqrt{e} \approx 0.6065$ έχουμε τα εξής ολικά μέγιστα: (1/2, 1/2), (1/2,−1/2), (−1/2, 1/2) και (−1/2,−1/2) καθώς και ένα μοναδικό ολικό ελάχιστο στο (0,0).

Στην περίπτωση 2D, έχουμε μια συνάρτηση τεσσάρων κορυφών που φαίνοται στο ακόλουθο σχήμα, όπου τα ολικά μέγιστα μπορούν να βρεθούν χρησιμοποιώντας τον εφαρμοσμένο Αλγόριθμο Πυγολαμπίδας μετά από περίπου 500 συναρτησιακούς υπολογισμούς. Αυτό αντιστοιχεί σε 25 πυγολαμπίδες που εξελίσσονται μέσα σε 20 γενιές ή επαναλήψεις. Οι αρχικές θέσεις των 25 πυγολαμπίδων και οι τελικές θέσεις τους μετά από 20 επαναλήψεις εμφανίζονται στην Εικόνα 14. Μπορούμε να δούμε ότι ο FA μπορεί πράγματι να υποδιαιρέσει αυτόματα τον πληθυσμό σε υποομάδες, στην Εικόνα 14.



Εικόνα 14: Οι αρχικές θέσεις και οι τελικές θέσεις των πυγολαμπίδων μετά από 20 επαναλήψεις

Πηγή:X-S. Yang,(2013),σελ. 6

ΚΕΦΑΛΑΙΟ 4. ΥΒΡΙΔΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Υβριδικοί αλγόριθμοι ονομάζονται οι αλγόριθμοι που προκύπτουν από τον συνδυασμό δυο ή περισσοτέρων αλγορίθμων με σκοπό την λύση ενός προβλήματος. Αυτό γίνεται είτε επιλέγοντας έναν σύμφωνα με τα δεδομένα του προβλήματος, είτε επιλέγοντας πολλούς κατά την διάρκεια εκτέλεσης του αλγορίθμου. Η βασική ιδέα στηρίζεται στο ότι ο συνδυασμός πολλών χαρακτηριστικών από τον κάθε αλγόριθμο θα δημιουργήσει έναν καλύτερο απ' ότι ο καθένας είναι ξεχωριστά. Άρα δεν είναι απλά ένας πολλαπλός αλγόριθμος που λύνει ένα διαφορετικό πρόβλημα, αλλά πολλοί αλγόριθμοι που συνδυάζονται και μόνο ο συνδυασμός τους λύνει το πρόβλημα, ωστόσο έχουν διαφορές ως προς τα χαρακτηριστικά και την απόδοση τους.

Οι υβριδικοί αλγόριθμοι χρησιμοποιούνται συχνά στην βελτιστοποίηση και σε εφαρμογές αναδρομικών αλγορίθμων του πραγματικού κόσμου, όπως για παράδειγμα σε αλγορίθμους διαίρει και βασίλευε, όπου το μέγεθος των δεδομένων μειώνεται καθώς προχωράμε βαθύτερα στην αναδρομή. Σε αυτή την περίπτωση, ένας αλγόριθμος χρησιμοποιείται για τη συνολική προσέγγιση (σε μεγάλα δεδομένα), αλλά βαθιά στην αναδρομή μεταβαίνει σε έναν διαφορετικό αλγόριθμο, ο οποίος είναι πιο αποτελεσματικός σε μικρού μεγέθους δεδομένα. Ένα κοινό παράδειγμα είναι οι αλγόριθμοι ταξινόμησης, όπως ο quicksort. Μια γενική διαδικασία για έναν απλό υβριδικό αναδρομικό αλγόριθμο είναι να βραχυκυκλώνει τη βασική περίπτωση, επίσης γνωστή ως πλήρους ανταγωνισμού αναδρομή.

Η ιδέα του συνδυασμού δύο ή περισσοτέρων αλγορίθμων σε ένα υβριδικό εμπνεύστηκε από την πιθανότητα ότι ο νέος αλγόριθμος θα έλυνε καλύτερα ένα πρόβλημα από τους μεμονωμένους αλγόριθμους που τον απαρτίζουν. Το αποτέλεσμα ήταν ένας νέος αλγόριθμος με τις προδιαγραφές των **Υβριδικών Αλγοριθμικών Τεχνικών** (HAT). Ο αλγόριθμος θα είχε την δύναμη των αλγορίθμων που συνδύαζε και τα εξής πλεονεκτήματα:

1. μπορεί να παράγει νέες λύσεις
2. οι λύσεις που θα παράγει θα βρίσκονται σε λιγότερο χρόνο
3. θα χειρίζεται αποδοτικά προβλήματα με μεγάλο μέγεθος εισόδου, και ειδικά προβλήματα NP

Έγινε προσπάθεια να μελετηθεί αν οι διαφορετικές υβριδικές τεχνικές εφαρμόζονται σε προβλήματα συνδυαστικής βελτιστοποίησης και έτσι ονομάστηκαν **Υβριδικοί Αλγόριθμοι Αναζήτησης Τεχνικών (HAST)**. Με σκοπό την εγγύηση της βέλτιστης λύσης, λαμβάνουμε υπόψη όλες τις πιθανές λύσεις. Δυστυχώς πολλά προβλήματα που ανήκουν στην κλάση NP–complete, έχουν τεράστιο εύρος λύσεων που δεν μπορούμε να το λάβουμε υπόψη μας. Οι ευρετικές τεχνικές χρησιμοποιούνται μόνο για να την δοκιμή πιθανών υποσυνόλων λύσεων και οι υπάρχοντες αλγόριθμοι δεν μπορούν να διαβεβαιώσουν την εύρεση βέλτιστων λύσεων. Υπάρχουν πολλοί αλγόριθμοι για την επίλυση προβλημάτων συνδυαστικής βελτιστοποίησης, η υβριδοποίηση αυτών αποσκοπεί στον συνδυασμό τους με ευρετικές τεχνικές για την παραγωγή ενός καλύτερου αλγορίθμου. Ο νέος αλγόριθμος θα παράγει λύσεις πολύ κοντά στις βέλτιστες ή σε μικρότερο χρόνο ή και τα δυο μαζί. Ο αλγόριθμος που παράγει ικανοποιητικά αποτελέσματα σε μικρότερο χρόνο μπορεί επίσης να χρησιμοποιηθεί σε μεγαλύτερα προβλήματα.

Στο κεφάλαιο αυτό θα δούμε παραδείγματα από κάποιους υβριδικούς αλγορίθμους που συνδυάζουν είτε τον αλγόριθμο αρμονίας, είτε πυγολαμπίδας ή και τους δυο μαζί. (Malek et. al.1989[23])

4.1 ΥΒΡΙΔΙΚΟΣ ΑΛΓΟΡΙΘΜΟΣ ΠΥΓΟΛΑΜΠΙΔΑΣ ΜΕ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΟΥ ΑΡΜΟΝΙΑΣ HSFA

Μια υβριδική μεταευρετική μέθοδος με βάση τον αλγόριθμο Αναζήτηση Αρμονίας (HS) και τον αλγόριθμο Πυγολαμπίδα (FA), παρουσιάζεται το 2013 από τους Wang και Guo για να λύσει συναρτήσεις βελτιστοποίησης. Ο υβριδικός αλγόριθμος ονομάζεται Υβριδικός Αλγόριθμος Πυγολαμπίδας με χρήση Αλγορίθμου Αναζήτησης Αρμονίας (HSFA). Στόχος του είναι να εκμεταλλευτεί τα πλεονεκτήματα των HS και FA, έτσι ώστε να έχει μεγαλύτερη ταχύτητας σύγκλισης από τους μεμονωμένους HS και FA. Επίσης, εισάγεται η έννοια των «κορυφαίων πυγολαμπίδων» με σκοπό να μειωθεί ο χρόνος εκτέλεσης. Επιπλέον ο HS χρησιμοποιείται για να μεταλλάξει μεταξύ τους τις πυγολαμπίδες κατά την ενημέρωση. Η μέθοδος HSFA επαληθεύεται από διάφορα σημεία αναφοράς. Τα πειράματα απέδειξαν ότι η

[23] σελ.1 - 2

εφαρμογή του HSFA δίνει καλύτερα αποτελέσματα από ότι ο κλασικός FA αλλά και από άλλες οκτώ μεθόδους βελτιστοποίησης.

Ο Yang το 2008 πρότεινε τον αλγόριθμο Πυγολαμπίδας που προέρχεται από ένα σμήνος πυγολαμπίδων, και γνωρίσαμε ότι είναι αρκετά ισχυρός και αποτελεσματικός. Επιπλέον η απόδοση του μπορεί να βελτιωθεί και κατά συνέπεια να παράγει ελπιδοφόρα εφικτά αποτελέσματα. Οι ερευνητές προσπαθούν να βρουν μια βέλτιστη λύση μέσα από την αντικειμενική συνάρτηση, για να πετύχουν την βελτιστοποίηση ενός προβλήματος. Ενώ στον μουσικό αυτοσχεδιασμό, οι μουσικοί αναζητούν την πιο ικανοποιητική αρμονία όπως καθορίζεται από την αισθητική. Η μέθοδος της Αναζήτησης Αρμονίας έχει πολλά κοινά με αυτήν την διαδικασία.

Στις περισσότερες περιπτώσεις ο FA μπορεί να βρει τις βέλτιστες λύσεις. Ωστόσο η αναζήτηση που κάνει βασίζεται στην τυχαιότητα, έτσι δεν μπορεί πάντα να εγγυάται για την εύρεση των βέλτιστων λύσεων. Από την άλλη προκειμένου να βελτιωθεί η ποικιλομορφία των πυγολαμπίδων, ο HS μπορεί να προσθέσει μια βελτίωση, η οποία μπορεί να αντιμετωπίζεται σαν ένας φορέας μετάλλαξης. Με τον συνδυασμό του αρχικού HS και FA, παράγεται ένας νέος υβριδικός FA για την αναζήτηση των καλύτερων τιμών της αντικειμενικής συνάρτησης.

Ωστόσο ο FA χρειάζεται πολύ περισσότερο χρόνο για να ψάξει για την καλύτερη λύση και η απόδοση του χειροτερεύει σημαντικά με την αύξηση του μεγέθους πληθυσμού. Στον HSFA το σύστημα με τις «κορυφαίες πυγολαμπίδες» εισάγεται για να μειώσει τον χρόνο εκτέλεσης κάτι που πραγματοποιείται με την μείωση του εξωτερικού βρόχου στον FA. Μέσω του συστήματος των κορυφαίων πυγολαμπίδων, η χρονική πολυπλοκότητα μειώνεται από $O(NP^2)$ σε O(KEEP*NP), όπου KEEP είναι ο αριθμός των «κορυφαίων πυγολαμπίδων». Αυτή η προσέγγιση αξιολογείται από διάφορα σημεία αναφοράς. Τα αποτελέσματα δείχνουν ότι ο HSFA εκτελείται πιο αποτελεσματικά και με ακρίβεια σε σχέση με τον FA και άλλους ευφυείς αλγόριθμους.

4.1.1 ΒΑΣΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ ΤΟΥ HSFA

Ας θυμηθούμε κάποια χαρακτηριστικά των αλγορίθμων Αναζήτησης Αρμονίας και Πυγολαμπίδας, πριν δούμε την υλοποίηση του Υβριδικού Αλγορίθμου τους.

A. ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ ΑΡΜΟΝΙΑΣ (HS)

Ο HS ως σχετική τεχνική βελτιστοποίησης έχει τους εξής φορείς:

HM: η μνήμη αρμονίας, **HMS**: το μέγεθος της μνήμης αρμονίας, **HMCR**: ο Δείκτη Αποδοχής Αρμονικής Μνήμης, **PAR**: ο Δείκτης Προσαρμογής του Τονικού Ύψους, και $\mathbf{b_w}$: το εύρος ζώνης ρύθμισης βήματος.

Η μέθοδος HS μπορεί να εξηγηθεί σύμφωνα με την εξής διαδικασία.

Ο μουσικός έχει 3 εφικτές επιλογές κατά την διαδικασία του μουσικού αυτοσχεδιασμού:

1. παίζει πολλά μουσικά κομμάτια που είναι ίδια με αυτά στην HMCR
2. παίζει ένα μουσικό κομμάτι λίγο παραλλαγμένο από ένα γνωστό
3. αυτοσχεδιάζει ένα νέο μουσικό κομμάτι

Αυτές οι τρεις επιλογές μπορούν να εξιδανικεύονται σε τρεις συνιστώσες: η χρήση της μνήμης HM, η ρύθμιση τόνου και η τυχαιοποίηση.

Παρόμοια με την επιλογή της βέλτιστης λύσης στους γενετικούς αλγορίθμους, ο HS μπορεί να εγγυάται ότι οι βέλτιστες αρμονίες δεν θα καταστραφούν από την μνήμη HM. Για να γίνει ο HS πιο ισχυρός, η παράμετρος HMCR πρέπει να ρυθμιστεί σωστά. Μέσα από πολλά πειράματα, στις περισσότερες περιπτώσεις, η παράμετρος παίρνει τις τιμές HMCR=0.7~0.95. Ο τόνος στο δεύτερο μέρος πρέπει να προσαρμοστεί ελαφρώς, γι' αυτόν τον λόγο θα πρέπει να χρησιμοποιείται μια κατάλληλη μέθοδος για να ρυθμίσει τη συχνότητα.

