ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

Μεταπτυχιακό πρόγραμμα σπουδών

Επιχειρησιακή Έρευνα

Τσαμπουνάρη Ειρήνη

Επιβλέπων Καθηγητής: Μαρινάκης Ιωάννης

Υλοποίηση και συγκριτική ανάλυση αλγορίθμων εμπνευσμένων από τη φύση για προβλήματα ολικής βελτιστοποίησης

Χανιά, 2020

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε το διάστημα μεταξύ Οκτωβρίου 2018 και Απριλίου 2019 στα πλαίσια του μεταπτυχιακού προγράμματος στην Επιχειρησιακή Έρευνα της σχολής Μηχανικών Παραγωγής & Διοίκησης του Πολυτεχνείου Κρήτης. Πρώτα απ' όλα θα ήθελα να εκφράσω τις ευχαριστίες και την ευγνωμοσύνη μου στον καθηγητή μου Μαρινάκη Ιωάννη για την πολύτιμη βοήθεια που μου παρείχε κατά την εκπόνηση της παρούσας διπλωματικής εργασίας καθώς και για την εμπιστοσύνη και την αγάπη που μου έδειξε όλα τα χρόνια των σπουδών μου. Επιπλέον, θα ήθελα να ευχαριστήσω την οικογένεια μου, τον πατέρα μου, Μανώλη, αδερφό μου, Μικέ Τσαμπουνάρη και μητέρα μου Αναστασία Ασλάνογλου για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια.

Π ε ϱ ι ε χ ό μ ε ν α

Εισαγωγή

Σκοπός της τρέχουσας διπλωματικής εργασίας είναι η σύγκριση κάποιων αλγορίθμων εμπνευσμένων από τη φύση σε προβλήματα ολικής βελτιστοποίησης. Τα προβλήματα αυτά είναι συναρτήσεις δοκιμής και είναι χρήσιμες για την αξιολόγηση των αλγορίθμων βελτιστοποίησης. Ουσιαστικά, μέσω αυτών παρατηρούμε την απόδοση κάθε μεθοδολογίας.

Αρχικά, για την καλύτερη κατανόηση των επόμενων κεφαλαίων, το πρώτο κεφάλαιο αναφέρεται σε βασικές έννοιες, όπως, η ολική βελτιστοποίηση, και στα προβλήματα τα οποία επιλύονται. Πρόκειται για προβλήματα που έχουν μη γραμμική αντικειμενική συνάρτηση.

Εν συνεχεία, στο δεύτερο κεφάλαιο επεξηγούνται οι ορισμοί των εξελικτικών αλγορίθμων και της νοημοσύνης σμήνους και γίνεται αναλυτική αναφορά στις μεθοδολογίες που υλοποιήθηκαν. Πιο συγκεκριμένα, οι αλγόριθμοι αυτοί είναι ο αλγόριθμος αναζήτησης της βαρυτικής έλξης (GSA), ο αλγόριθμος της μεγάλης έκρηξης (BB-BC), ο αλγόριθμος βελτιστοποίησης του γκρίζου λύκου (GWO), ο αλγόριθμος των μαϊμούδων (SMO), ο αλγόριθμος της φάλαινας (WO), ο αλγόριθμος της λιβελούλας (DA) και ο αλγόριθμος των μυρμηλεοντίδων (ALO).

Στο τρίτο και τελευταίο κεφάλαιο εμφανίζονται τα αποτελέσματα και σχολιάζονται σαφώς.

Κεφάλαιο 1ο

Βασικές έννοιες

1.1 Γενικά στοιχεία

Το ευρύτερο πεδίο της βελτιστοποίησης ή λήψης βέλτιστων αποφάσεων περιλαμβάνει την αντίστοιχη θεωρία και εφαρμογές σε όλους σχεδόν του τομείς των επιστημών, και ειδικά των τεχνικών και οικονομικών επιστημών. Η βελτιστοποίηση είναι πηγή τεχνητής ευφυΐας με την έννοια του προσδιορισμού βέλτιστων λύσεων σε πολύπλοκα προβλήματα που ξεπερνούν τις δυνατότητες εμπειρογνωμόνων. Με την χρήση ηλεκτρονικών υπολογιστών η επίλυση των προβλημάτων επιτυγχάνεται ταχύτατα μέσω κατάλληλων αλγορίθμων.

Κάθε πρόβλημα μη γραμμικής βελτιστοποίησης εμπεριέχει μια αντικειμενική συνάρτηση προς μεγιστοποίηση ή ελαχιστοποίηση. Ανάλογα με την υφή του, μπορεί επίσης το εκάστοτε πρόβλημα να περιλαμβάνει περιορισμούς ισότητας ή ανισότητας. Στην παρούσα έρευνα, τα μη γραμμικά προβλήματα, τα οποία είναι προς βελτιστοποίηση, δεν περιλαμβάνουν περιορισμούς.

Τα προβλήματα που εμφανίζονται στη φύση μπορεί να είναι δύσκολα προβλήματα βελτιστοποίησης ή να μοντελοποιούνται από συνεχείς συναρτήσεις που εμφανίζουν ένα μεγάλο πλήθος τοπικά βέλτιστων λύσεων κάνοντας κατά αυτόν τον τρόπο δύσκολη αν όχι αδύνατη την επίλυση τους. Υπάρχουν πολλά παραδείγματα στη βιβλιογραφία που αποτυγχάνουν να εντοπίσουν το ολικό ελάχιστο μιας αντικειμενικής συνάρτησης. Για το λόγο αυτό τα τελευταία χρόνια η έρευνα κατευθύνθηκε προς στοχαστικούς ευρετικούς αλγόριθμους όπως ο simulated annealing και ειδικότερα σε αυτούς που εμπνέονται από φυσικά ή βιολογικά φαινόμενα όπως οι γενετικοί αλγόριθμοι. Εν γένει οι αλγόριθμοι αυτοί υπερέχουν έναντι των κλασικών αιτιοκρατικών μεθόδων επειδή έχουν την ιδιότητα να μην εγκλωβίζονται σε τοπικά βέλτιστες λύσεις και συνήθως μπορούν να εντοπίσουν μία από τις περιοχές της ολικά βέλτιστης λύσης μετά από ένα μεγάλο αριθμό αποτιμήσεων της αντικειμενικής συνάρτησης.

Η διαδικασία εύρεσης λύσεων στους στοχαστικού ευρετικούς αλγόριθμους καθώς και στα υβριδικά σχήματα αυτών πραγματοποιείται πάντοτε μέσω μιας επαναληπτικής διαδικασίας. Σε κάθε επανάληψη δημιουργούνται από τον αλγόριθμο μία ή περισσότερες λύσεις από τον αντίστοιχο αριθμό ατόμων (individuals) που ανήκουν στο χώρο των πιθανών λύσεων. Η καταλληλότητα των λύσεων αυτών εκτιμάται μέσω της αντικειμενικής συνάρτησης. Πολλές φορές οι αλγόριθμοι αυτοί, κατά τη διαδικασία εύρεσης λύσεων προσαρμόζονται συνεχώς στο χώρο των λύσεων μεταβάλλοντας τις παραμέτρους τους. Η παραπάνω γενική διαδικασία συνεχίζεται έως ότου ολοκληρωθεί ένας συγκεκριμένος αριθμός επαναλήψεων (γενιές) ή ικανοποιηθεί ένα κριτήριο σύγκλισης (επίτευξη προκαθορισμένης τιμής). Τα τρία κύρια μειονεκτήματα των στοχαστικών αλγορίθμων είναι τα εξής: α) πολλές φορές δυσκολεύονται να προσεγγίσουν την ακριβή τιμή της βέλτιστης λύσης του προβλήματος αν και συνήθως εντοπίζουν γρήγορα την περιοχή στην οποία αυτή βρίσκεται, β) απαιτούν μεγάλο αριθμό εκτιμήσεων της αντικειμενικής συνάρτησης και επομένως έχουν υψηλό υπολογιστικό κόστος προκειμένου να δώσουμε καλό αποτέλεσμα και γ) συνήθως έχουν ένα μεγάλο αριθμό παραμέτρων ο οποίος κάνει δύσκολη τη βέλτιστη ρύθμιση του αλγόριθμου.