Εάν ο νέος τόνος x_{new} ενημερώνεται από την

$$x_{new} = x_{old} + b_w \times \epsilon - 1), (2)$$

όπου ϵ είναι ένας τυχαίος αριθμός στο [0, 1] και x_{old} είναι ο τρέχον μουσικός τόνος. Εδώ, το b_w είναι το εύρος ζώνης. Η παράμετρος PAR θα πρέπει επίσης να ρυθμιστεί κατάλληλα. Αν η PAR είναι πολύ κοντά στο 1, τότε η λύση είναι πάντα ενημερωμένη και ο HS είναι δύσκολο να συγκλίνει. Αν είναι κοντά στο 0, τότε ο HS μπορεί συγκλίνει πρόωρα. Έτσι, εδώ θέτουμε τις τιμές PAR=0.1~0.5 [1].

Για να βελτιωθεί η ποικιλομορφία, απαραίτητη είναι η τυχαιοποίηση. Η τυχαιοποίηση επιτρέπει στην μέθοδο να αναζητήσει πιο πολλά υποσχόμενες περιοχές, ώστε να βρει την βέλτιστη λύση. Ο ψευδοκώδικας που περιγράφει τον HS μπορεί να παρουσιαστεί στον Αλγόριθμο της Εικόνας 15. Όπου D είναι ο αριθμός των μεταβλητών απόφασης και rand είναι ένας τυχαίος πραγματικός αριθμός στο διάστημα (0, 1) που προέρχονται από μια ομοιόμορφη κατανομή.

```
Begin
    Step 1. Initialize the HM.
    Step 2. Evaluate the fitness.
    Step 3. while the halting criteria is not satisfied do
        for d = 1:D do
            if rand < HMCR then        // memory consideration
                x_new(d) = x_a(d) where a ∈ (1, 2, ..., HMS)
                if rand < PARthen          // pitch adjustment
                    x_new(d) = x_old(d) + bw × (2 × rand − 1)
                endif
            else                               // random selection
                x_new(d) = x_min,d + rand × (x_max,d − x_min−d)
            endif
        endfor d
        Update the HM as x_w = x_new, if f(x_new) < f(x_w) (minimization objective)
        Update the best harmony vector
    Step 4. end while
    Step 5. Output results.
End.
```

Εικόνα 15: Παραλλαγή Ψευδοκώδικα του Αλγορίθμου Αναζήτησης Αρμονίας

Β. ΑΛΓΟΡΙΘΜΟΣ ΠΥΓΟΛΑΜΠΙΔΑΣ (FA)

Τα δυο σημαντικά ζητήματα του αλγορίθμου είναι η ελκυστικότητα και η μεταβολή της έντασης του φωτός. Για λόγους απλότητας, πολλά χαρακτηριστικά των πυγολαμπίδων εξιδανικεύονται σε τρεις κανόνες που περιγράφονται στο κεφάλαιο 3. Με βάση αυτούς τους τρεις κανόνες, ο ψευδοκώδικας του FA περιγράφεται στην Εικόνα 16.

```
Begin
    Step 1. Initialization. Set G = 1; define γ; set step size α and β0 at r = 0.
    Step 2. Evaluate the light intensity I determined by f(x)
    Step 3. While G < MaxGeneration do
                for i = 1 : NP (all NP fireflies) do
                    for j = 1 : NP (NP fireflies) do
                        if (Ij < Ii),
                            move firefly i towards j;
                        end if
                    Update attractiveness;
                    Update light intensity;
                    end for j
                end for i
                G = G + 1;
    Step 4. end while
    Step 5. Output the results.
End.
```

Εικόνα 16: Παραλλαγή   Ψευδοκώδικα του Αλγορίθμου Πυγολαμπίδας

Δυο πυγολαμπίδες μπορούν να ενημερώνονται ως εξής:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_i^t - x_j^t\right) + \alpha \epsilon_i^t \quad , \qquad (3)$$

όπου α είναι το μέγεθος του βήματος, b_0 είναι η ελκυστικότητα για r=0, το δεύτερο μέρος είναι η έλξη, ενώ το τρίτο είναι η τυχαιότητα. Στην παρούσα εργασία μας, παίρνουμε $\beta_0 = 1$, $\alpha \in [0,1]$, και γ=1.

4.1.2 ΥΛΟΠΟΙΗΣΗ ΑΛΓΟΡΙΘΜΟΥ HSFA

Με βάση την εισαγωγή στους FA και HS ο συνδυασμός των δυο προσεγγίσεων περιγράφεται ως HSFA. Αυτός επικυρώνει τις μη βέλτιστες λύσεις για να επιταχύνει την ταχύτητα της σύγκλισης. Ο HS και ο FA έχουν εμπειρία στην εξερεύνηση του χώρου αναζήτησης και αντίστοιχη αξιοπιστία στις λύσεις. Με την στρατηγική μετάλλαξη που προσφέρει ο HS δίνεται η δυνατότητα στον FA να εξερευνήσει το νέο διάστημα έρευνας και να αξιοποιήσει αντίστοιχα τον πληθυσμό. Επίσης μπορεί να ξεπερασθούν οι έλεγχοι τις εξερεύνησης του FA. Για την καταπολέμηση των τυχαίων περιπάτων του FA, οι Wang & L. Guo πρότειναν να έγινε η προσθήκη του φορέα μετάλλαξης, ο οποίος περιλαμβάνει δυο βελτιώσεις.

Η πρώτη είναι η εισαγωγή του συστήματος «κορυφαίων πυγολαμπίδων» στον FA, με σκοπό να μειωθεί ο χρόνος εκτέλεσης, ο οποίος είναι ανάλογος με τον ελιτισμό του συστήματος. Στον FA λόγω του διπλού βρόχου η χρονική πολυπλοκότητα είναι $O(NP^2)$ και η απόδοση του χειροτερεύει σημαντικά με την αύξηση του μεγέθους πληθυσμού. Αυτή η βελτίωση πραγματοποιείται από την αναγωγή του εξωτερικού βρόχου στον FA. Στον HSFA επιλέγουμε συγκεκριμένες πυγολαμπίδες με βέλτιστες ή σχεδόν βέλτιστες τιμές της συνάρτησης fitness (πχ την λαμπρότερη πυγολαμπίδα). Αυτό γίνεται για να σχηματίσουμε ένα σύνολο από κορυφαίες πυγολαμπίδες ώστε να προσελκύουν τις υπόλοιπες με τις μικρότερες τιμές. Μέσω του συστήματος κορυφαίων πυγολαμπίδων η χρονική πολυπλοκότητα του HSFA μειώνεται από $O(NP^2)$ σε O(KEEP* NP), όπου KEEP είναι ο αριθμός των κορυφαίων πυγολαμπίδων. Σε γενικές γραμμές, η KEEP είναι πολύ μικρότερη χρονικά από O(NP) και έτσι ο χρόνος εκτέλεσης του HSFA είναι πολύ μικρότερος από τον FA .Προφανώς αν KEEP=NP ο αλγόριθμος HSFA έχει την ίδια πολυπλοκότητα με τον FA. Αν η KEEP παίρνει πολύ μικρές τιμές μόνο λίγες πυγολαμπίδες επιλέγονται για να αποτελούν τις κορυφαίες και έτσι η σύγκλιση θα είναι ταχύτατη, άλλωστε η πρόωρη σύγκλιση μπορεί να μην παράγει διαφορετικές λύσεις από τις υπάρχουσες. Αν η KEEP παίρνει πολύ μεγάλες τιμές (κοντά στο NP) σχεδόν όλες οι πυγολαμπίδες που χρησιμοποιούνται για να σχηματίσουν τις κορυφαίες, θα επιλεγούν και θα είναι πιθανόν οι βέλτιστες. Με αποτέλεσμα ο αλγόριθμος να εκτελεστεί άσχημα και με πολύ αργή σύγκλιση. Συνήθως χρησιμοποιούμε KEEP=2.

Η δεύτερη είναι η προσθήκη του HS που χρησιμοποιείται ως χειριστής μετάλλαξης και προσπαθεί να βελτιώσει την ποικιλομορφία του πληθυσμού για να αποφευχθεί η πρόωρη σύγκλιση. Στον αρχικό FA αν η πυγολαμπίδα i είναι φωτεινότερη από την πυγολαμπίδα j, η πυγολαμπίδα j θα κινηθεί προς την πυγολαμπίδα i και έπειτα υπολογίζονται οι νέας γενιάς πυγολαμπίδες και τροποποιείται-ενημερώνεται η ένταση φωτός. Αν δεν είναι φωτεινότερη, η πυγολαμπίδα j δεν κάνει τίποτα. Ωστόσο στον HSFA αν η πυγολαμπίδα i δεν είναι φωτεινότερη από την j, η j θα υποστεί μετάλλαξη ώστε να βελτιωθεί η ένταση φωτεινότητας της. Πιο συγκεκριμένα, στον HSFA για γενικά τμήματα αναζήτησης συντονίζουμε κάθε στοιχείο x_{kj} (k=1,2,3,…,D) σε x_j (η θέση της πυγολαμπίδας j) χρησιμοποιώντας τον αλγόριθμο HS. Όταν ένα ξ1 (τυχαίος ακέραιος) δεν είναι μικρότερος από τον HMCR ,δηλαδή ξ1≥HMCR, το στοιχείο x_{kj} ενημερώνεται τυχαία οπουδήποτε, και όταν ξ1 < HMCR, ενημερώνουμε το x_{kj} με βάση το x_{r_l}. Κάτω από αυτές τις συνθήκες, η λειτουργία προσαρμογής τόνου του HS εφαρμόζεται για την ενημέρωση του στοιχείου x_{kj} , ώστε αν ξ2<PAR να γίνει αύξηση της ποικιλομορφίας του πληθυσμού, όπου ξ1 και ξ2 είναι δυο

τυχαίοι ομοιόμορφα κατανεμημένοι αριθμοί στο [0,1], ο r_l είναι ακέραιος αριθμός στο [1,NP] και ο NP είναι το μέγεθος του πληθυσμού. G.

Στην Εικόνα 17 παρουσιάζεται ο ψευδοκώδικας του υβριδικού αλγόριθμου Πυγολαμπίδας με χρήση αλγορίθμου Αρμονικής Αναζήτησης (HSFA).

```
Begin
    Step 1. Initialization. Set t = 1; define γ; set α, β0 at r = 0; set HMCR and PAR; set the
            number of top fireflies KEEP.
    Step 2. Evaluate the light intensity I.
    Step 3. While  t < MaxGeneration do
            Sort the fireflies by light intensity I;
            for  i = 1 : KEEP (all Top fireflies) do
                for  j = 1 : NP (all fireflies) do
                    if  (Ij < Ii)  then
                        Move firefly i towards j;
                    else
                        for  k = 1 : D (all elements) do      // Mutate
                            if  (rand < HMCR) then
                                r1 = ⌈NP ∗ rand⌉
                                xv(k) = xr1(k)
                                if (rand < PAR) then
                                    xv(k) = xv(k) + bw × (2 × rand − 1)
                                end if
                            else
                                xv(k) = xmin,k + rand × (xmax,k − xmin,k)
                            end if
                        end for  k
                    end if
                    Update attractiveness;
                    Update light intensity;
                end for  j
            end for  i
            Evaluate the light intensity I.
            Sort the population by light intensity I;
            t = t + 1;
    Step 4. end while
End.
```

Εικόνα 17: Ο Ψευδοκώδικας της μεθόδου HSFA

4.1.3 ΑΠΟΤΕΛΕΣΜΑΤΑ

Η μέθοδος HSFA δοκιμάστικε σε προβλήματα βελτιστοποίησης μέσα από διάφορες προσομοιώσεις που διεξήχθησαν σε προβλήματα ελέγχου. Για να κάνουμε μια δίκαιη σύγκριση μεταξύ των διαφορετικών μεθόδων, όλα τα πειράματα έγιναν με τους ίδιους όρους. Ο HSFA συγκρίθηκε σε προβλήματα βελτιστοποίησης με άλλες εννιά μεθόδους, οι οποίες είναι ACO, BBO, DE, ES, FA, GA, HS, PSO και SGA. Για τον HSFA οι παράμετροι του ρυθμίστηκαν ως εξής: γ=1.0, HMCR=0.9 και PAR=0.1. Τριάντα έξι συναρτήσεις δοκιμάστηκαν από τον HSFA και βρίσκονται στον πίνακα 1.

No. Name	No. Name
F01 Beale	F19 Holzman 2 function
F02 Bohachevsky #1	F20 Levy
F03 Bohachevsky #2	F21 Pathological function
F04 Bohachevsky #3	F22 Penalty #1
F05 Booth	F23 Penalty #2
F06 Branin	F24 Powel
F07 Easom	F25 Quartic with noise
F08 Foxholes	F26 Rastrigin
F09 Freudenstein-Roth	F27 Rosenbrock
F10 Goldstein-Price	F28 Schwefel 2.26
F11 Hump	F29 Schwefel 1.2
F12 Matyas	F30 Schwefel 2.22
F13 Ackley	F31 Schwefel 2.21
F14 Alpine	F32 Sphere
F15 Brown	F33 Step
F16 Dixon and Price	F34 Sum function
F17 Fletcher-Powell	F35 Zakharov
F18 Griewank	F36 Wavy1

Πίνακας 1: 36 συναρτήσεις που δοκιμάστηκαν από την μέθοδο HSFA

Πηγή: Wang & L. Guo, (2013), σελ. 2

Λόγω της τυχαιότητας των αλγορίθμων, για την λήψη αντιπροσωπευτικών χαρακτηριστικών παρήγαμε 500 υλοποιήσεις της κάθε μεθόδου σε κάθε πρόβλημα. Οι πίνακες 2 και 3 απεικονίζουν το μέσο όρο και τα καλύτερα αποτελέσματα που βρέθηκαν από τον κάθε αλγόριθμο αντίστοιχα. Έχουν χρησιμοποιηθεί δύο διαφορετικές κλίμακες για την εξομάλυνση των τιμών των πινάκων. Η διάσταση της κάθε συνάρτησης έχει οριστεί σε 30. Από τον Πίνακα 5, κατά μέσο όρο, ο HSFA μπορεί εύκολα να βρίσκει τα ελάχιστα συναρτήσεων στις είκοσι οκτώ από τις τριάντα έξι συναρτήσεις. Ο FA βρίσκεται στην δεύτερη θέση με δέκα στις τριάντα έξι. Ο πίνακας 3 δείχνει ότι ο HSFA και ο FA εκτελούνται το ίδιο και καλύτερα, ο πρώτος σε είκοσι δύο από τις τριάντα έξι και ο δεύτερος σε δεκαεπτά συναρτήσεις αντίστοιχα. Οι ACO, DE, GA και βέλτιστης απόδοσης σε οκτώ. Από τους ανωτέρω πίνακες, μπορούμε να δούμε ότι, για τις συναρτήσεις χαμηλών διαστάσεων, τόσο ο FA όσο και ο HSFA εκτελούνται καλά, και η απόδοσή τους έχει μικρή διαφορά σε σχέση με κάθε άλλον. Περαιτέρω, γραφικές παραστάσεις σύγκλισης των δέκα μεθόδων για αρκετές αντιπροσωπευτικές συναρτήσεις απεικονίζονται στα σχήματα 1 έως 4 της Εικόνας 18 που δείχνουν τη διαδικασία βελτιστοποίησης. Οι τιμές εδώ είναι οι πραγματικές μέσες τιμές συναρτήσεων από τα παραπάνω πειράματα.

ΜΕΣΟΙ ΟΡΟΙ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

	ACO	BBO	DE	ES	FA	GA	HS	HSFA	PSO	SGA
F01	1.01	1.01	1.00	1.02	1.08	1.04	1.07	1.00	1.01	1.25
F02	1.43	2.71	1.00	2.55	1.00	1.39	16.66	1.00	3.13	1.22
F03	1.25	1.84	1.00	2.28	1.00	1.17	11.77	1.01	3.50	1.26
F04	3.9E4	1.7E5	49.66	2.8E5	1.00	4.0E4	2.0E6	1.5E3	3.7E5	2.5E5
F05	1.01	1.02	1.00	1.11	1.00	1.01	1.15	1.00	1.05	1.19
F06	1.03	1.02	1.00	1.09	1.00	1.02	1.03	1.00	1.03	3.01
F07	2.40	2.48	2.27	2.35	2.23	1.83	1.88	1.00	1.71	2.99
F08	1.72	1.72	1.72	1.72	1.72	1.72	1.72	1.72	1.72	1.00
F09	1.03	1.01	1.00	2.05	17.29	1.00	6.55	1.00	1.04	1.24
F10	2.40	2.40	2.40	3.06	2.40	2.40	3.09	2.40	2.70	1.00
F11	1.00	1.00	1.00	1.03	1.00	1.00	1.03	1.00	1.02	1.25
F12	1.00	1.00	1.00	1.01	1.00	1.00	1.02	1.00	1.00	1.14
F13	4.32	2.56	3.68	5.56	1.39	4.98	5.70	1.00	4.83	2.63
F14	36.17	7.98	43.16	73.74	11.68	33.13	70.32	1.00	53.62	8.48
F15	570.98	14.02	27.90	1.1E3	141.86	99.02	652.65	1.00	485.92	12.10
F16	1.6E3	75.20	317.08	1.2E4	7.35	942.08	1.1E4	1.00	1.6E3	26.84
F17	21.98	2.35	7.67	21.63	5.30	7.33	19.01	1.00	15.46	2.26
F18	8.49	5.40	14.18	66.70	2.31	28.49	139.02	1.00	52.77	5.69
F19	2.8E3	167.15	544.18	1.9E4	25.71	1.3E3	1.9E4	1.00	2.7E3	39.88
F20	93.79	13.59	68.16	276.60	20.05	92.42	282.65	1.00	173.17	9.33
F21	3.20	2.49	1.74	1.00	3.69	2.65	3.88	1.78	2.55	2.42
F22	1.2E8	9.7E3	2.8E5	5.1E7	6.64	5.8E5	7.8E7	1.00	7.9E6	9.81
F23	2.2E7	2.9E4	3.1E5	1.4E7	7.36	6.7E5	2.2E7	1.00	3.5E6	5.0E3
F24	112.59	8.00	48.08	188.98	1.04	25.76	133.99	1.00	52.91	2.92
F25	1.2E3	103.34	637.38	1.8E4	17.91	1.4E3	1.8E4	1.00	4.1E3	62.77
F26	24.37	4.58	21.06	32.51	7.75	20.84	29.89	1.00	23.03	7.29
F27	37.23	2.38	5.34	49.70	1.00	10.38	34.12	1.04	12.06	2.00
F28	37.76	18.45	73.51	92.92	93.82	31.93	109.20	1.00	112.28	21.21
F29	4.79	2.52	6.72	7.41	1.00	5.40	7.13	1.93	4.81	4.31
F30	42.08	6.33	16.76	63.65	9.36	30.16	53.57	1.00	35.27	8.32
F31	2.98	3.13	3.87	4.56	1.00	3.93	4.78	1.15	3.97	2.79
F32	205.80	13.56	37.41	382.80	1.87	131.27	361.49	1.00	151.62	14.65
F33	40.44	20.26	53.05	312.01	5.14	111.20	471.25	1.00	194.10	15.72
F34	274.21	27.02	46.57	546.26	6.17	138.85	550.25	1.00	188.96	25.75
F35	1.2E5	1.46	3.32	3.57	1.18	3.12	3.34	1.00	3.12	2.64
F36	9.82	5.26	14.12	29.95	10.67	16.90	35.57	1.00	23.95	5.37

End.

Πίνακας 2: Μέσοι όροι αποτελεσμάτων βελτιστοποίησης ACO, BBO, DE, ES, FA, GA, HS, FAHS, PSO και SGA

Πηγή: Wang & L. Guo, (2013), σελ. 6

ΚΑΛΥΤΕΡΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

	ACO	BBO	DE	ES	FA	GA	HS	HSFA	PSO	SGA
F01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F02	1.00	1.71	1.00	1.53	1.00	1.00	1.49	1.00	1.72	1.00
F03	1.00	1.28	1.00	1.12	1.00	1.00	1.28	1.00	1.34	1.13
F04	2.0E14	2.0E14	3.2E10	2.3E15	2.5E8	1.00	1.1E15	3.8E11	1.0E15	4.5E15
F05	1.00	1.00	1.00	1.02	1.00	1.00	1.00	1.00	1.00	1.00
F06	1.01	1.01	1.00	1.01	1.00	1.01	1.00	1.00	1.00	2.51
F07	2.5E6	3.3E6	7.2E5	1.6E6	1.00	1.7E4	2.1E5	2.88	3.7E5	3.3E6
F08	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.00
F09	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00
F10	2.65	2.65	2.65	2.74	2.65	2.65	2.65	2.65	2.65	1.00
F11	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.03
F12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
F13	8.34	3.93	7.05	11.02	1.00	8.71	11.56	1.54	9.51	3.98
F14	63.46	11.42	88.82	159.53	12.22	40.65	144.04	1.00	102.48	10.66
F15	218.30	9.54	47.47	591.91	33.50	128.64	792.73	1.00	358.64	9.44
F16	4.7E3	109.24	1.3E3	4.0E4	1.00	315.53	6.1E4	2.31	5.9E3	43.61
F17	42.58	3.24	11.83	41.06	1.20	9.50	43.49	1.00	29.07	3.02
F18	7.99	3.62	13.85	63.26	1.00	14.42	160.01	1.08	37.87	2.32
F19	3.1E3	135.16	893.82	3.9E4	1.56	566.65	5.4E4	1.00	6.1E3	3.678
F20	251.29	32.94	142.30	720.64	7.96	131.48	863.43	1.00	483.70	23.66
F21	3.96	2.89	1.65	1.00	4.60	2.91	4.87	1.90	2.72	2.37
F22	31.83	55.55	1.5E5	1.3E8	8.26	89.30	3.0E8	1.00	5.8E6	15.15
F23	1.00	4.4E3	1.5E6	8.8E7	27.10	3.3E4	1.6E8	4.89	2.8E7	29.223
F24	2.2E3	88.70	1.1E3	4.7E3	1.60	215.01	3.4E3	1.00	940.82	34.50
F25	3.0E3	380.11	2.9E3	1.0E5	1.00	3.1E3	1.3E5	2.01	2.4E4	54.677
F26	39.66	5.65	30.04	58.39	6.03	27.36	42.32	1.00	38.57	8.91
F27	54.77	2.13	12.53	87.36	1.27	10.85	58.14	1.00	21.11	2.77
F28	164.45	67.82	335.75	447.01	430.29	85.82	596.00	1.00	551.44	68.85
F29	8.78	4.00	16.50	15.31	1.00	10.85	18.42	3.27	6.75	7.29
F30	63.53	10.38	27.71	105.79	7.15	47.04	88.91	1.00	58.02	12.83
F31	3.80	5.63	7.23	9.39	1.00	7.31	10.20	1.93	7.56	4.53
F32	740.24	30.59	184.72	1.8E3	1.00	322.80	1.8E3	2.87	725.62	31.006
F33	149.29	66.43	224.57	1.4E3	3.86	255.71	2.4E3	1.00	1.0E3	42.71
F34	491.51	35.62	100.26	1.1E3	1.64	113.66	1.1E3	1.00	400.61	27.555
F35	3.44	2.50	5.89	6.67	1.00	4.28	5.48	1.46	3.83	3.74
F36	11.05	6.01	18.56	40.10	9.07	18.47	43.18	1.00	31.18	4.64

End.

Πίνακας 3: Τα καλύτερα αποτελέσματα βελτιστοποίησης

Πηγή: Wang & L. Guo, (2013), σελ. 7

Η F26 είναι μια πολύπλοκη πολυτροπική συνάρτηση και έχει μοναδικό ολικό βέλτιστο το 0 και αρκετά διαδοχικά τοπικά βέλτιστα. Το Σχήμα 1 δείχνει ότι ο HSFA συγκλίνει με ολική τιμή 0 με την γρηγορότερη ταχύτητα. Εδώ ο FA συγκλίνει στην αρχή λίγο πιο γρήγορα, αλλά είναι πιθανόν να παγιδευτεί σε επιμέρους ελάχιστα με αποτέλεσμα η τιμή της συνάρτησης να μειώνεται ελαφρώς.

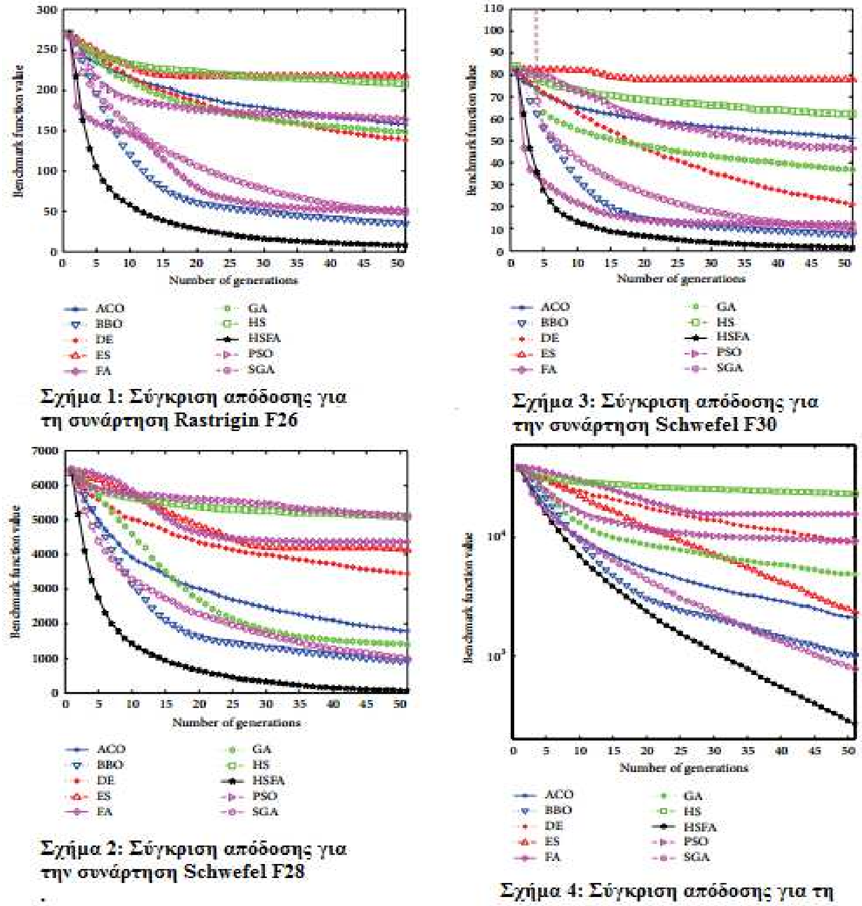
Η F28 είναι επίσης ένα πολυτροπικό πρόβλημα και έχει ένα ολικό βέλτιστο την τιμή 0. Για αυτό το πρόβλημα ο HSFA είναι ο καλύτερος από τους άλλους αλγόριθμους και βρίσκει την βέλτιστη λύση νωρίτερα. Σε αυτήν την συνάρτηση, το σχήμα δείχνει ότι ο HSFA ξεπερνά σημαντικά τους άλλους κατά την διαδικασία βελτιστοποίησης.