1.2 Το πρόβλημα της ολικής βελτιστοποίησης

Θεωρούμε το πρόβλημα της ελαχιστοποίησης μιας συνεχούς και διαφορίσιμης συνάρτησης $f : D \subseteq \mathbb{R}^n \to \mathbb{R}$, στο κυρτό χωρίο$[x] \epsilon IR^n$ όπου, $[x] \subseteq D$ και D ανοικτό σύνολο. Δηλαδή αναζητούμε το ολικό ελάχιστο

$$f^* = \min_{x \in [x]} f(x)$$

καθώς και το σύνολο των σημείων ολικού ελάχιστου

$x^* \;\; = \{\, x^* \in [x] : f(x^*) = f^* \,\}$

Να σημειώσουμε εδώ ότι η ελαχιστοποίηση μιας συνάρτησης f είναι ισοδύναμη με την μεγιστοποίηση μιας άλλης συνάρτησης g, όπου g = – f. Στη συνέχεια με τον όρο βελτιστοποίηση θα εννοούμε την ελαχιστοποίηση της συνάρτησης f.

1.3 Ποια προβλήματα επιλύονται

Τα προβλήματα που επιλύονται ανήκουν στην κατηγορία των συναρτήσεων *δοκιμής* και είναι χρήσιμα για την αξιολόγηση των χαρακτηριστικών των αλγορίθμων βελτιστοποίησης όπως: το ποσοστό σύγκλισης, η ακρίβεια και η ευρωστία. Δηλαδή, με άλλα λόγια, μέσω αυτών των συναρτήσεων ελέγχονται οι γενικές επιδόσεις του κάθε αλγόριθμου. Οι μέθοδοι που αναπτύξαμε θα ελεγχθούν με τις ακόλουθες τέσσερις συναρτήσεις δοκιμής, οι οποίες έχουν ένα εκ των προτέρων γνωστό ολικό ελάχιστο. Σε αυτό το σημείο αξίζει να σημειωθεί ότι οι συναρτήσεις που χρησιμοποιήσαμε ανήκουν στην κατηγορία της μονοκριτήριας βελτιστοποίησης, διότι υπάρχουν κι άλλα τέτοια μαθηματικά προβλήματα τα οποία ανήκουν στην πολυκριτήρια βελτιστοποίηση, η οποία έχει περισσότερες από μία αντικειμενικές συναρτήσεις ή κριτήρια. Οι συναρτήσεις με τις οποίες ασχολείται η έρευνα είναι η Rosenbrock, η Sphere, η Rastrigin και η Griewank. Στη συνέχεια ακολουθεί η παρουσίαση αυτών.

Η συνάρτηση **Rosenbrock**:

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

Είναι μια μη κυρτή συνάρτηση που εισήχθη από τον Howard. Το ολικό ελάχιστο είναι μέσα σε μια μακρά, στενή, παραβολική επίπεδη κοιλάδα. Ωστόσο, η σύγκλιση στο ολικό ελάχιστο είναι δύσκολη.

Η συνάρτηση **Sphere**:

$$f(x) = \sum_{i=1}^{n} x_i^2$$

Η συνάρτηση **Rastrigin**:

$$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$$

Η συνάρτηση **Griewank**:

$$f(x) = \sum_{i=1}^{n} \frac{(x_i - 100)^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i - 100}{\sqrt{(i)}}\right) + 1$$

Στις συναρτήσεις Sphere, Rastrigin και Griewank, τα ολικά ελάχιστα είναι f (x *)=0 με x*=(0, ⋯, 0), ενώ στη Rosenbrock τα ολικά ελάχιστα είναι f (x *) = 0 με x*=(1, ⋯, 1).

Κεφάλαιο 2ο

Εξελικτικοί αλγόριθμοι επίλυσης συναρτήσεων δοκιμής

Εισαγωγή

Στο κεφάλαιο αυτό θα γίνει αναφορά στους αλγόριθμους που υλοποιήθηκαν με σκοπό τη βελτιστοποίηση των μαθηματικών προβλημάτων που προαναφέρθηκαν. Αρχικά, θα οριστεί τί είναι ένας εξελικτικός αλγόριθμος και γιατί επιλέξαμε τη χρήση αυτών έναντι άλλων κλασικών μεθόδων για την επίλυση των προβλημάτων, και στη συνέχεια θα γίνει εκτενής παρουσίαση των μεθόδων αυτών.

2.1 Ορισμός Εξελικτικών Αλγόριθμων

Τα τελευταία τριάντα χρόνια, παρατηρείται ένα συνεχώς αυξανόμενο ενδιαφέρον για την ανάπτυξη συστημάτων επίλυσης προβλημάτων βασισμένων στις αρχές της φυσικής εξέλιξης. Τα συστήματα αυτά λειτουργούν διαιρώντας έναν πληθυσμό κωδικοποιημένων πιθανών λύσεων του προβλήματος που προσπαθούμε να επιλύσουμε, και εφαρμόζοντας πάνω σε αυτόν διάφορες διαδικασίες εμπνευσμένες από τη βιολογική εξέλιξη. Έτσι, περνώντας από γενιά σε γενιά, τα συστήματα αυτά δημιουργούν συνεχώς νέους πληθυσμούς πιθανών λύσεων εξελίσσοντας τους προηγούμενους πληθυσμούς.

Οι γενετικοί αλγόριθμοι είναι ένα παράδειγμα τέτοιου συστήματος που μαζί με τον εξελικτικό προγραμματισμό, τις στρατηγικές εξέλιξης, τα συστήματα ταξινόμησης και το γενετικό προγραμματισμό αποτελούν μια κατηγορία συστημάτων επίλυσης προβλημάτων που είναι ευρύτερα γνωστή με τον όρο Εξελικτικοί Αλγόριθμοι.

Οι ΕΑ είναι αλγόριθμοι ανίχνευσης-αναζήτησης, βασισμένοι στη μηχανική της φυσικής επιλογής και της φυσικής γενετικής. Οι ΕΑ μιμούνται τις διαδικασίες βιολογικής εξέλιξης με την υλοποίηση των ιδεών της φυσικής επιλογής και της επικράτησης του ισχυρότερου, έτσι ώστε να παρέχουν αποτελεσματικές λύσεις σε προβλήματα αναζήτησης και βελτιστοποίησης. Υπάρχουν διάφορα εξελικτικά υπολογιστικά μοντέλα τα οποία όμως βασίζονται στις ίδιες αρχές, δηλαδή στις αρχές προσομοίωσης της εξέλιξης ατομικών δομών μέσω των διαδικασιών της επιλογής και της αναπαραγωγής.

2.2 Νοημοσύνη σμήνους (swarm intelligence)

Η νοημοσύνη σμήνους (swarm intelligence) είναι η συλλογική συμπεριφορά μη κατανεμημένων, αυτοοργανωμένων φυσικών ή τεχνητών συστημάτων. Η ιδέα εφαρμόζεται στον τομέα της τεχνητής νοημοσύνης. Ο όρος εισήχθη από τους Gerardo Beni και Jing Wang το 1989. Η νοημοσύνη σμήνους είναι μια ιδιότητα συστημάτων που επιδεικνύουν συλλογικά ευφυή συμπεριφορά. Μέσα από απλούς κανόνες εμφανίζονται φαινόμενα (σμήνη πουλιών, κοπάδια ψαριών, λάμψη πυγολαμπίδων, τα μυρμήγκια υπολογίζουν βέλτιστες διαδρομές προς την τροφή τους, μέλισσες να ενημερώνουν τη κυψέλη για νέκταρ) τα οποία οδηγούνται μέσα από τη συλλογική συμπεριφορά και ευφυΐα.