Τέλος ο HSFA συγκλίνει προς την καλύτερη λύση εξαίρετα σε σχέση με τους άλλους. Ο BBO είναι χειρότερος σε αυτήν την περίπτωση από τον HSFA αλλά έχει την δεύτερη καλύτερη επίδοση.

Στο Σχήμα 4 της Εικόνας 18 φαίνεται το αρχικό στάδιο της διαδικασίας βελτιστοποίησης όπου ο FA συγκλίνει ταχύτερα από τον HSFA ενώ ο HSFA είναι ικανός να βελτιώνει τις λύσεις του σταθερά μακροπρόθεσμα.

Αρχικά στις πρώτες 20 επαναλήψεις ο FA δείχνει να συγκλίνει ταχύτερα. Ωστόσο φαίνεται να παγιδεύεται σε επιμέρους ελάχιστα όσο η τιμή της συνάρτησης μειώνεται ελαφρά (μετά από 20 επαναλήψεις) και έχει πολύ καλύτερες επιδόσεις από τον HSFA μετά από 30 επαναλήψεις. Από τα σχήματα 1-4 οι επιδόσεις τους HSFA είναι κατά πολύ καλύτερες από των υπολοίπων. Γενικά ο BBO και ο FA, ειδικά ο FA, είναι κατώτεροι από τον HSFA. Έμμεσα αποδεικνύεται ότι ο HSFA είναι μια πιο αποτελεσματική μέθοδος βελτιστοποίησης σε σχέση με τις άλλες. (Wang & L. Guo,2013[24])

[24] σελ: 1 - 9



Σχήμα 1: Σύγκριση απόδοσης για τη συνάρτηση Rastrigin F26

Σχήμα 3: Σύγκριση απόδοσης για την συνάρτηση Schwefel F30

Σχήμα 2: Σύγκριση απόδοσης για την συνάρτηση Schwefel F28

Σχήμα 4: Σύγκριση απόδοσης για τη συνάρτηση F33

Εικόνα 18: Γραφικές παραστάσεις σύγκλισης των μεθόδων ACO, BBO, DE, ES, FA, GA, HS, FAHS, PSO και SGA

Πηγή: Wang & L. Guo, (2013), σελ. 5

ΚΕΦΑΛΑΙΟ 5. ΥΛΟΠΟΙΗΣΗ ΥΒΡΙΔΙΚΟΥ ΑΛΓΟΡΙΘΜΟΥ

Στο σημείο αυτό θα προβούμε στην υλοποίηση του υβριδικού αλγορίθμου Πυγολαμπίδας με χρήση αλγορίθμου Αναζήτησης Αρμονίας (Firefly Algorithm with Harmony Search ή HSFA). Στα προηγούμενα κεφάλαια αναφέραμε τα χαρακτηριστικά των δύο βασικών αλγορίθμων που θα χρησιμοποιήσουμε, δηλαδή του αλγορίθμου Αναζήτησης Αρμονίας (HS) και του αλγορίθμου Πυγολαμπίδας (FA). Ο υβριδικός αλγόριθμος που θα υλοποιηθεί θα καλείται **Υυβριδικός αλγόριθμος Πυγολαμπίδας με χρήση Αρμονικής Αναζήτησης (HSFA)**.

Η υλοποίηση έγινε στην προγραμματιστική γλώσσα Matlab. Επίσης θα πρέπει να αναφέρουμε ότι για να μην έχουμε υψηλή υπολογιστική πολυπλοκότητα και για να εξάγουμε απλά συμπεράσματα επιλέγουμε συναρτήσεις βελτιστοποίησης δυο διαστάσεων.

5.1 ΣΥΝΑΡΤΗΣΕΙΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ BENCHMARK

Για τα πειραματικά αποτελέσματα θα χρησιμοποιήσουμε benchmark συναρτήσεις που είναι διαδεδομένες για την εξαγωγή αποτελεσμάτων στην βελτιστοποίηση αλγορίθμων, οποίες δίνουν χρήσιμες πληροφορίες για την ταχύτητα σύγκλισης, την ακρίβεια, την ευρωστία και γενικά για την απόδοση των προς εξέταση αλγορίθμων.

Ας δούμε κάποιες από τις benchmark συναρτήσεις που θα χρησιμοποιήσουμε:

- Η συνάρτηση ***Ackley's*** έχει τον εξής τύπο:

  $f(x,y) = -20 \times e\left(-20 \times \sqrt{0.5 \times (x^2+y^2)}\right) - e \times (0.5(2 \times \cos(2\pi \times x) + \left(0.5(2 \times \cos(2\pi \times y))\right) + 20 + e,$

  ορίζεται στο διάστημα -5≤ x,y≤5 και παρουσιάζει ολικό ελάχιστο στο x*={0,0}, με f(0,0)=0.

- Η συνάρτηση ***Beal's*** έχει τον εξής τύπο:

  $f(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2,$

ορίζεται στο διάστημα -4.5≤ x,y≤4.5 και παρουσιάζει ολικό ελάχιστο στο x*={3,0.5}, με f(3,0.5)=0.

- Η συνάρτηση ***Goldstein Price*** έχει τον εξής τύπο:

$$f(x,y) = \left(1 + (x + y + 1)^2 \times (19 - 14x + 3x^2 - 14y + 6xy + 3y^2)\right) \times (30 + (2x - 3y)^2 \times (18 - 32x + 12x^2 + 48y - 36xy + 27y^2)),$$

ορίζεται στο διάστημα -2≤x,y≤2 και παρουσιάζει ολικό ελάχιστο στο x*={0,-1}, με f(0,-1)=3.

- Η συνάρτηση ***Booth's*** έχει τύπο:

$$f(x,y) = (x + 2y - 7)^2 + (2x + y - 5)^2,$$

ορίζεται στο διάστημα -10≤x,y≤10 και παρουσιάζει ολικό ελάχιστο στο x*={1,3}, με f(1,3)=0.

- Η συνάρτηση ***Matya's*** με τύπο:

$$f(x,y) = 0.26 \times (x^2 + y^2) - 0.48xy,$$

ορίζεται στο διάστημα -10≤x,y≤10 και παρουσιάζει ολικό ελάχιστο στο x*={0,0}, με f(0,0)=0.

5.2 ΥΛΟΠΟΙΗΣΗ ΣΕ MATLAB

Στο σημείο αυτό θα παρουσιάσουμε το κύριο θέμα της εργασίας που είναι ο υβριδικός αλγόριθμος HSFA. Παρακάτω παραθέτονται τα κομμάτια–αρχεία που τον απαρτίζουν. Όπως προαναφέραμε η υλοποίηση έγινε σε γλώσσα Matlab.

*Αρχείο **HS.m**. Ο αλγόριθμος της Αρμονικής Αναζήτησης*

```
function [best] = HS
   [HMS, MAXIT, HMCR, PAR, bw, ~, ~, ~, ~] = get_variables();
   Minimize = should_minimize();
   [MinBounds, MaxBounds, f, D] = get_objective_function();

   plot_3d_function(f, MinBounds, MaxBounds);
```

```
    [HM, HF, HMN] = initialize(MinBounds, MaxBounds, HMS, D, f);
    [WorstFit, WorstLoc] = findWorstLoc(Minimize, HF);

    for iteration = 1:MAXIT
        if rand() < HMCR
            for i = 1 : D
                HMN(i) = randi([1, HMS], 1);
                if rand() < PAR
                    HMN(i) = HMN(i) + bw .* (2 .* rand - 1);
                end
            end
        else
            HMN = MinBounds + (MaxBounds - MinBounds) .* rand(1, D);
        end
        XW = f(HMN(1), HMN(2));
        if XW < WorstFit
            HM(WorstLoc,:) = HMN;
            HF(WorstLoc) = XW;
            [WorstFit, WorstLoc] = findWorstLoc(Minimize, HF);
        end
    end
    printResults(sort_results(HM, HF, Minimize), HMS);
end

function [HM, HF, HMN] = initialize(MinBounds, MaxBounds, HMS, D, f)
    HM = zeros(HMS, D);
    for i = 1:HMS
        HM(i, :) = MinBounds + (MaxBounds - MinBounds) .* rand(1, D);
    end
    HF = f(HM(:, 1), HM(:, 2));
    HMN = zeros(2,1);
end

function [WorstFit, WorstLoc] = findWorstLoc(Minimize, HF)
    if Minimize == 1
        [WorstFit, WorstLoc] = max(HF);
    else
        [WorstFit, WorstLoc] = min(HF);
    end
end

function [best] = sort_results(HM, HF, Minimize)
    x = HM;
    y = HF;

    if Minimize == 1
        [y, index] = sort(y);
    elseif Minimize == 0
        [y, index] = sort(y, 'descend');
    end
    x = x(index, :);

    best = x;
    best(:, 3) = y;
end
```

*Αρχείο **FA.m**. Ο αλγόριθμος της Πυγολαμπίδας*

```
function [best] = FA(n, maxGeneration)
   [~, ~, ~, ~, ~, alpha, gamma, beta0, delta] = get_variables();
   Minimize = should_minimize();
   [MinBounds, MaxBounds, f] = get_objective_function();

   [ffLoc, I] = initialize_fa_population(f, MinBounds, MaxBounds, n);
   [x, y, z] = plot_3d_function(f, MinBounds, MaxBounds);

   initialize_fireflies_plot();
   for step = 1 : maxGeneration
      plot_fireflies(I, ffLoc, Minimize, x, y, z);
      ffLocTmp = ffLoc;
      for i = 1 : n
         for j = 1 : n
            if (Minimize == 1 && I(j) < I(i)) ...
                  || (Minimize == 0 && I(j) > I(i))
               ffLoc(i, :) = move_i_towards_j(ffLoc, ffLocTmp, i, j, ...
                  alpha, beta0, gamma, MinBounds, MaxBounds);
            end
         end
      end
      drawnow;
      hold off;
      [I, ffLoc] = update_ligh_intensity(f, ffLoc, Minimize);
      alpha = new_alpha(alpha,delta);
   end

    best = ffLoc;
    best(:,3) = I;
    printResults(best, n);
end
```

➢

*Αρχείο **check_bounds.m**.*

Για ένα συγκεκριμένο σημείο (x,y) η συνάρτηση ελέγχει αν κάθε ένα από τα x,y είναι μέσα στο εύρος τιμών της συνάρτησης. Αν δεν είναι, τότε το θέτει μέσα στα όρια χρησιμοποιώντας το μέγιστο ή ελάχιστον όριο αντίστοιχα με το αν είναι μεγαλύτερο ή μικρότερο.

```
function [newLoc] = check_bounds(loc, MinBounds, MaxBounds)
   newLoc = loc;
   for index = 1 : length(loc)
      if loc(index) < MinBounds(index), newLoc(index) = MinBounds(index); end
      if loc(index) > MaxBounds(index), newLoc(index) = MaxBounds(index); end
   end
end
```

*Αρχείο **get_objective_function.m**.*

Στο αρχείο αυτό μπορούμε να ορίσουμε ποια αντικειμενική συνάρτηση θέλουμε να χρησιμοποιήσουμε για τα πειράματα. Η τιμή **D** είναι οι διαστάσεις του προβλήματος. Μιας και έχουμε προβλήματα με δυο μεταβλητές το έχουμε ορίσει σταθερά ίσο με 2. Μέσα σε αυτό το αρχείο έχουμε την αντικειμενική συνάρτηση function (get_objective_function) που επιστρέφει την συνάρτηση που θέλουμε να χρησιμοποιήσουμε και κάποιες άλλες που ορίζουν στην ουσία μια συνάρτηση η καθεμία τους με το εύρος τιμών της. Αν θέλουμε μπορούμε να προσθέσουμε οποιαδήποτε συνάρτηση και να της δώσουμε το κατάλληλο όνομα και τιμές. Για να ορίσουμε τώρα ποια συνάρτηση θα χρησιμοποιείται, την ορίζουμε στην get_objective_function εκεί που αυτή την στιγμή βρίσκεται η get_ackleys(). Αν θέλουμε να χρησιμοποιήσουμε κάποια άλλη απλά αντικαθιστάμε το get_ackleys() με πχ το get_booths().

```
function [MinBounds, MaxBounds, f, D] = get_objective_function
   % D is the problem dimension - Currently only 2D problems are supported
   D = 2; % Problem dimension
   [MinBounds, MaxBounds, strFunction] = get_ackleys();
   f = vectorize(inline(strFunction));
end

function [MinBounds, MaxBounds, strFunction] = get_ackleys
   str1 = '-20 * exp(-0.2*sqrt(0.5*(x^2 + y^2)))';
   str2 = '- exp(0.5*(cos(2*pi*x) + cos(2*pi*y))) + 20 + exp(1)';
   strFunction = strcat(str1,str2);
   MinBounds   = [-5 -5];
   MaxBounds   = [5 5];
end

function [MinBounds, MaxBounds, strFunction] = get_beales
   str1 = '(1.5 - x + x * y)^2 + (2.25 - x + x * y^2)^2';
   str2 = '+ (2.625  - x + x * y^3)^2';
   strFunction = strcat(str1,str2);
   MinBounds   = [-4.5 -4.5];
   MaxBounds   = [4.5 4.5];
end

function [MinBounds, MaxBounds, strFunction] = get_goldstein_price
   str1 = '(1 + (x + y + 1)^2 * (19 - 14*x + 3*x^2 - 14*y + 6*x*y + 3*y^2))';
   str2 = '* (30 + (2*x - 3*y)^2 * (18 - 32*x  + 12*x^2 + 48*y - 36*x*y + 27*y^2))';
   strFunction = strcat(str1,str2);
   MinBounds   = [-2 -2];
   MaxBounds   = [2 2];
end

function [MinBounds, MaxBounds, strFunction] = get_booths
   str1 = '(x + 2*y - 7)^2 + (2*x + y - 5)^2';
   str2 = '';
   strFunction = strcat(str1,str2);
   MinBounds   = [-10 -10];
   MaxBounds   = [10 10];
end
```

```
function [MinBounds, MaxBounds, strFunction] = get_matyas
   str1 = '0.26 * (x^2 + y^2) - 0.48*x*y';
   str2 = '';
   strFunction = strcat(str1,str2);
   MinBounds   = [-10 -10];
   MaxBounds   = [10 10];
end
```