Τα συστήματα νοημοσύνης σμήνους κατά κανόνα αποτελούνται από έναν πληθυσμό απλών, αυτόνομων πρακτόρων ή διαμεσολαβητών που αλληλοεπιδρούν μεταξύ τους και με το περιβάλλον τους σε τοπικό επίπεδο. Τα συστήματα είναι εμπνευσμένα από τη φύση και ειδικότερα από τα βιολογικά συστήματα. Οι πράκτορες ακολουθούν πολύ απλούς κανόνες, και παρόλο που δεν υπάρχει καμία συγκεντρωτική δομή ελέγχου να υπαγορεύει πως πρέπει να συμπεριφέρονται οι πράκτορες, τοπικές - και έως ένα βαθμό τυχαίες - αλληλεπιδράσεις μεταξύ τέτοιων πρακτόρων οδηγούν στην εμφάνιση μιας ευφυούς, καθολικής συμπεριφοράς, άγνωστης στους αυτόνομους πράκτορες. Φυσικά παραδείγματα της νοημοσύνης σμήνους περιλαμβάνουν τις αποικίες μυρμηγκιών, τη βακτηριδιακή ανάπτυξη, τις αγέλες ή τα κοπάδια ζώων και τα κοπάδια ψαριών.

Ένα άλλο παράδειγμα νοημοσύνης σμήνους είναι τα σμήνη πουλιών που κινούνται σαν ένα σώμα παρά του ότι αποτελούνται από πολλές ανεξάρτητες οντότητες. Αν παρατηρήσει κανείς τα πουλιά θα προσέξει ότι κινούνται σαν μια ομάδα με ένα κοινό προσανατολισμό. Αυτό το σύνολο που προκύπτει είναι μέσα από τη συνολική συμπεριφορά των πουλιών. Ένα πουλί βλέπει γύρω το περιβάλλον του και αναλόγως συνυπολογίζει σε σχέση με τα άλλα πουλιά για να πάρει τις δικές του αποφάσεις. Ο νόμος του Couzin αναφέρει τρεις αρχές: αν ένα πουλί βρίσκεται μακριά από τα άλλα τότε καταλαβαίνει ότι πρέπει να αναπτύξει ταχύτητα αν όμως πλησιάζει πολύ σε ένα άλλο τότε καταλαβαίνει ότι θα συγκρουστεί και μειώνει την ταχύτητα του, να υπάρχει δηλαδή στο σμήνος μια συνοχή, αν βρει εμπόδιο το αποφεύγει, διαχωρίζεται από το σύνολο και ξανασμίγουν και τρίτο πρέπει να έχουν μια κοινή κατεύθυνση να ακολουθούν μια συγκεκριμένη πορεία. Αρχή της συλλογικής συμπεριφοράς είναι ότι τα πουλιά πάντα προσπαθούν να διατηρήσουν μια ελάχιστη απόσταση μεταξύ των ίδιων αλλά και των άλλων. Αυτός ο κανόνας έχει υψηλή προτεραιότητα και αντιστοιχεί στη συμπεριφορά των ζώων στη φύση.

2.3 Εξελικτικοί Αλγόριθμοι που υλοποιήθηκαν

Στην εργασία αυτή, για την επίλυση των προβλημάτων μας έχει χρησιμοποιηθεί μία σειρά αλγορίθμων εμπνευσμένων από τη φύση, όπως θα περιγραφούν στη συνέχεια. Αυτοί είναι:

- *Gravitational Search Algorithm, GSA*
- *Big Bang – Big Crunch, BBBC*
- *Grey Wolf Optimization, GWO*
- *Spider Monkey Optimization, SMO*
- *The Whale Optimization Algorithm, WO*
- *Dragonfly Optimization Algorithm, DA*
- *Antlion Optimization Algorithm, ALO*

2.3.1 Αλγόριθμος αναζήτησης της βαρυτικής έλξης (Gravitational Search Algorithm, GSA)

Ο αλγόριθμος αναζήτησης της βαρυτικής έλξης είναι ένας καινούριος αλγόριθμος και βασίζεται στους νόμους της βαρύτητας και της αλληλεπίδρασης διαφορετικών μαζών. Σε αυτό τον αλγόριθμο οι *λύσεις* είναι ένα *σύνολο από μάζες* που αλληλεπιδρούν μεταξύ τους βάσει της βαρύτητάς τους και τους νόμους της κίνησης. Κάθε πράκτορας θεωρείται ένα αντικείμενο που η απόδοσή του μετριέται με τη μάζα του. Όλα τα αντικείμενα έλκονται μεταξύ τους βάσει της δύναμης της βαρύτητας, και αυτή η δύναμη προκαλεί μία συνολική κίνηση των αντικειμένων προς αυτά που έχουν μεγαλύτερη μάζα. Έτσι, οι μάζες συνεργάζονται μεταξύ τους χρησιμοποιώντας τη δύναμη της βαρύτητας. Τα αντικείμενα που έχουν μεγάλη μάζα (αντιστοιχούν σε καλές λύσεις) κινούνται λιγότερο, ενώ τα αντικείμενα που είναι ελαφρύτερα κινούνται περισσότερο δίνοντας μεγαλύτερες δυνατότητες αναζήτησης στον αλγόριθμο. Τα τέσσερα συγκεκριμένα χαρακτηριστικά κάθε αντικειμένου είναι τα εξής:

- Θέση.
- Μάζα αδράνειας, συμβολίζεται με M_i και είναι ένα μέτρο που δείχνει την αντίσταση που ασκείται σε ένα αντικείμενο να αλλάξει την κατάσταση της κίνησής του όταν εφαρμόζεται μία δύναμη. Ένα αντικείμενο με μεγάλη μάζα αδράνειας αλλάζει την ταχύτητά του πιο αργά και αν έχει μικρή μάζα αδράνειας αλλάζει την ταχύτητά του πιο γρήγορα.
- Ενεργή μάζα βαρύτητας, συμβολίζεται με M_a και είναι ένα μέτρο της δύναμης του βαρυτικού πεδίου σε ένα συγκεκριμένο αντικείμενο. Το βαρυτικό πεδίο ενός αντικειμένου με μικρή ενεργή βαρυτική μάζα είναι πιο αδύναμο από το βαρυτικό πεδίο ενός αντικειμένου με μεγάλη ενεργή βαρυτική μάζα.
- Παθητική μάζα βαρύτητας, συμβολίζεται με M_p και είναι μέτρο της δύναμης που ασκεί το αντικείμενο μέσα στο βαρυτικό του πεδίο. Μέσα σε ένα συγκεκριμένο βαρυτικό πεδίο ένα αντικείμενο με μικρότερη παθητική μάζα βαρύτητας έχει μικρότερη δύναμη από ένα αντικείμενο με μεγαλύτερη παθητική μάζα βαρύτητας. Η θέση αντιστοιχεί στη λύση του προβλήματος,

ενώ η μάζα αδράνειας και οι μάζες βαρύτητας υπολογίζονται μέσω μίας συνάρτησης καταλληλότητας (fitness function). Με άλλα λόγια κάθε μάζα είναι μία λύση, και ο αλγόριθμος εξελίσσεται με την κατάλληλη προσαρμογή των μαζών αδράνειας και βαρύτητας. Με το πέρασμα του χρόνου (επαναλήψεων) αναμένεται ότι οι ελαφρότερες μάζες θα έλκονται από βαρύτερες μάζες. Η καλύτερη μάζα θα είναι και η καλύτερη λύση του προβλήματος.

Η κάθε μάζα ακολουθεί τους παρακάτω νόμους:

- Νόμος της βαρύτητας: κάθε σωματίδιο έλκει κάθε άλλο σωματίδιο ενώ η δύναμη της βαρύτητας ανάμεσα σε δύο σωματίδια είναι ανάλογη του γινομένου των μαζών τους και αντιστρόφως ανάλογη της απόστασης μεταξύ τους, R.
- Νόμος της κίνησης: η τρέχουσα ταχύτητα οποιασδήποτε μάζας είναι ίση με το άθροισμα της προηγούμενης ταχύτητας και της μεταβολής στη ταχύτητα. Η μεταβολή στην ταχύτητα ή στην επιτάχυνση κάθε μάζας είναι ίση με τη δύναμη που ενεργείται στο σύστημα διαιρεμένη από τη μάζα αδράνειας.

Έστω ότι έχουμε ένα σύστημα με N μάζες. Η θέση στο σύστημα συμβολίζεται με x_{id} όπου $i = 1, \ldots, N$ και $d = 1, \ldots, n$ οι διαστάσεις της μάζας. Τη χρονική στιγμή t η δύναμη που ασκείται στη μάζα i από τη μάζα j, δίνεται από την εξίσωση:

$$F_{ijd}(t) = G(t)\frac{M_{pi}(t) \times M_{\alpha j}(t)}{R_{ij}(t) + \varepsilon}\left(x_{id}(t) - x_{jd}(t)\right) \tag{1.1}$$

όπου $M_{pi}(t)$ είναι η παθητική μάζα βαρύτητας που συσχετίζεται με τη μάζα i, $M_{\alpha j}(t)$ είναι η ενεργή μάζα βαρύτητας που συσχετίζεται με τη μάζα j, το $G(t)$ είναι μία σταθερά βαρύτητας, ε μία μικρή σταθερά και $R_{ij}(t)$ η ευκλείδεια απόσταση των δύο μαζών τη χρονική στιγμή t. Για να δώσουν στοχαστικότητα στον αλγόριθμό τους οι ερευνητές υποθέτουν ότι η συνολική δύναμη που ασκείται σε μία μάζα i θα εξαρτάται από τις υπόλοιπες μάζες με βάση την ακόλουθη εξίσωση:

$$F_{id}(t) = \sum_{j=1, j\neq i}^{N} rand_j\, F_{ijd}(t) \quad 1.2$$

όπου το $rand_j$ είναι ένας τυχαίος αριθμός μεταξύ (0,1). Η επιτάχυνση μίας μάζας τη χρονική στιγμή t δίνεται από την εξίσωση:

$$\alpha_i(t) = \frac{F_{id}(t)}{M_{ii}(t)} \quad 1.3$$

όπου το $M_{ii}(t)$ είναι η μάζα αδράνειας. Η ταχύτητα και η θέση της κάθε μάζας δίνεται από τις ακόλουθες εξισώσεις:

$$u_{id}(t+1) = rand_i \times u_{ij}(t) + \alpha_i d(t) \quad 1.4$$

$$x_{id}(t+1) = x_{id}(t) + u_{id}(t+1) \quad 1.5$$

όπου $rand_i$ είναι ένας τυχαίος αριθμός μεταξύ (0,1). Η σταθερά βαρύτητας G, αρχικοποιείται στο ξεκίνημα των επαναλήψεων και μειώνεται με το πέρασμα των επαναλήψεων. Έτσι η σταθερά βαρύτητας είναι μία συνάρτηση της αρχικής τιμής της και του χρόνου t:

$$G(t) = G(G_0, t) \quad 1.6$$

Οι μάζες αδράνειας και βαρύτητας θεωρούνται ίσες και δίνονται από τις εξισώσεις:

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1, \dots, N \quad 1.7$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad 1.8$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \quad 1.9$$

όπου $fit_i(t)$ είναι η τιμή της αντικειμενικής συνάρτησης τη χρονική στιγμή t, $worst(t)$ συμβολίζει την τιμή της αντικειμενικής συνάρτησης της χειρότερης μάζας τη χρονική στιγμή t και $best(t)$ συμβολίζει την τιμή της αντικειμενικής συνάρτησης της καλύτερης μάζας τη χρονική στιγμή t. Για να αυξηθούν οι ικανότητες αναζήτησης του αλγορίθμου και επειδή το να έλκονται όλες οι μάζες μεταξύ τους ίσως δημιουργήσει πρόβλημα στην αναζήτηση, έχει προταθεί να ασκούν έλξη μόνο οι Kbest μάζες. Αρχικά το Kbest παίρνει μία μεγάλη τιμή η οποία στη συνέχεια μειώνεται γραμμικά με τις επαναλήψεις μέχρι να φθάσει στις τελικές επαναλήψεις όπου μόνο μία μάζα θα ασκεί δύναμη στις άλλες. Ο λόγος που γίνεται αυτό είναι για να έχει αρχικά ο αλγόριθμος περισσότερες ικανότητες αναζήτησης σε πολλά σημεία του χώρου λύσεων στις αρχικές επαναλήψεις και να έχει ο αλγόριθμος μεγαλύτερες ικανότητες αναζήτησης γύρω από το βέλτιστο σημείο στο τέλος των επαναλήψεων. Έτσι, η εξίσωση έλξης γίνεται:

$$F_{id}(t) = \sum_{j=1, j\neq i}^{Kbest} rand_j \, F_{ijd}(t) \qquad 1.10$$

Παρακάτω παρουσιάζεται ο αλγόριθμος υπό μορφή ψευδοκώδικα της παραπάνω μεθόδου:

Αλγόριθμος Αναζήτησης Βαρυτικής έλξης

Αρχικοποίηση (εισαγωγή δεδομένων)
Δημιουργία αρχικού πληθυσμού
Υπολογισμός αντικειμενικής συνάρτησης αρχικών λύσεων
for i=1,

  Υπολογισμός $G(t)$, $best(t)$, $worst(t)$ and $M_i(t)$ για κάθε i
  Υπολογισμός της συνολικής δύναμης και της επιτάχυνσης
  Υπολογισμός της ταχύτητας και της θέσης

end

2.3.2 Ο αλγόριθμος της μεγάλης έκρηξης (Big Bang – Big Crunch Algorithm, BB - BC)

Ο αλγόριθμος βελτιστοποίησης Big Bang-Big Crunch (BB-BC) είναι μια νέα μέθοδος βελτιστοποίησης εμπνευσμένη από τη φύση, που στηρίζεται στη θεωρία Big Bang και Big Crunch, μία από τις θεωρίες της εξέλιξης του σύμπαντος. Σύμφωνα με αυτή τη θεωρία, η μεγάλη έκρηξη παράγει διαταραχή και τυχαιότητα, τα οποία είναι και τα βασικά χαρακτηριστικά του, ενώ στη φάση Big Crunch τυχαία κατανενημένα σωματίδια έλκονται από ένα σημείο. Στην πράξη ο BB-BC παράγει τυχαία σημεία στη φάση του Big Bang και συρρικνώνει αυτά τα σημεία σε ένα ενιαίο σημείο μέσω ενός κέντρου μάζας στη φάση του Big Crunch. Μετά από διαδοχικές φάσεις του Big Bang και Big Crunch, η τυχαιότητα αυτή της αναζήτησης στο χώρο του αλγορίθμου γίνεται ολοένα και μικρότερη και ο αλγόριθμος συγκλίνει σε μία λύση.

Η δημιουργία του αρχικού πληθυσμού ονομάζεται Big Bang. Έτσι οι λύσεις διαδίδονται σε όλο το χώρο αναζήτησης με ομοιόμορφη κατανομή. Η επόμενη φάση που ακολουθεί είναι η Big Crunch η οποία είναι ένας τελεστής σύγκλισης που έχει πολλά εισερχόμενα στοιχεία αλλά ένα μόνο εξερχόμενο, το οποίο ονομάζουμε κέντρο μάζας. Το σημείο αυτό αναφέρεται ως $\vec{x}^c$ και υπολογίζεται ως εξής:

$$\vec{x}^c = \frac{\sum_{i=1}^{N} \frac{1}{f^i} \vec{x}^i}{\sum_{i=1}^{N} \frac{1}{f^i}} \qquad 2.1$$

όπου x είναι σημείο στο χώρο των λύσεων, f^i είναι η τιμή της αντικειμενικής συνάρτησης της λύσης αυτής και N είναι το μέγεθος του πληθυσμού. Μετά τη φάση του Big Crunch ο αλγόριθμος δημιουργεί νέες λύσεις ώστε να χρησιμοποιηθούν για τη φάση του Big Bang της επόμενης επανάληψης. Οι νέες αυτές λύσεις υπολογίζονται από την ακόλουθη εξίσωση:

$$x_i^{new} = x^c + \sigma \qquad 2.2$$

όπου η τιμή του σ υπολογίζεται ως ακολούθως:

$$\sigma = \frac{r\alpha(x_{max} - x_{min})}{k} \qquad 2.3$$

όπου

$$\begin{cases} x_{min} = x^{gbest} - 0.5 \times n(overall\ search\ space) \\ x_{max} = x^{gbest} + 0.5 \times n(overall\ search\ space) \end{cases}$$