*Αρχείο **get_variables.m**.*

Εδώ ορίζονται κάποιες τιμές για τους αλγορίθμους που έχουν υλοποιηθεί. Μπορούμε να αλλάζουμε από εδώ τις τιμές και να τρέχουμε όλους του αλγορίθμους με τις ίδιες τιμές.

```
function [HMS, MAXIT, HMCR, PAR, BW, ALPHA, GAMMA, BETA0, DELTA] = get_variables
   %% HS VARIABLES
   HMS   =   10    ; % Harmony memory size
   MAXIT = 10000    ; % Maximum iterations
   HMCR  =    0.95 ; % Memory considetation
   PAR   =    0.3  ; % Pitch adjustment rate
   BW    =    0.2  ; % The pitch adjustment bandwidth

   %% FA VARIABLES
   ALPHA =    0.2  ; % Randomness 0--1 (highly random)
   GAMMA =     1.0  ; % Absorption coefficient
   BETA0 =    1    ; % maximum attractiveness
   DELTA =    0.97 ; % Randomness reduction (similar to an annealing schedule)
end
```

Αρχείο initialize_fa_population.m

Αποτελεί μια τυπική αρχικοποίηση του πληθυσμού των πυγολαμπίδων. Δεν έχει τίποτα ιδιαίτερο που να επηρεάζει τα πειράματα.

```
function [ffLoc, I] = initialize_fa_population(f, MinBounds, MaxBounds, n)
   ffLoc = [rand(n,1) * (MaxBounds(1) - MinBounds(1)) + MinBounds(1), ...
      rand(n,1) * (MaxBounds(2) - MinBounds(2)) + MinBounds(2)];
   I = f(ffLoc(:,1), ffLoc(:,2));
end
```

*Αρχείο **initialize_fireflies_plot.m**.*

Αρχικοποιεί το διάγραμμα των πυγολαμπίδων, δηλαδή καθορίζει τις αρχικές θέσεις αυτών.

```
function [ffp] = initialize_fireflies_plot()
   ffp = figure(2);
```

```
    set(ffp, 'name', 'Fireflies positions');
    pos = get(ffp,'Position');
    pos(1) = pos(1) + pos(3) + 50;
    set(ffp, 'Position', pos);
end
```

*Αρχείο **move_i_towards_j.m**.*

Μετακινεί την πυγολαμπίδα i προς την φωτεινότερη j.

```
function [newLoc] = move_i_towards_j(aloc, loc, i, j, a, b, g, ...
    MinBounds, MaxBounds)
    r = sqrt(sum((aloc(i,:) - loc(j,:)).^2));
    newLoc = aloc(i,:) + b .* exp(-g .* r .^ 2) .* ...
        (loc(j,:) - aloc(i,:)) + a .* (rand() - 0.5);
    newLoc = check_bounds(newLoc, MinBounds, MaxBounds);
end
```

*Αρχείο **new_alpha.m**.*

Δίνει μια νέα τιμή στο alpha σε κάθε γύρο αλλάζοντάς το σύμφωνα με τον συντελεστή στο get_variables.m.

```
function [a] = new_alpha(alpha, delta)
    a = alpha * delta;
end
```

*Αρχείο **plot_3d_function.m**.*

"Ζωγραφίζει" το 3D διάγραμμα που απεικονίζει την αντικειμενική συνάρτηση.

```
function [x, y, z] = plot_3d_function(f, MinBounds, MaxBounds)
    step = 0.1;
    [x, y] = meshgrid(MinBounds(1) : step : MaxBounds(1), ...
        MinBounds(2) : step : MaxBounds(2));
    z = f(x, y);
    fp = figure(1);
    set(fp, 'name', '3D plot of the function');
    surfc(x, y, z);
end
```

*Αρχείο **plot_fireflies.m**.*

"Ζωγραφίζει τις θέσεις των πυγολαμπίδων πάνω στην κάτοψη του διαγράμματος της συνάρτησης.

```
function plot_fireflies(I, ffLoc, Minimize, x, y, z)
   contour(x, y, z, 15);
   hold on;
   max_I = max(I);
   for i = 1 : length(I)
      tmp = I(i) / max_I;
      if Minimize == 1, tmp = 1- tmp; end
      color = [tmp, abs(1-tmp), 0];
      plot(ffLoc(i, 1), ffLoc(i, 2), ...
         '.', ...
         'markersize', 20, ...
         'markerfacecolor', color, ...
         'MarkerEdgeColor', color);
   end
end
```

*Αρχείο **print_results.m**.*

Εκτυπώνει τα αποτελέσματα των αλγορίθμων.

```
function printResults(results, n)
   printmat(results, 'Results', num2str(linspace(1, n, n)), 'x y f(x,y)');
end
```

*Αρχείο **should_minimize.m**.*

Εδώ ορίζουμε το αν θέλουμε να γίνει μεγιστοποίηση ή ελαχιστοποίηση της αντικειμενικής συνάντησης. Για ελαχιστοποίηση δίνεται η τιμή στο minin ίση 1 ενώ για μεγιστοποίηση 0.

```
function [minim] = should_minimize
   % If we want to perform a minimization procedure then set minim to 0
   % If we want to perform a maximization procedure then set minim to 1

   % Min <- minim = 1
   % Max <- minim = 0
   minim = 1;
end
```

*Αρχειο **update_light_intensity.m**.*

Ενημερώνει την φωτεινότητα των πυγολαμπίδων.

```
function [I, ffLoc] = update_ligh_intensity(f, ffLoc, Minimize)
```

```
    I = f(ffLoc(:,1), ffLoc(:,2));
    if Minimize == 1
        [I, index] = sort(I);
    elseif Minimize == 0
        [I, index] = sort(I, 'descend');
    end
    ffLoc = ffLoc(index, :);
end
```

*Αρχείο **HSFA.m**.*

Ο υβριδικός αλγόριθμος πυγολαμπίδας με χρήση αναζήτησης Αρμονίας.

```
function [best] = HSFA(n, keep, maxGeneration)
    [~, ~, HMCR, PAR, bw, alpha, gamma, beta0, delta] = get_variables();
    Minimize = should_minimize();
    [MinBounds, MaxBounds, f, D] = get_objective_function();

    [ffLoc, I] = initialize_fa_population(f, MinBounds, MaxBounds, n);
    [x, y, z] = plot_3d_function(f, MinBounds, MaxBounds);

    initialize_fireflies_plot();
    for step = 1 : maxGeneration
        [ffLoc, I] = sort_fireflies(ffLoc, I, Minimize);
        plot_fireflies(I, ffLoc, Minimize, x, y, z);
        ffLocTmp = ffLoc;
        for i = 1 : keep
            for j = 1 : n
                if (Minimize == 1 && I(j) < I(i)) ...
                        || (Minimize == 0 && I(j) > I(i))
                    ffLoc(j, :) = move_i_towards_j(ffLoc, ffLocTmp, j, ...
                        i, alpha, beta0, gamma, MinBounds, MaxBounds);
                else
                    ffLoc(j, :) = mutate(ffLoc, HMCR, PAR, bw, D, n, ...
                        MinBounds, MaxBounds);
                end
                I(j) = f(ffLoc(j,1), ffLoc(j,2));
            end
        end
        drawnow;
        hold off;
        [I, ffLoc] = update_ligh_intensity(f, ffLoc, Minimize);
        alpha = new_alpha(alpha,delta);
    end

    best = ffLoc;
    best(:,3) = I;
    printResults(best, n);
end

function [ffLoc, I] = sort_fireflies(ffLoc, I, Minimize)
    if Minimize == 1
        [I, index] = sort(I);
    elseif Minimize == 0
        [I, index] = sort(I, 'descend');
```

```
    end
    ffLoc = ffLoc(index, :);
end

function [loc] = mutate(ffLoc, HMCR, PAR, bw, D, n, MinBounds, MaxBounds)
    loc = rand(1, D);
    for k = 1 : D
        if rand < HMCR
            r = randi([1, n]);
            loc(1, k) = ffLoc(r, k);
            if rand() < PAR
                loc(1, k) = loc(1, k) + bw .* (2 .* rand - 1);
            end
        else
            loc(1, k) = MinBounds(k) + rand * (MaxBounds(k) - MinBounds(k));
        end
    end
    loc = check_bounds(loc, MinBounds, MaxBounds);
end
```

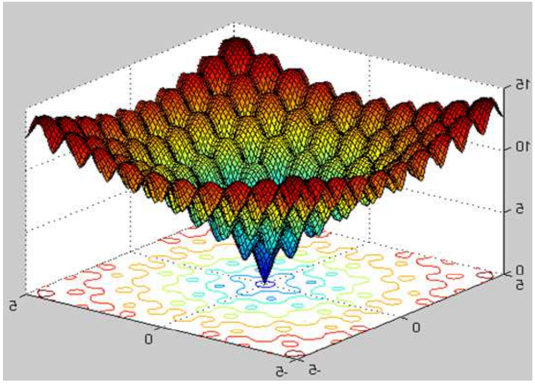
5.3 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Στην προηγούμενη ενότητα είδαμε τον κώδικα του υβριδικού αλγορίθμου και των αλγορίθμων που τον απαρτίζουν καθώς και τις υπολειτουργίες του στην προγραμματιστική γλώσσα MATLAB. Στο κομμάτι αυτό θα δούμε κάποια αριθμητικά αποτελέσματα για τις benchmark συναρτήσεις του 5.1.

Θα πρέπει να αναφέρουμε ότι τα αποτελέσματα προέκυψαν ύστερα από αρκετές δοκιμές και με συνεχή τροποποίηση των παραμετρικών τιμών των δυο αλγορίθμων. Τα αποτελέσματα ήταν ικανοποιητικά στις περισσότερες περιπτώσεις, ωστόσο οι παράμετροι της τυχαιότητας καθώς και η παγίδευση των αλγορίθμων σε τοπικά βέλτιστα έδωσαν έναν υψηλό δείκτη δυσκολίας στην επίλυση.

Οι παράμετροι των αλγορίθμων έμειναν σταθεροί ή αυξομειώθηκαν ελαφρώς. Στον αλγόριθμο Αναζήτησης Αρμονίας το μέγεθος της Αρμονικής Μνήμης ορίστηκε ίσο με 10, το HMCR με 0.95, το PAR με 0.3 και το BW με 0.2 .Ενώ στον Αλγόριθμο Πυγολαμπίδας η παράμετρος α αυξομειώθηκε ελαφρώς από 0.2 έως 0.8 , η γ από 0.7 έως 1, η β_0 από 0.8 έως 1 και η δ από 0.96 έως 0.98. Ας δούμε τα πειραματικά αποτελέσματα των συναρτήσεων.

1. Για την συνάρτηση **Ackley 's**, η γραφική παράσταση της απεικονίζεται στην Εικόνα 19 και μας δείχνει το ολικό βέλτιστο που είναι το ελάχιστο στο σημείο (0,0) στο διάστημα [-5,5].



Εικόνα 19: Γραφική Παράσταση συνάρτησης Ackley 's

Οι τιμές που δίνει ο αλγόριθμος Αναζήτησης Αρμονίας απεικονίζονται στον Πίνακα 4:

Results =

	x	y	f(x,y)
1	-4.76143	4.69613	14.07276
2	-4.83929	4.46976	14.04159
3	4.29009	-3.80611	12.78141
4	-4.74203	1.46073	12.20309
5	3.86130	4.20304	12.20020
6	3.27129	-3.62082	12.03682
7	-0.33127	-4.75218	11.73323
8	-0.21796	-4.53542	11.51642
9	2.38025	-1.22941	8.28512
10	0.04428	0.07020	0.40975

Πίνακας 4: Τιμές που δίνει ο HS για την συνάρτηση Ackley 's

Παρατηρούμε ότι συγκλίνει κοντά στην βέλτιστη λύση σε 10000 επαναλήψεις και δίνει μια σχετικά καλή προσέγγιση πολύ κοντά στην πραγματική λύση, με x=0.04428 ≈ 0 και y=0.07020 ≈ 0, δηλαδή f(0.04428,0.0720)=0.40975 ≈ 0.

Ο αλγόριθμος πυγολαμπίδας δίνει τα αποτελέσματα του Πίνακα 5 για 10 πυγολαμπίδες σε εξέλιξη 50 γενεών.