όπου α είναι μία παράμετρος, r είναι ένας τυχαίος αριθμός που παράγει μια γεννήτρια αριθμών κανονικής κατανομής (διαφορετικός για κάθε λύση), x_{min} και x_{max} είναι η ελάχιστη και η μέγιστη τιμή αντίστοιχα, των τιμών των διανυσμάτων των λύσεων, και k είναι ο αριθμός των επαναλήψεων που βρίσκεται ο αλγόριθμος. Μετά τη δεύτερη έκρηξη (Big Bang) το κέντρο μάζας υπολογίζεται πάλι σύμφωνα με την προηγούμενη εξίσωση. Αυτά τα δύο βήματα, δηλαδή το Big Bang και το Big Crunch επαναλαμβάνονται μέχρι να ικανοποιηθεί το κριτήριο τερματισμού του αλγορίθμου. Τελικά, χρησιμοποιείται μία άλλη σχέση που καταγράφεται παρακάτω για τον υπολογισμό των νέων θέσεων των διανυσμάτων προκειμένου να βελτιωθεί η αποτελεσματικότητα του αλγόριθμου. Η σχέση αυτή χρησιμοποιεί την κανονική κατανομή, γύρω από ένα νέο σημείο, ανάμεσα στο κέντρο μάζας και της καλύτερης λύσης χρησιμοποιώντας την ακόλουθη εξίσωση:

$$x^{(k+1,i)} = \beta x^{c(k)} + (1-\beta)x^{gbest(k)} + \sigma \qquad 2.4$$

όπου β είναι η παράμετρος της θέσης των νέων λύσεων, $x^{c(k)}$ το κέντρο μάζας, όπως υπολογίζεται παραπάνω, και $x^{gbest(k)}$ είναι το διάνυσμα θέσης της ολικά καλύτερης λύσης. Έρευνες έχουν δείξει ότι υπάρχει σημαντική βελτίωση της ποιότητας των λύσεων και της αποτελεσματικότητας του αλγορίθμου χρησιμοποιώντας την παραπάνω εξίσωση συνδυαστικά με τις προηγούμενες. Κατά μία έννοια, θα μπορούσαμε να ισχυριστούμε ότι ο σταθμισμένος μέσο όρος των $x^{c(k)}$ και $x^{gbest(k)}$ με συντελεστές στάθμισης β και $1-\beta$, μπορεί να θεωρηθεί σαν τεχνική ισοδύναμη με μια ελιτιστική στρατηγική όπου η βέλτιστη λύση επηρεάζει την κατεύθυνση της αναζήτησης στο χώρο των λύσεων.

Παρακάτω παρουσιάζονται τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα:

Αλγόριθμος της μεγάλης έκρηξης

Βήμα 1: Καθορισμός παραμέτρων
Βήμα 2: Αρχικοποίηση πληθυσμού
Βήμα 3: Υπολογισμός των τιμών της αντικειμενικής συνάρτησης για κάθε μέλος του πληθυσμού
Βήμα 4: Υπολογισμός του κέντρου μάζας
Βήμα 5: Υπολογισμός των νέων λύσεων
Βήμα 6: Επανάλαβε τα βήματα 3 – 5 μέχρι να ικανοποιηθεί το κριτήριο τερματισμού

2.3.3 Ο αλγόριθμος βελτιστοποίησης του γκρίζου λύκου (Grey Wolf Optimization Algorithm, GWO)

Ο αλγόριθμος βελτιστοποίησης του γκρίζου λύκου είναι μια νέα μεθευρετική μέθοδος βελτιστοποίησης εμπνευσμένη από τους γκρίζους λύκους. Οι γκρίζοι λύκοι θεωρούνται θηρευτές, που σημαίνει ότι βρίσκονται στην κορυφή της τροφικής αλυσίδας. Οι γκρίζοι λύκοι συνήθως ζουν σε αγέλες όπου το μέγεθος της κάθε αγέλης ανέρχεται από 5 ως 12 άτομα κατά μέσο όρο. Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός ότι υπάρχει μια πολύ αυστηρή κοινωνική ιεραρχία. Η ηγεσία της αγέλης αποτελείται από ένα αρσενικό και από ένα θηλυκό λύκο όπου ονομάζονται άλφα (α). Οι άλφα παίρνουν αποφάσεις σχετικά με το κυνήγι, το χρόνο που κοιμούνται, τι ώρα σηκώνονται κτλ. Η αγέλη υπακούει στις αποφάσεις των άλφα. Εντούτοις, μερικές φορές παρουσιάζεται δημοκρατική συμπεριφορά, στην οποία ένας άλφα ακολουθεί τους άλλους λύκους στην αγέλη. Στις συγκεντρώσεις, ολόκληρη η αγέλη αναγνωρίζει τους άλφα κρατώντας την ουρά τους προς τα κάτω. Ο άλφα λύκος συχνά αποκαλείται και κυρίαρχος λύκος καθώς στις εντολές του πρέπει να υπακούει ολόκληρη η αγέλη. Ο άλφα λύκος δεν είναι αναγκαστικά το ισχυρότερο μέλος, αλλά το καλύτερο, όσον αφορά τη διαχείριση της αγέλης. Αυτό δείχνει ότι η οργάνωση και η πειθαρχία της αγέλης είναι πολύ πιο σημαντική από τη δύναμή της. Στο δεύτερο επίπεδο της ιεραρχίας ανήκουν ο βήτα λύκος (β). Οι βήτα είναι υποτελείς λύκοι όπου έχουν βοηθητικό χαρακτήρα στους άλφα στη λήψη αποφάσεων. Ο βήτα είναι είτε αρσενικός είτε θηλυκός λύκος και αποτελεί το διάδοχο του άλφα σε περίπτωση που πεθάνει ή γεράσει αρκετά. Ο βήτα πρέπει να σέβεται τον άλφα και μπορεί να δίνει εντολές σε κατώτερα μέλη. Συνήθως έχει συμβουλευτικό χαρακτήρα στον άλφα και πειθαρχεί την αγέλη. Η χαμηλότερη κατάταξη περιλαμβάνει τον ωμέγα λύκο (ω) ο οποίος έχει το ρόλο του αποδιοπομπαίου τράγου. Αυτός ο λύκος υποτάσσεται σε όλους τους άλλους κυρίαρχους και είναι ο τελευταίος που θα φάει. Φαινομενικά, ο ωμέγα λύκος δεν έχει κάποια σημαντικότητα όμως έχει παρατηρηθεί ότι στην αγέλη παρουσιάζονται προβλήματα όταν υπάρχει απουσία του ωμέγα. Αυτό οφείλεται στο ξέσπασμα της βίας των άλλων ατόμων προς τους ωμέγα και βοηθά στην ικανοποίηση των κυρίαρχων ατόμων και στη διατήρηση της κυριαρχίας. Τέλος, αν ένας λύκος δεν είναι άλφα, βήτα ή ωμέγα, ονομάζεται δευτερεύον ή δέλτα (δ) σε ορισμένες περιπτώσεις. Ο δέλτα πρέπει να υποτάσσεται στον άλφα και βήτα, αλλά κυριαρχεί στον ωμέγα. Σε αυτήν την κατηγορία ανήκουν λύκοι ανιχνευτές, προστάτες, ηλικιωμένοι, κυνηγοί και λύκοι που βοηθάν άλλους.

Εκτός από την ιεραρχία των λύκων, μία ακόμα ενδιαφέρουσα κοινωνική συμπεριφορά που παρουσιάζεται σε αγέλες λύκων είναι ότι κυνηγούν ομαδικά. Οι κύριες φάσεις του κυνηγιού είναι οι εξής:

- Παρακολουθούν, κυνηγούν και προσεγγίζουν το θήραμα.
- Περικυκλώνουν το θήραμα μέχρι να σταματήσει.
- Επιτίθενται στο θήραμα.

Αυτή η συμπεριφορά των γκρίζων λύκων καθώς κυνηγούν καθώς και η κοινωνική ιεραρχία τους μοντελοποιείται προκειμένου να υλοποιηθεί ο αλγόριθμος. Έτσι λοιπόν θεωρούμε την βέλτιστη λύση ως τον άλφα λύκο (α) και, τη δεύτερη και τρίτη καλύτερη λύση τον βήτα λύκο (β) και δέλτα λύκο (δ) αντίστοιχα. Οι υπόλοιπες λύσεις αντιστοιχούν στον ωμέγα (ω). Όπως προαναφέρθηκε οι γκρίζοι λύκοι περικυκλώνουν το θήραμα τους καθώς κυνηγούν. Αυτή η συμπεριφορά περιγράφεται με την παρακάτω σχέση:

$$\vec{D} = \left|\vec{C} \cdot \vec{x}_p(t) - \vec{x}(t)\right| \quad 3.1$$

$$\vec{x}(t+1) = \vec{x}_p(t) - \vec{A} \cdot \vec{D} \quad 3.2$$

όπου t συμβολίζει την τρέχουσα επανάληψη, το $\vec{A}$ και το $\vec{C}$ είναι διανύσματα, το $\vec{x}_p$ είναι το διάνυσμα θέσης τους θηράματος και το $\vec{x}$ είναι το διάνυσμα θέσης του λύκου. Τα διανύσματα $\vec{A}$ και $\vec{C}$ υπολογίζονται ως ακολούθως:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad 3.3$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad 3.4$$

όπου τα στοιχεία του α μειώνονται γραμμικά από το 2 στο 0 με το πέρασμα των επαναλήψεων και, $\vec{r}_1$ και $\vec{r}_2$ τυχαία διανύσματα στο διάστημα [0,1]. Εδώ πρέπει να σημειωθεί ότι τα τυχαία διανύσματα $\vec{r}_1$ και $\vec{r}_2$ επιτρέπουν στα άτομα να φτάσουν σε οποιοδήποτε σημείο στο χώρο μέσω των εξισώσεων 3.1 και 3.2.

Οι γκρίζοι λύκοι έχουν τη δυνατότητα να αναγνωρίζουν τη θέση του θηράματος και να το περικυκλώνουν. Το κυνήγι συνήθως καθοδηγείται από τον άλφα και τον βήτα. Ο δέλτα ενδέχεται επίσης να συμμετέχει στο κυνήγι περιστασιακά. Ωστόσο σε ένα τυχαίο χώρο αναζήτησης δεν μπορούμε να έχουμε γνώση για τη θέση της λείας. Για αυτό το λόγο, προκειμένου να προσομοιώσουμε μαθηματικά την συμπεριφορά αυτή των ατόμων υποθέτουμε ότι ο άλφα, δηλαδή βέλτιστη λύση, ο βήτα και ο δέλτα γνωρίζουν καλύτερα το πού βρίσκεται το θήραμα. Επομένως αποθηκεύουμε τις τρεις πρώτες καλύτερες λύσεις που υπάρχουν μέχρι στιγμής και οι θέσεις των υπολοίπων ατόμων, συμπεριλαμβανομένων των ωμέγα, ανανεώνονται σύμφωνα με τις θέσεις των τριών καλύτερων ατόμων. Έτσι λοιπόν προτείνονται οι παρακάτω εξισώσεις για τον υπολογισμό των θέσεων :

$$\vec{D}_a = \left|\vec{C_1} \cdot \vec{x}_a - \vec{x}\right|, \vec{D}_\beta = \left|\vec{C_2} \cdot \vec{x}_\beta - \vec{x}\right|, \vec{D}_\delta = \left|\vec{C_3} \cdot \vec{x}_\delta - \vec{x}\right| \quad 3.5$$

$$\vec{x}_1 = \vec{x}_\alpha - A_1 \cdot (\vec{D}_\alpha),\ \vec{x}_2 = \vec{x}_\beta - A_2 \cdot (\vec{D}_\beta),\ \vec{x}_3 = \vec{x}_\delta - A_3 \cdot (\vec{D}_\delta) \quad 3.6$$

$$\vec{x}(t+1) = \frac{\vec{x}_1 + \vec{x}_2 + \vec{x}_3}{3} \qquad 3.7$$

Όπως προαναφέρθηκε οι γκρίζοι λύκοι τελειώνουν το κυνήγι με την επίθεση στο θήραμα τους. Προκειμένου να μοντελοποιηθεί αυτή η διαδικασία η τιμή του α μπορεί να μειώνεται σε διαφορετικές επαναλήψεις. Εδώ πρέπει να σημειωθεί ότι ταυτόχρονα με το α μειώνεται επίσης και η διακύμανση του A. Με άλλα λόγια, το A είναι μία τυχαία τιμή στο διάστημα [2α, 2α], όπου το α μειώνεται από το 2 στο 0 με το πέρασμα των επαναλήψεων όταν η τυχαία τιμή του α βρίσκεται στο διάστημα αυτό. Όταν η τυχαία τιμή του A βρίσκεται στο διάστημα [-1, 1], η θέση της επόμενης επανάληψης ενός λύκου μπορεί να είναι ανάμεσα στην τρέχουσα θέση και στη θέση της λείας.
Παρακάτω παρουσιάζονται τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα:

Αλγόριθμος του γκρίζου λύκου

Αρχικοποίηση του αρχικού πληθυσμού
Αρχικοποίηση του α, A και C
Υπολογισμός της αντικειμενικής συνάρτησης κάθε ατόμου
Καθορισμός $\vec{x}_\alpha$, $\vec{x}_\beta$ και $\vec{x}_\delta$
for i=1,
    Υπολογισμός νέας θέσης κάθε ατόμου μέσω της σχέσης 3.7
    Υπολογισμός του α, A και C
    Υπολογισμός της νέας τιμής της αντικειμενικής συνάρτησης κάθε ατόμου
    Υπολογισμός των νέων διανυσμάτων θέσης $\vec{x}_\alpha$, $\vec{x}_\beta$ και $\vec{x}_\delta$
end

2.3.4 Ο αλγόριθμος των μαϊμούδων (Spider Monkey Optimization, SMO)

Μια κοινωνία διάσπασης-σύντηξης είναι μια κοινωνία στην οποία το μέγεθος και η σύνθεση της κοινωνικής ομάδας μεταβάλλεται με την πάροδο του χρόνου όπου τα ζώα μετακινούνται στο περιβάλλον. Δηλαδή συγχωνεύονται σε μια ομάδα (συγχώνευση) όπως για παράδειγμα όταν κοιμούνται σε ένα μέρος και χωρίζονται (σχάση) όπως όταν ψαρεύουν σε μικρές ομάδες κατά τη διάρκεια της ημέρας. Η σύνθεση των ομάδων των ζώων που ζουν σε τέτοιες κοινωνίες, χαρακτηρίζεται από μια δυναμικότητα. Τέτοια είδη κοινωνικών συστημάτων μπορούν να ελαχιστοποιήσουν τον ανταγωνισμό των μελών που αφορά την αναζήτηση της τροφής, για αυτό και διαιρούνται σε μικρότερες ομάδες προκειμένου να ψάξουν γι' αυτή. Τα μέλη της ομάδας αλληλεπιδρούν μεταξύ τους προκειμένου να διατηρήσουν κοινωνικούς δεσμούς και εδαφικά όρια. Σε τέτοιες κοινωνίες όλα τα άτομα κοιμούνται σε ένα μέρος μαζί αλλά αναζητούν τροφή χωρισμένα σε υποομάδες και πηγαίνοντας προς διαφορετικές κατευθύνσεις κατά τη διάρκεια της ημέρας. Αυτή η μορφή κοινωνικής διαμόρφωσης εμφανίζεται σε διάφορα είδη πρωτευόντων θηλαστικών όπως οι χιμπατζήδες, οι μπαμπουίνοι, οι μπονόμπο και οι μαϊμούδες αράχνη. Αυτές οι κοινωνίες μεταβάλλουν συχνά το μέγεθος και τη σύνθεση τους δημιουργώντας μία ισχυρή κοινωνική ομάδα που ονομάζεται μητρική ομάδα. Σε μία τέτοια κοινωνία η κυρία μητρική ομάδα μπορεί να διαχωριστεί σε μικρότερες υποομάδες ή άτομα για να προσαρμοστεί στις περιβαλλοντικές ή κοινωνικές συνθήκες. Για παράδειγμα μέλη μιας ομάδας διαχωρίζονται από την κύρια ομάδα προκειμένου να κυνηγήσουν ή να αναζητήσουν τροφή κατά τη διάρκεια της ημέρας όμως τη νύχτα επιστρέφουν στην αρχική ομάδα για να μοιραστούν την τροφή και να πάρουν μέρος σε άλλες δραστηριότητες.