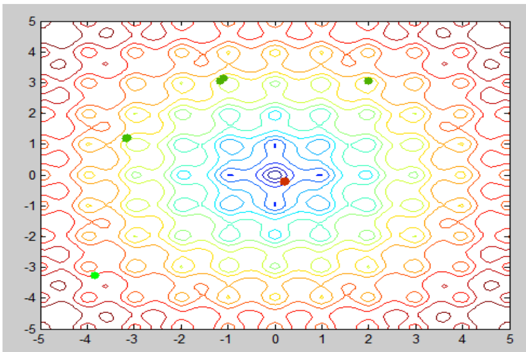
```
>> FA(10,50);

Results =
```

	x	y	f(x,y)
1	0.06283	0.85549	2.84507
2	0.05222	0.84487	2.84919
3	0.04390	0.83656	2.85717
4	-0.02633	-1.90400	4.96810
5	-0.05587	-1.94002	4.97312
6	-0.02362	-1.89751	4.98000
7	-0.05769	-1.93504	4.98277
8	-0.07322	-1.95052	5.02316
9	0.00780	-1.87205	5.03923
10	4.69957	3.15603	12.60655

Πίνακας 5: Τιμές που δίνει ο FA για την συνάρτηση Ackley ‘s

Αν και η αναζήτηση του σημείου δίνει ένα καλό αποτέλεσμα κοντά στο σημείο (0,0) από την αρχή, η τιμή της συνάρτησης δεν είναι ικανοποιητική, δηλαδή ίση με 0. Θα πρέπει να αναφέρουμε ότι η ρύθμιση των παραμέτρων παίζει ένα σημαντικό ρόλο στα αποτελέσματα, κυρίως λόγω της τυχαιότητας. Στην Εικόνα 20 παρουσιάζεται η τελική θέση των πυγολαμπίδων στο διάστημα [-5,5].



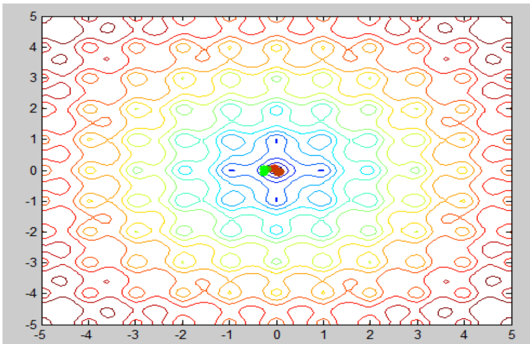
Εικόνα 20: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Ackley ‘s του Αλγορίθμου Πυγολαμπίδας

Στο υβριδικό αλγόριθμο HSFA τα αποτελέσματα είναι ξεκάθαρα από την πρώτη κιόλας επανάληψη, όπως βλέπουμε στον Πίνακα 6. Χρησιμοποιήσαμε 10 πυγολαμπίδες από τις οποίες οι 5 είναι οι «κορυφαίες πυγολαμπίδες», όπως τις είδαμε στο κεφάλαιο 4, σε εξέλιξη 50 γενεών.

Results =	x	y	f(x,y)
1	-0.04463	0.04192	0.27058
2	-0.04260	0.04424	0.27170
3	-0.04437	0.04247	0.27170
4	-0.04492	0.04192	0.27183
5	-0.04465	0.04223	0.27193
6	-0.03917	0.04767	0.27334
7	-0.03743	0.04921	0.27409
8	-0.03659	0.05004	0.27506
9	-0.04463	0.04767	0.29508
10	-0.04615	0.10637	0.65296

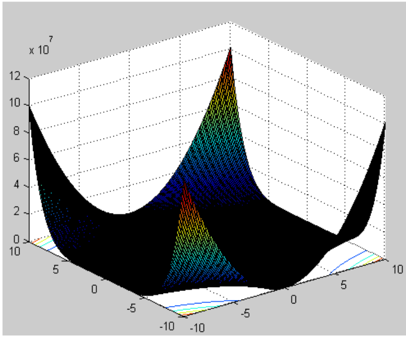
Πίνακας 6: Τιμές που δίνει ο HSFA για την συνάρτηση Ackley ‘s

Παρατηρούμε ότι από τις αρχικές επαναλήψεις το ολικό βέλτιστο προσεγγίζεται ικανοποιητικά και αποτελεσματικά. Σίγουρα ο υβριδικός αλγόριθμος είναι ο πιο αποτελεσματικός από τους τρεις σε όλους τους τομείς. Οι τελικές θέσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 21.



Εικόνα 21: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Ackley ‘s με την χρήση του υβριδικού Αλγορίθμου HSFA

2. Η συνάρτηση **Beale’s**, έχει την γραφική παράσταση που απεικονίζεται στην Εικόνα 22 και παρουσιάζει ακρότατο, ολικό ελάχιστο, στο σημείο (3,0.5) με f(3,0.5)=0 στο διάστημα $[-10,10]$.



Εικόνα 22: Γραφική Παράσταση συνάρτησης Beale ‘s

Ο αλγόριθμος Αναζήτησης Αρμονίας δίνει τα αποτελέσματα που φαίνονται στον Πίνακα 7 για 10000 επαναλήψεις.

Results =

	x	y	f(x,y)
1	2.92358	0.47854	0.00109
2	3.94225	0.64993	0.07033
3	5.94794	0.80575	0.19025
4	6.95659	0.82714	0.24723
5	7.80633	0.85626	0.24947
6	8.00000	0.86101	0.25541
7	8.00000	0.86175	0.25691
8	8.10398	0.86012	0.25809
9	9.00000	0.88144	0.29102
10	10.00000	0.89277	0.29871

Πίνακας 7: Τιμές που δίνει ο HS για την συνάρτηση Beale ‘s

Παρόμοια με πριν, το ολικό βέλτιστο προσεγγίζεται ικανοποιητικά με τις ίδιες παραμέτρους. Το ολικό ελάχιστο της συνάρτησης είναι το (3,0.5) ≈ (2.92358 , 0.47854). Το αποτέλεσμα της συνάρτησης βελτιστοποίησης είναι f(2.92358 , 0.47854) = 0.00109 ≈ 0.

Ο αλγόριθμος αναζήτησης πυγολαμπίδας χρησιμοποιεί 10 πυγολαμπίδες σε εξέλιξη 100 γενεών. Τα αποτελέσματα του φαίνονται στον Πίνακα 8.

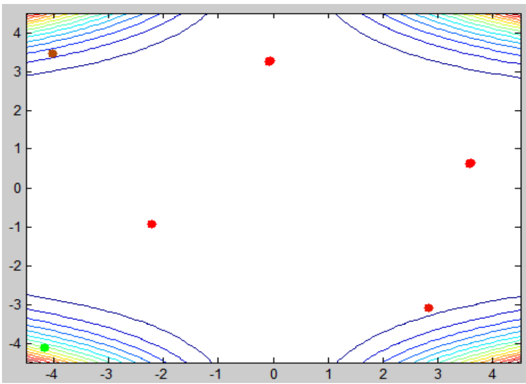
```
>> FA(10,100);

Results =
```

	x	y	f(x,y)
1	2.70044	0.42360	0.02121
2	2.70250	0.42566	0.02124
3	2.70414	0.42731	0.02132
4	2.70972	0.43288	0.02191
5	-1.69777	1.40649	1.16387
6	-1.69660	1.40767	1.16410
7	-1.69587	1.40840	1.16436
8	-1.68490	1.41937	1.17971
9	0.22859	3.89622	288.18178
10	0.89185	-3.63637	1873.31126

Πίνακας 8: Τιμές που δίνει ο FA για την συνάρτηση Beale 's

Από τις πρώτες κιόλας επαναλήψεις ο αλγόριθμος δείχνει να προσεγγίζει το ολικό ελάχιστο ικανοποιητικά με x=2.70044 ≈ 3 και y=0.42360≈ 0.5. Δείχνει ότι έχει αρκετά γρήγορη σύγκλιση σε σχέση με τον αλγόριθμο Αναζήτησης Αρμονίας. Η τελική θέση των πυγολαμπίδων απεικονίζονται στην Εικόνα 23.



Εικόνα 23: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Beale 's του Αλγορίθμου Πυγολαμπίδας

Ο υβριδικός αλγόριθμος HSFA υπερτερεί σε σχέση με τους υπόλοιπους μεμονωμένους αλγορίθμους, καθώς έχει πολύ πιο γρήγορη σύγκλιση στο ολικό ελάχιστο με μεγαλύτερη ακρίβεια εύρεσης λύσης και με καλύτερη προσέγγιση της τιμής της λύσης. Πρέπει να αναφέρουμε ότι χρησιμοποιήθηκαν 10 πυγολαμπίδες, με 2 από αυτές να είναι οι «κορυφαίες» σε 100 γενεές. Τα αποτελέσματα φαίνονται στον Πίνακα 9.

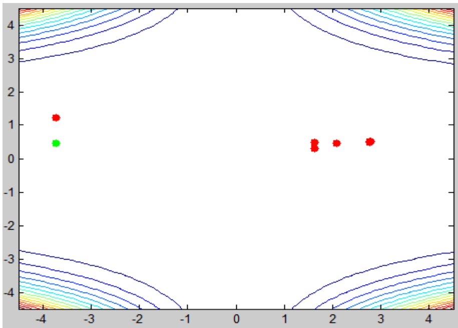
```
>> HSFA(10,2,100);

Results =
```

	x	y	f(x,y)
1	2.86380	0.46574	0.00351
2	2.86928	0.46154	0.00352
3	2.86715	0.46908	0.00358
4	2.86008	0.46201	0.00370
5	2.85868	0.46154	0.00379
6	2.85847	0.46041	0.00388
7	2.84914	0.45422	0.00499
8	2.88025	0.48219	0.00621
9	2.81223	0.46821	0.01325
10	3.00441	0.45107	0.05349

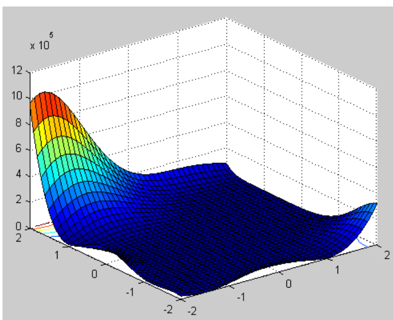
Πίνακας 9: Τιμές που δίνει ο HSFA για την συνάρτηση Beale 's

Αν παρατηρήσουμε τις τιμές καταλαβαίνουμε ότι για x=2.86380 ≈ 3 και y=0.46574 ≈ 0.5 η συνάρτηση f(2.86380, 0.46574)=0.00351 ≈ 0, άρα η τιμή της προσεγγίζεται ικανοποιητικά τις τιμές των άλλων αλγορίθμων. Οι τελικές θέσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 24.



Εικόνα 24: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Beale 's με την χρήση του υβριδικού Αλγορίθμου HSFA

3. Η συνάρτηση Goldstein's price έχει την γραφική παράσταση της Εικόνας 25.



Εικόνα 25: Γραφική Παράσταση συνάρτησης Goldstein 's price

Για ακόμη μια φορά ο αλγόριθμος Αναζήτησης Αρμονίας αποτελεί μια πειστική μέθοδο εύρεσης της βέλτιστης τιμής, ωστόσο με αρκετά αργή σύγκλιση. Οι τιμές που παράγει ο αλγόριθμος για 10000 επαναλήψεις απεικονίζονται στον Πίνακα 10.

```
>> HS();

Results =
```

	x	y	f(x,y)
1	-1.20193	1.88080	699889.58972
2	1.99405	-1.30052	156275.04998
3	0.31961	1.35626	38255.85967
4	-0.79179	-1.94568	25729.95496
5	0.25634	1.12915	24068.61469
6	1.14431	0.44534	1049.56970
7	1.42488	0.33896	743.34226
8	0.71184	-0.36183	259.49441
9	0.63756	-0.68985	152.42594
10	-0.04020	-1.00854	3.37762

Πίνακας 10: Τιμές που δίνει ο HS για την συνάρτηση Goldstein 's price

Παρατηρούμε ότι προσεγγίζει ικανοποιητικά το ολικό ελάχιστο της συνάρτηση με x=-0.04020 ≈0 και y=-1.00854 ≈-1 και f(-0.04020, -1.00854)=3.37762 ≈ 3.

Ο αλγόριθμος πυγολαμπίδας από την άλλη μας δείχνει από την αρχή τις δυνατότητες του καθώς από την πρώτη μέχρι την τελευταία εκτέλεση του, οι προσεγγιστικές τιμές που παράγει είναι η μια καλύτερη από την άλλη. Τα αποτελέσματα που παράχθηκαν από την χρήση 10 πυγολαμπίδων σε εξέλιξη 100 γενεών, φαίνονται στον Πίνακα 11:

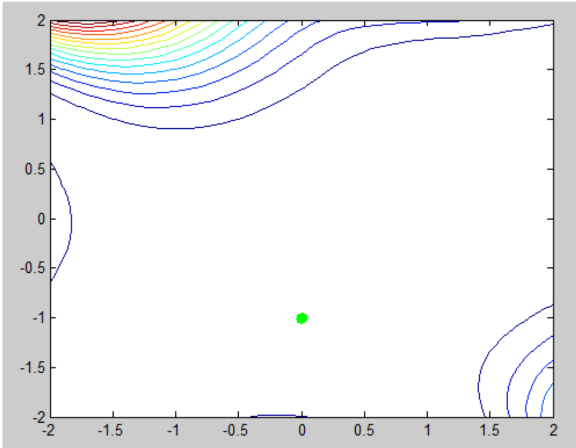
```
>> FA(10,100);

Results =
```

	x	y	f(x,y)
1	-0.00895	-0.99594	3.03466
2	-0.00925	-0.99623	3.03471
3	-0.00986	-0.99685	3.03508
4	-0.00534	-0.99233	3.04070
5	-0.01351	-1.00050	3.04466
6	-0.01376	-1.00075	3.04577
7	-0.00376	-0.99075	3.04724
8	-0.00334	-0.99032	3.04938
9	-0.01465	-1.00164	3.05025
10	0.00123	-0.98575	3.08323

Πίνακας 11: Τιμές που δίνει ο FA για την συνάρτηση Goldstein 's price

Παρατηρούμε ότι για x=-0.00985 ≈ 0 και y=-0.9959 ≈ -1, η τιμή της f(-0.00985, -0.9959)= 3.08323 ≈ 3, δηλαδή πολύ κοντά στην τιμή της συνάρτησης. Η τελική θέση των πυγολαμπίδων απεικονίζεται στην Εικόνα 27.