Οι μαϊμούδες αράχνη ανήκουν σε αυτήν την κατηγορία κοινωνίας που μόλις περιγράφηκε. Είναι κοινωνικά ζώα και ζουν σε ομάδες μέχρι 50 ατόμων. Βρίσκουν την τροφή τους με πολύ διαφορετικό τρόπο, καθώς ένα θηλυκό καθοδηγεί την ομάδα και είναι υπεύθυνο γι' αυτήν. Σε περίπτωση που δεν βρει επαρκή ποσότητα τροφής διαιρεί την ομάδα σε μικρότερες προκειμένου να ψάξει η κάθε μία μικρότερη ομάδα χωριστά. Οι υποομάδες που δημιουργούνται δεν είναι μόνιμες αλλά προσωρινές, κι επίσης μπορούν να διαφέρουν κατά τη διάρκεια της ημέρας. Όταν πλησιάσουν δύο διαφορετικές ομάδες παρατηρείται επιθετικότητα στα αρσενικά κι αυτό εκδηλώνεται με γρυλίσματα και φωνές, οι οποίες εμφανίζονται σε μεγάλη απόσταση μεταξύ των δύο υποομάδων και δεν συνεπάγονται καμία φυσική επαφή. Αυτό δείχνει το σεβασμό και των δύο στα διακριτά σύνορα κάθε ομάδας. Ο κύριος λόγος δημιουργίας τέτοιου είδους κοινωνικού συστήματος είναι ο ανταγωνισμός των μελών όταν υπάρχει έλλειψη τροφίμων λόγω εποχής. Όταν μια μεγάλη ομάδα διαθέτει φαγητό σε μια συγκεκριμένη τοποθεσία, είναι πιθανό να υπάρχει λιγότερη τροφή ανά μέλος της ομάδας σε σύγκριση με μια μικρότερη ομάδα. Όταν η έλλειψη τροφής είναι μεγάλη, το μέσο μέγεθος της υποομάδας είναι μικρότερο συγκριτικά με περιόδους όπου υπάρχει μεγαλύτερη διαθεσιμότητα τροφίμων. Αυτό υποδηλώνει ότι ο ανταγωνισμός για την απόκτηση των πόρων όταν είναι περισσότερο δυσεύρετοι απαιτεί υποδιαίρεση των ομάδων. Ένας ακόμα λόγος που γίνεται αυτό το σπάσιμο σε ομάδες αναζήτησης τροφής αλλά παρόλα αυτά ανήκουν σε μία μεγαλύτερη κοινωνικά κύρια ομάδα είναι το πλεονέκτημα των μελών να ζευγαρώνουν ευκολότερα και να έχουν περισσότερη ασφάλεια από εξωτερικές απειλές. Οι μαϊμούδες αράχνες εκδηλώνουν τις προθέσεις τους και επικοινωνούν γενικότερα μέσω στάσεων όπως στάσεις σεξουαλικής δεκτικότητας και επίθεσης. Κατά τη διάρκεια ταξιδιών, όταν διάφορα μέλη βρίσκονται σε μεγάλη απόσταση μεταξύ τους, αλληλεπιδρούν χρησιμοποιώντας μια κραυγή που μοιάζει με το χλιμίντρισμα του αλόγου. Κάθε άτομο έχει το δικό του χαρακτηριστικό ήχο ώστε τα υπόλοιπα μέλη να μπορούν να το αναγνωρίσουν. Αυτή η επικοινωνία εξ’ αποστάσεως επιτρέπει στα μέλη να μένουν μακριά από τους εχθρούς, να μοιράζονται φαγητό και να συμμετέχουν σε διαλόγους.

Ο αλγόριθμος αυτός είναι εμπνευσμένος από την κοινωνική συμπεριφορά των μαϊμούδων αραχνών και προσομοιώνει την συμπεριφορά αναζήτησης τροφής τους. Η συμπεριφορά αυτή δείχνει ότι οι μαϊμούδες αράχνες ανήκουν στην κατηγορία της κοινωνικής δομής που αναφέρθηκε παραπάνω όπου τα μέλη διασπώνται και ενώνονται. Ο αλγόριθμος αποτελείται από έξι φάσεις :

- Local Leader phase
- Global Leader phase
- Local Leader Learning phase
- Global Leader Learning phase
- Local Leader Decision phase
- Global Leader Decision phase

Αρχικοποίηση πληθυσμού

Ο αλγόριθμος δημιουργεί έναν ομοιόμορφα κατανεμημένο αρχικό πληθυσμό N πιθήκων όπου κάθε πίθηκος SM_i (i =1, 2, ..., N) είναι ένα διάνυσμα D διαστάσεων. D είναι ο αριθμός των μεταβλητών απόφασης ενώ το SM_i αντιπροσωπεύει τον i πίθηκο στον πληθυσμό. Κάθε άτομο του πληθυσμού αποτελεί και μία λύση του προβλήματος και αρχικοποιείται ως εξής :

$$SM_{ij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj}) \quad 4.1$$

όπου $U(0,1)$ τυχαίος αριθμός ομοιόμορφα κατανεμημένος στο διάστημα [0,1] και, SM_{minj} και SM_{maxj} τα όρια του i ατόμου στην j διεύθυνση.

Local Leader Phase (LLP)

Σε αυτή τη φάση, το άτομο τροποποιεί την τρέχουσα θέση του με βάση τον τοπικό ηγέτη. Υπολογίζεται η τιμή της αντικειμενικής συνάρτησης κι αν η νέα τιμή είναι καλύτερη της προηγούμενης τότε το διάνυσμα θέσης αντικαθίσταται από το καινούριο. Η εξίσωση για την ανανέωση του διανύσματος θέσης είναι:

$$SM_{newij} = SM_{ij} + U(0,1) \times \left(LL_{kj} - SM_{ij}\right) + U(-1,1) \times (SM_{rj} - SM_{ij}) \quad 4.2$$

όπου SM_{ij} είναι η j διάσταση του i ατόμου, LL_{kj} είναι η j διάσταση του k τοπικού ηγέτη ομάδας. Το SM_{rj} είναι η j διάσταση του k ατόμου που επιλέγεται τυχαία μέσα στην k ομάδα έτσι ώστε $\mathrm{r} \neq \mathrm{i}$, $U(0,1)$ είναι ένας ομοιόμορφα κατανεμημένος τυχαίος αριθμός στο διάστημα [0,1]. Παρακάτω παρουσιάζονται τα βήματα της διαδικασίας σε μορφή ψευδοκώδικα. Όπου pr είναι ο ρυθμός διακύμανσης όπου ελέγχει τη διακύμανση στην τρέχουσα θέση και ανήκει στο διάστημα [0.1,0.8].