Εικόνα 26: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Goldstein ‘s price του Αλγορίθμου Πυγολαμπίδας

Ο συνδυασμός του σταθερού, ως προς την εύρεση λύσης, αλγορίθμου Αναζήτησης Αρμονίας και του γρήγορου αλγορίθμου Πυγολαμπίδας, κάνουν τον υβριδικό αλγόριθμο HSFA να παράγει εκπληκτικά αποτελέσματα. Ο υβριδικός αλγόριθμος ανακαλύπτει το ολικό ελάχιστο της συνάρτησης από την πρώτη εκτέλεση με τρομακτικά καλύτερη προσέγγιση της τιμής.

Για τον αλγόριθμο χρησιμοποιήθηκαν 10 πυγολαμπίδες, εκ των οποίων οι 2 είναι οι κορυφαίες» σε 100 επαναλήψεις. Οι τιμές που παράγει ο αλγόριθμος είναι αυτές του Πίνακα 12.

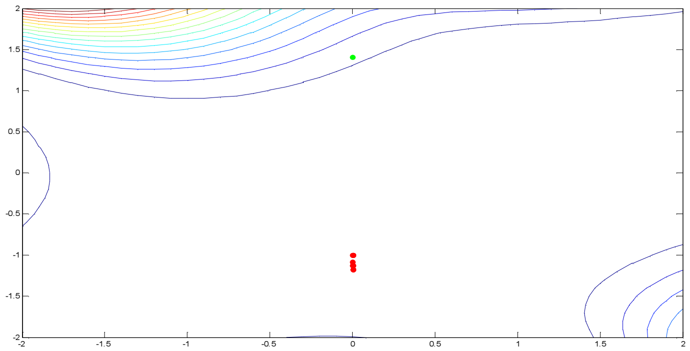
```
>> HSFA(10,2,100);

Results =
```

	x	y	f(x,y)
1	0.00055	-1.00013	3.00010
2	0.00021	-1.00047	3.00013
3	-0.00237	-1.00305	3.00388
4	0.00379	-0.99689	3.00524
5	0.00335	-0.99657	3.00540
6	0.00609	-0.99457	3.01484
7	-0.02378	-0.99443	3.18134
8	-0.02378	-0.95938	3.97831
9	-0.08744	-0.99443	5.24138
10	-0.08744	-1.11614	11.17482

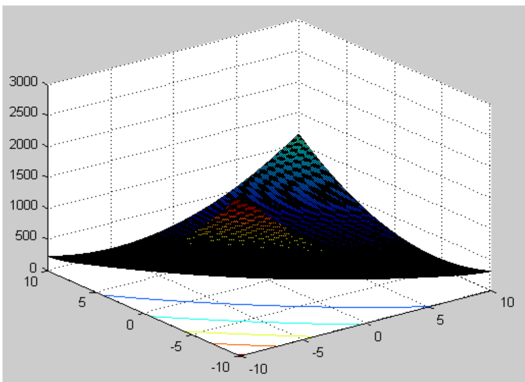
Πίνακας 12: Τιμές που δίνει ο HSFA για την συνάρτηση Goldstein ‘s price

Παρατηρούμε ότι για x=0.00055≈0, και y=-1.00013≈-1, η τιμή που δίνει η συνάρτηση είναι η f(0.00055, -1.00013)=3.00010≈3, μια πάρα πολύ καλή προσέγγιση του ολικού βέλτιστου και παράλληλα μια από τις πιο προφανείς σε σχέση με τους υπόλοιπους αλγορίθμους. Οι κινήσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 27.



Εικόνα 27: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Goldstein 's price με την χρήση του υβριδικού Αλγορίθμου HSFA

4. Η συνάρτηση Booth's, της οποίας η γραφική παράσταση απεικονίζεται στην Εικόνα 28, παρουσιάζει ακρότατο, ολικό ελάχιστο στο σημείο [1,3] με f(1.3)=0 στο διάστημα [-10,10].



Εικόνα 28: Γραφική Παράσταση συνάρτησης Booth 's

Ο αλγόριθμος Αναζήτησης Αρμονίας για την συνάρτηση αυτή ξεχωρίζει διότι βρίσκει την λύση απόλυτα από την πρώτη κιόλας επανάληψη, χωρίς να αντιμετωπίζει καμία δυσκολία. Η ρύθμιση των παραμέτρων του τον κάνουν να συγκλίνει απότομα στην λύση και να την διατηρεί στις 10000 επαναλήψεις που κάνει. Πρέπει να σημειωθεί ότι στο συγκεκριμένο πείραμα υπερτερεί σε σχέση με τους υπόλοιπους αλγορίθμους. Οι τιμές που παράγει φαίνονται στον Πίνακα 13.

```
>> HS();

Results =
```

	x	y	f(x,y)
1	1.00000	3.00000	0
2	1.00000	3.00000	0
3	1.00000	3.00000	0
4	1.00000	3.00000	0
5	1.00000	3.00000	0
6	1.00000	3.00000	0
7	1.00000	3.00000	0
8	1.00000	3.00000	0
9	1.00000	3.00000	0
10	1.00000	3.00000	0

Πίνακας 13: Τιμές που δίνει ο HS για την συνάρτηση Booth 's

Ο αλγόριθμος Πυγολαμπίδας έδειξε ένα καλό αποτέλεσμα πολύ κοντά στην βέλτιστη λύση, όμως σε καμιά περίπτωση δεν έχει την αποτελεσματικότητα του αλγορίθμου Αναζήτησης Αρμονίας. Στον παράδειγμα αυτό έγινε χρήση 10 πυγολαμπίδων για 100 γενεές και οι τιμές που παράγει ο αλγόριθμος είναι αυτές του Πίνακα 14.

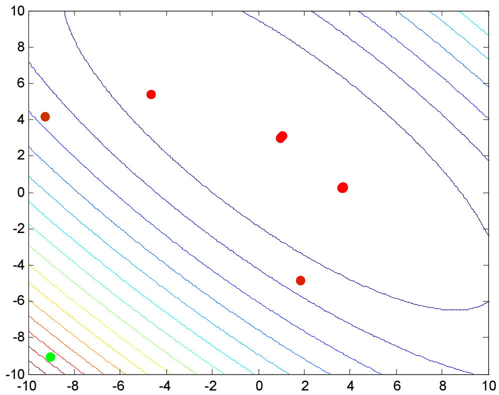
```
>> FA(10,100);

Results =
```

	x	y	f(x,y)
1	0.96645	3.04076	0.00300
2	0.97847	3.05279	0.00716
3	3.67607	0.32402	14.32224
4	3.64231	0.29057	14.34086
5	3.63907	0.28668	14.34880
6	3.72028	0.36890	14.35445
7	-4.76437	5.29397	86.66487
8	1.84673	-4.82073	256.42767
9	-9.27595	4.18720	437.42606
10	-8.96158	-8.96158	2164.81077

Πίνακας 14: Τιμές που δίνει ο FA για την συνάρτηση Booth 's

Το ολικό ελάχιστο είναι πολύ κοντά στο επιθυμητό με x*=(0.96645, 3.04076) ≈ (1,3) Οι τελικές θέσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 29.



Εικόνα 29: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Booth 's του Αλγορίθμου Πυγολαμπίδας

Ο υβριδικός αλγόριθμος προσεγγίζει την λύση αρκετά καλά, ωστόσο κληρονομεί την αδυναμία του αλγορίθμου πυγολαμπίδας και άρα δεν έχει τα αποτελέσματα του αλγορίθμου Αρμονίας . Για 10 πυγολαμπίδες, εκ των οποίων οι 2 είναι οι «κορυφαίες» σε εξέλιξη 100 γενεών, ο υβριδικός αλγόριθμος παράγει τις τιμές του Πίνακα 15.

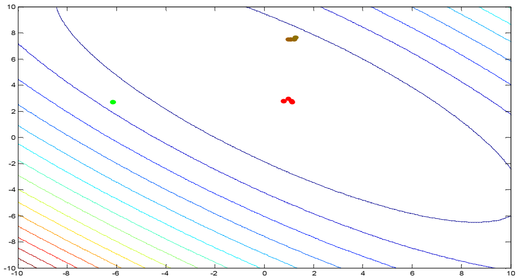
```
>> HSFA(10,2,100);

Results =
```

	x	y	f(x,y)
1	0.93991	3.07885	0.01124
2	0.96487	2.97777	0.01489
3	1.12709	2.82752	0.05415
4	0.82294	3.07885	0.07614
5	1.12709	3.07885	0.19201
6	0.93991	2.82752	0.24973
7	0.93991	2.82752	0.24973
8	0.82294	7.48297	94.29174
9	0.93991	7.53750	100.78126
10	0.96487	7.56855	103.08028

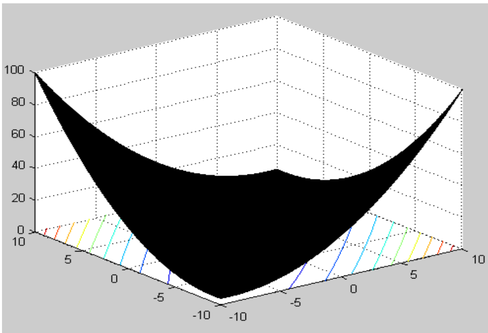
Πίνακας 15: Τιμές που δίνει ο HSFA για την συνάρτηση Booth 's

Αν δούμε την καλύτερη λύση που βρίσκει για x=0.93991 ≈ 1 και y=3.07885 ≈3, καταλαβαίνουμε ότι βρίσκεται πολύ κοντά στην βέλτιστη λύση με f(0.93991, 3.07885)=0.01124. Ωστόσο σε καμιά περίπτωση δεν μπορεί να ξεπεράσει την επίδοση του Αλγορίθμου Αναζήτησης Αρμονίας. Η τελικές θέσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 30.



Εικόνα 30: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Booth 's με την χρήση του υβριδικού Αλγορίθμου HSFA

5. Για την συνάρτηση Matya's, η γραφική παράσταση απεικονίζεται στην Εικόνα 31.



Εικόνα 31: Γραφική Παράσταση συνάρτησης Matya 's

Η συνάρτηση παρουσιάζει ολικό ελάχιστο στο σημείο [0,0], με f(0,0)=0 στο διάστημα $[-10,10]$.

Ο αλγόριθμος Αναζήτησης Αρμονίας δίνει μια ικανοποιητική προσέγγιση του ολικού βέλτιστου καθώς σε 20000 επαναλήψεις παράγει τις τιμές του Πίνακα 16. Ωστόσο η προσέγγιση που κάνει δεν δίνει μεγάλη ακρίβεια λύσεων.

```
>> HS();

Results =
```

	x	y	f(x,y)
1	-0.41370	-0.39529	0.00663
2	-0.22944	-0.04344	0.00939
3	0.84600	0.89283	0.03078
4	0.92020	0.84590	0.03257
5	1.00000	0.91851	0.03847
6	0.91642	1.00000	0.03847
7	1.00000	0.91527	0.03848
8	1.00000	0.91412	0.03848
9	0.93501	1.00000	0.03850
10	0.93525	1.00000	0.03850

Πίνακας 16: Τιμές που δίνει ο HS για την συνάρτηση Matya 's

Ο αλγόριθμος Πυγολαμπίδας δίνει μια ακόμα καλύτερη προσέγγιση με μεγαλύτερη ακρίβεια ως προς την βέλτιστη λύση. Χρησιμοποιήθηκαν 10 πυγολαμπίδες σε 100 γενεές για την παραγωγή των αποτελεσμάτων του Πίνακα 17.

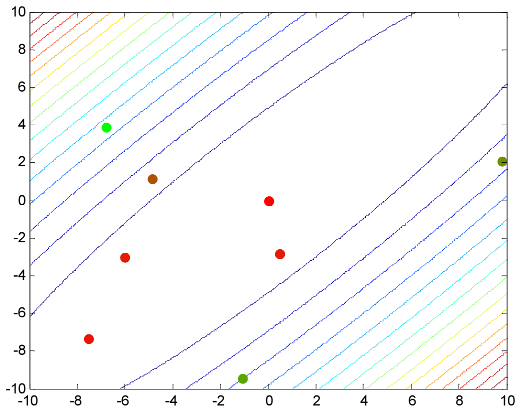
```
>> FA(10,100);

Results =
```

	x	y	f(x,y)
1	0.01174	-0.00947	0.00011
2	0.01430	-0.00691	0.00011
3	-7.55607	-7.34483	2.23152
4	0.50116	-2.85979	2.87963
5	0.49973	-2.86275	2.88241
6	-6.01370	-3.02179	3.05428
7	-4.84528	1.12696	9.05519
8	9.77130	2.10214	16.11379
9	-1.08461	-9.45681	18.63466
10	-6.79351	3.89010	28.61917

Πίνακας 17: Τιμές που δίνει ο FA για την συνάρτηση Matya 's

Παρατηρούμε οτι στις δυο πρώτες γραμμές του πίνακα έχουμε τις καλύτερες προσεγγιστικές τιμές της βέλτιστης λύσης. Για $x=0.01174 \approx 0$ και $y=-0.000947 \approx 0$, έχουμε ότι $f(0.01174, -0.000947)=0.00011\approx0$. Οι τελικές θέσεις των πυγολαμπίδων παρουσιάζονται στην Εικόνα 32.



Εικόνα 32: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Matya 's του Αλγορίθμου Πυγολαμπίδας

Τέλος ο υβριδικός αλγόριθμος παράγει τις προσεγγιστικές λύσεις του Πίνακα 18, με την χρήση 10 πυγολαμπίδων, εκ των οποίων οι 2 είναι οι «κορυφαίες» σε 100 γενεές.

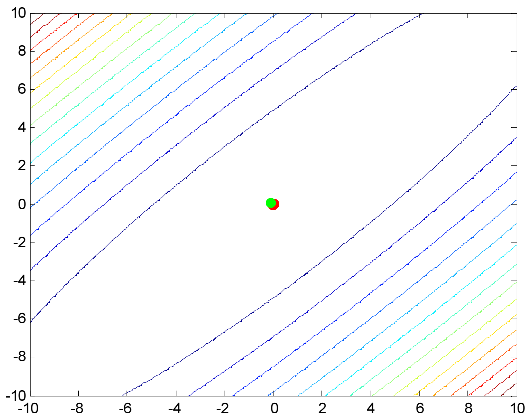
```
>> HSFA(10,2,100);

Results =
```

	x	y	f(x,y)
1	-0.00163	0.00193	3.17447e-006
2	-0.00132	0.00224	3.18216e-006
3	0.00031	0.00387	3.34884e-006
4	-0.00425	-0.00068	3.41668e-006
5	-0.00450	-0.00094	3.47071e-006
6	-0.00466	-0.00109	3.50463e-006
7	-0.00706	-0.00350	4.28776e-006
8	-0.00769	-0.00413	4.57067e-006
9	9.71907	0.14065	23.90869
10	9.76625	0.00693	24.76622

Πίνακας 18: Τιμές που δίνει ο HSFA για την συνάρτηση Matya 's

Παρατηρούμε ότι το 80% των λύσεων είναι πολύ κοντά στην βέλτιστη λύση. Άρα χωρίς να υπερβάλουμε ο υβριδικός αλγόριθμος είναι κατά πολύ ανώτερος ως προς την ταχύτητα σύγκλισης και την προσεγγιστική ακρίβεια της βέλτιστης λύσης των υπόλοιπων αλγορίθμων. Οι κινήσεις των πυγολαμπίδων απεικονίζονται στην Εικόνα 33.



Εικόνα 33: Η τελική θέση των πυγολαμπίδων στην συνάρτηση Matya 's με την χρήση του υβριδικού Αλγορίθμου HSFA

5.4 ΣΥΜΠΕΡΑΣΜΑΤΑ ΕΡΕΥΝΑΣ

Σε αυτό το τελευταίο κομμάτι της εργασίας θα αναφέρουμε κάποιες παρατηρήσεις του προέκυψαν ύστερα από την μελέτη και την πειραματική έρευνα αυτών των τριών αλγορίθμων.

Αρχικά στον αλγόριθμο Αναζήτησης Αρμονίας (HSA) παρατηρείται το πρόβλημα καθορισμού των παραμέτρων. Ωστόσο παρόλο που αποφεύγεται η ρύθμιση των HMCR και PAR, αποσιωπάται η ανάγκη πιθανής ρύθμισης των συντελεστών θορύβου. Μια μεγάλη συνέπεια αυτού του προβλήματος είναι η παγίδευση του αλγορίθμου σε τοπικά βέλτιστα. Ο αλγόριθμος παραμένει κολλημένος σε ένα τοπικό βέλτιστο χωρίς να έχει την δυνατότητα άλματος σε μια περιοχή του χώρου λύσης όπου θα υπάρχουν καλύτερες τιμές και συνεπώς η εύρεση του ολικού βέλτιστου είναι μη εφικτή.

Στον αλγόριθμο Πυγολαμπίδας η ρύθμιση των παραμέτρων είναι το μείζων πρόβλημα που παρατηρείται. Η τυχαιότητα που παρέχει η μεταβλητή *α* αρκετές φορές είναι ευεργετική στην εφαρμογή του αλγορίθμου με αποτέλεσμα την άμεση εύρεση του ολικού βέλτιστου, από την άλλη όμως μια λάθος θεώρηση της ίσως προκαλέσει την ολοκληρωτική αποτυχία του.

Στον υβριδικό αλγόριθμο Πυγολαμπίδας με χρήση Αρμονικής Αναζήτησης, ένα πλήθος κορυφαίων πυγολαμπίδων εισάγεται για να μειωθεί ο χρόνος εκτέλεσης του αλγορίθμου. Επίσης χρησιμοποιείται μετάλλαξη των πυγολαμπίδων κατά την ενημέρωση τους. Το νέο αρμονικό διάνυσμα παίρνει τη θέση της νέας πυγολαμπίδας μόνο εάν αυτή είναι καλύτερη από ό,τι η προηγούμενη. Η μέθοδος αυτή υπερτερεί γενικά από τους HS και FA. Ο HSFA προσπαθεί να εκμεταλλευτεί πλεονεκτήματα των FA και HS, έτσι ώστε να ξεφύγουν όλες οι πυγολαμπίδες από την παγίδευση σε τοπικά βέλτιστα. Ο HSFA είναι σε θέση να κάνει χρήση της ωφέλιμης γνώσης αποτελεσματικά και να βρει πολύ καλύτερες τιμές σε σχέση με τους άλλους αλγορίθμους βελτιστοποίησης.

Στην πειραματική έρευνα που κάναμε παρατηρήσαμε ότι σε ποσοστό 70% των πειραμάτων ο υβριδικός αλγόριθμος HSFA είναι κατά πολύ ανώτερος ως προς την εύρεση λύσης και την ταχύτητα σύγκλισης από τους υπόλοιπους αλγορίθμους. Συγκεκριμένα έδωσε τα καλύτερα αποτελέσματα στις συναρτήσεις Ackley's, Goldstein 's price και Matya 's και τα αμέσως καλύτερα στις συναρτήσεις Beale 's και Booth's. Στην τελευταία περίπτωση μάλιστα τα αποτελέσματα είχαν μεγάλη ακρίβεια λύσης. Από την άλλη πλευρά ο αλγόριθμος Αναζήτησης Αρμονίας είχε την καλύτερη επίδοση στις συνάρτησεις Beale 's και Booth's, κάτι που προέκυψε διότι το ολικό βέλτιστο των συναρτήσεων είναι προφανές και ελάχιστες οι πιθανότητες να κολλήσει σε τοπικά βέλτιστα. Στην Booth's η σύγκλιση του ήταν ακαριαία από την πρώτη κιόλας επανάληψη. Επίσης είχε την δεύτερη καλύτερη επίδοση στην συνάρτηση Ackley's. Τέλος ο αλγόριθμος Πυγολαμπίδας είχε αρκετά καλές επιδόσεις στις συναρτήσεις Goldstein's price, και Booth's. Αυτό σαφώς δεν σημαίνει ότι ο αλγόριθμος Πυγολαμπίδας υστερεί συγκριτικά με τους άλλους. Άλλωστε το πείραμα έγινε με ένα καθορισμένο δείγμα συναρτήσεων και όχι σε ένα γενικό εύρος συναρτήσεων βελτιστοποίησης.

Συνοψίζοντας θα επαναλάβουμε για ακόμη μια φορά ότι ο υβριδικός αλγόριθμος είναι ο πιο αποδοτικός στην βελτιστοποίηση των benchmark συναρτήσεων και ενδείκνυται και στην βελτιστοποίηση προβλημάτων μεγαλύτερων διαστάσεων. Ωστόσο και οι μεμονωμένοι αλγόριθμοι Αναζήτησης Αρμονίας και Πυγολαμπίδας είναι αρκετά καλά εργαλεία βελτιστοποίησης. Στο άμεσο μέλλον μια μεγάλη γκάμα εφαρμογών αναμένεται να αναπτυχθούν γύρω από αυτούς τους αλγορίθμους. Ας έχουμε τα μάτια μας ανοιχτά…

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Dennis J. E. , Schnabel., R. B.(1983) "Numerical Methods for Unconstrained Optimization and Nonlinear Equations". Prentice Hall

2. Ζησιμόπουλος,Β. (2007): "Συνδυαστική Βελτιστοποίηση", Αθήνα, Πανεπιστημιακές Σημειώσεις ΕΚΠΑ

3. Konaka,A, Coitb, D.W. (2006), : "Multi-objective optimization using genetic algorithms: A tutorial reliab". Eng. Syst. Safe., 91(9):992–1007

4. Maaroju,N., Garg, D. ,(2008),: " Choosing the best heuristic for NP- problem. International Journal of Information Technology and Knowledge Management" Vol:1 , σελ. 537 -547

5. Blum,C., Roli,A., (2003), :"Metaheuristics in Combinatorial Optimization: Overviewand Conceptual Comparison ACM Computing Surveys", Vol. 35, No. 3, σελ. 269–308

6. Geem, Z.W. , (2009) : "MUSIC INSPIRED HARMONY SEARCH ALGORITHM: THEORY AND APLICATIONS", (ED), Springer, Berlin

7. Lee, K.S., Geem, Z.W. ,(2004): " A New Structural Optimization Method Based on the Harmony Search Algorithm". Comput. Struct. 82(9/10),σελ. 781–798

8. Yang, X.-S. , (2009a) "Harmony Search as a Metaheuristic Algorithm", in: Music-Inspired Harmony Search Algorithm: Theory and Applications (Editor Z. W. Geem), Studies in Computational Intelligence, Springer Berlin, vol. 191, σελ. 1-14

9. Geem Z.W. (2006) "Improved Harmony Search from Ensemble of Music Players", Springer - Verlag, KES 2006, Part I, LNAI 4251, σελ. 86-93.

10. Mahdavi M, Fesanghary M, Damangir E , (2007): "An improved harmony search algorithm for solving optimization problems". Appl Math Comput 188: 1567–1579

11. Omran, M.G.H., Mahdavi, M. (2008): "Global-best harmony search. Applied Mathematics and Computation "198(2), σελ. 643–656

12. Wang C.M., Huang Y.F. (2010) "Self-adaptive harmony search algorithm for optimization", Expert Systems with Applications, Vol.37, σελ. 2826-2837

13. Geem, Z.W., Sim, K.B. (2010): "Parameter-setting-free harmony search algorithm. Applied Mathematics and Computation" 217(8), 3881–3889

14. Fister, I., Fister , Jr., Yang, X.-S.,Brest,J. (2013): "A comprehensive review of firefly algorithms ,Swarm and Evolution Computation", vol.13,σελ. 34 -46

15. X. -S. Yang, (2010a) "Firefly algorithm, Levy flights and global optimization, in Research and Development in Intelligent Systems XXVI", σελ. 209–218, Springer, London, UK.

16. X. -S. Yang (2010b): "Nature – Inspired Metaheuristic Algorithms" , Second Edition, Luniver Press:UK

17. X. -S. Yang (2009b): "Firefly Algorithms for Multimodal Optimization, in Stochastc algorithms, Foundations and Applications" SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, σελ. 169-178

18. Khadwilard,A., Chansombat,S., Thepphakorn,T., Thapatsuwan,P., Chainate,W., Pongcharoen,P., 2012: "Application of firefly algorithm and itsparameter setting for job shop scheduling". The Journal of Industrial Technology, Vol. 8, No. 1.σελ9

19. L. dos Santos Coelho, D. L. de Andrade Bernert, V. C. Mariani, (2011).: "A chaotic firefly algorithm applied to reliability-redundancy optimization": IEEE Congress on Evolutionary Computation (CEC'11), σελ. 517-521 .

20. Sayadi, M. K.; Ramezanian, R.; Ghaffari-Nasab, N. (2010). "A discrete firefly metaheuristic with local search for makespan minimization in permutation flow shop scheduling problems". Int. J. of Industrial Engineering Computations 1: σελ.1–10.

21. Apostolopoulos, T.; Vlachos, A. (2011). "Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem". International Journal of Combinatorics 2011: Article ID 523806.

22. Rampriya B., Mahadevan K. and Kannan S., (2010). Unit commitment in deregulated power system using Lagrangian firefly algorithm, Proc. of IEEE Int. Conf. on Communication Control and Computing Technologies (ICCCCT), pp. 389-393

23. Malek, M., M. Guruswamy, H. Owens, and M. Pandya, (1989) :"A Hybrid Algorithm Technique," The University of Texas at Austin, Dept. of Computer Sciences Technical Report TR-89-06.

24. Guo,L.,Wang, G.-G,Wang,H.,Wang,D., (2013):" An Effective Hybrid Firefly Algorithm with Harmony Search for Global Numerical Optimization", Hindawi Publishing Corporation ,The ScientificWorld Journal ,Vol. 2013, (ID 125625), σελ. 1-9

25. Adorio, E. P., & Diliman, U. P. MVF - Multivariate Test Functions Library in C for Unconstrained Global Optimization (2005). Retrieved June 2013,(http://www.geocities.ws/eadorio/mvf.pdf)