Αλγόριθμος1 Ενημέρωση θέσης της φάσης Local Leader Phase

for each member $SM_i \in k^{th}$ group
  for each j ∈ {1,...,D}
    if $U(0,1) >$pr
      $SM_{newij} = SM_{ij} + U(0,1) \times \left(LL_{kj} - SM_{ij}\right) + U(-1,1) \times (SM_{rj} - SM_{ij})$
    else
      $SM_{newij} = SM_{ij}$
    end if
  end for
end for

Global Leader Phase (GLP)

Σε αυτή τη φάση όλα τα μέλη του πληθυσμού ανανεώνουν τη θέση τους λαμβάνοντας υπόψη τον αρχηγό της ομάδας και τα υπόλοιπα μέλη. Η εξίσωση που ενημερώνει το διάνυσμα θέσης είναι η ακόλουθη:

$$SM_{newij} = SM_{ij} + U(0,1) \times \left(GL_j - SM_{ij}\right) + U(-1,1) \times (SM_{rj} - SM_{ij}) \qquad 4.3$$

όπου GL_j είναι η j διάσταση του διανύσματος θέσης του αρχηγού της ομάδας, όπου $j \in \{1,2,\dots,D\}$ και παίρνει μία τυχαία τιμή δείκτη. Σε αυτή τη φάση το διάνυσμα θέσης κάθε μέλους ανανεώνεται βάσει μιας πιθανότητας $prob_i$ που υπολογίζεται μέσω των αντικειμενικών συναρτήσεων. Έτσι, η καλύτερη υποψήφια λύση έχει περισσότερες πιθανότητες να βελτιωθεί. Παρακάτω παρουσιάζεται η σχέση της πιθανότητας $prob_i$:

$$prob_i = \frac{fitness_i}{\sum_{i=1}^{N} fitness_i} \qquad 4.4$$

όπου $fitness_i$ είναι η τιμή της αντικειμενικής συνάρτησης του i ατόμου. Έπειτα υπολογίζεται η νέα τιμή της αντικειμενικής συνάρτησης και υιοθετείται η καλύτερη λύση. Παρακάτω παρουσιάζεται η διαδικασία αυτή σε μορφή ψευδοκώδικα.

Αλγόριθμος2 Ενημέρωση θέσης της φάσης Global Leader Phase (GLP)

$count = 0$
while $count < group\ size$
    for each member SM_i∈ group
        if $U(0,1) < prob_i$
            $count = count + 1$
            Randomly select $j \in \{1,2,\dots,D\}$
            Randomly select SM_r∈ group s.t. r ≠ i.
            $SM_{newij} = SM_{ij} + U(0,1) \times \left(GL_j - SM_{ij}\right) + U(-1,1) \times (SM_{rj} - SM_{ij})$
        end if
    end for
end while

Global Leader Learning (GLL) Phase

Εδώ, το διάνυσμα θέσης του αρχηγού ενημερώνεται μέσω της βέλτιστης επιλογής στον συνολικό πληθυσμό, δηλαδή η λύση του ατόμου με την καλύτερη τιμή αντικειμενικής συνάρτησης στον πληθυσμό. Έπειτα ελέγχεται αν θα γίνει αντικατάσταση της λύσης, αν δηλαδή η νέα τιμή της αντικειμενικής συνάρτησης είναι καλύτερη από την προηγούμενη. Στην περίπτωση που δεν είναι, ο μετρητής $GlobalLimitCount$ αυξάνεται κατά ένα.

Local Leader Learning (LLL) Phase

Σε αυτή τη φάση, η θέση του τοπικού ηγέτη ενημερώνεται με την εφαρμογή της βέλτιστης επιλογής λύσης στη συγκεκριμένη ομάδα. Στη συνέχεια, υπολογίζεται η τιμή της αντικειμενικής συνάρτησης, συγκρίνεται με την παλιά και εάν ο τοπικός ηγέτης δεν αντικατασταθεί από τη νέα λύση ο μετρητής $LocalLimitCount$ αυξάνεται κατά ένα.

Local Leader Decision (LLD) phase

Εάν κάποια θέση τοπικού ηγέτη δεν ενημερωθεί μέχρι έναν προκαθορισμένο αριθμό επαναλήψεων (μετρητής που ονομάζεται *LocalLeaderLimit*), τότε όλα τα μέλη αυτής της ομάδας ενημερώνουν τις θέσεις τους είτε με τυχαία αρχικοποίηση είτε με τη χρήση πληροφοριών από τον κύριο ηγέτη και από τους τοπικούς μέσω της παρακάτω εξίσωσης, με βάση τη διακύμανση pr:

$$SM_{newij} = SM_{ij} + U(0,1) \times \left(GL_j - SM_{ij}\right) + U(0,1) \times (SM_{ij} - LL_{kj}) \qquad 4.5$$

Είναι σαφές από την εξίσωση ότι το νέο διάνυσμα θέσης αυτού του ατόμου προσελκύεται από τον συνολικό ηγέτη και απωθείται από τον τοπικό ηγέτη της ομάδας. Η διαδικασία ενημέρωσης θέσης της φάσης αυτής παρουσιάζεται σε μορφή βημάτων στον αλγόριθμο παρακάτω:

Αλγόριθμος3 Local Leader Decision (LLD) phase

if $LocalLimitCount > LocalLeaderLimit$
    $LocalLimitCount = 0$
        for each $j \in \{1,2,\dots,D\}$
            if $U(0,1) \geq pr$
                $SM_{ij} = SM_{minj} + U(0,1) \times \left(SM_{maxj} - SM_{minj}\right)$
            else
                $SM_{newij} = SM_{ij} + U(0,1) \times \left(GL_j - SM_{ij}\right) + U(0,1) \times (SM_{ij} - LL_{kj})$
            end if
    end for
end if

Global Leader Decision (GLD) phase

Σε αυτή τη φάση, η θέση του συνολικού ηγέτη παρακολουθείται για ένα συγκεκριμένο αριθμό επαναλήψεων (*GlobalLeaderLimit*) και αν δεν ενημερωθεί, ο ηγέτης διαιρεί τον πληθυσμό σε μικρότερες ομάδες. Η διαδικασία παρουσιάζεται στον αλγόριθμο παρακάτω σε μορφή βημάτων:

Αλγόριθμος4 Global Leader Decision (GLD) phase

```
if GlobalLimitCount > GlobalLeaderLimit
    GlobalLimitCount = 0
    if Number of groups < MG
        Divide the population into groups
    else
        Combine all the groups to make a single group
    end if
Update Local Leaders position
end if
```

Ακολουθεί ο ψευδοκώδικας του γενικού αλγόριθμου:

Αλγόριθμος των μαϊμούδων

```
Αρχικοποίηση πληθυσμού, LocalLeaderLimit, GlobalLeaderLimit και pr
Υπολογισμός αντικειμενικής συνάρτησης όλων των μελών
Ορισμός κύριου ηγέτη και τοπικών ηγετών
for i=1,...
        Εφαρμογή αλγορίθμου1
        Εφαρμογή αλγορίθμου2
        Υπολογισμός της αντικειμενικής συνάρτησης για τις δύο νέες ομάδες
        πληθυσμού
        Σύγκριση των τριών τιμών της ατικειμενικής συνάρτησης και διατήρηση
        της βέλτιστης για  κάθε άτομο του πληθυσμού
        Ενημέρωση GlobalLeader
        Ενημέρωση LocalLeader κάθε ομάδας
        Εφαρμογή αλγορίθμου3
        Εφαρμογή αλγορίθμου4
end for
```

2.3.5 Ο αλγόριθμος της φάλαινας (The Whale Optimization Algorithm, WO)

Ο εμπνευσμένος αυτός αλγόριθμος από τη φύση μιμείται την κοινωνική συμπεριφορά των φαλαινών και προσομοιώνει το ομαδικό τρόπο που τρέφονται χρησιμοποιώντας φυσαλίδες. Οι φάλαινες θεωρούνται ως τα μεγαλύτερα θηλαστικά στον κόσμο. Μια ενήλικη φάλαινα μπορεί να φτάσει μέχρι 30 μέτρα μήκος και 180 τόνους βάρους. Υπάρχουν διάφορα κύρια είδη αυτού του γιγαντιαίου θηλαστικού. Η τρέχουσα εργασία ασχολείται συγκεκριμένα με την μεγάπτερη φάλαινα. Οι φάλαινες θεωρούνται αρπακτικά ζώα. Δεν κοιμούνται ποτέ επειδή πρέπει να αναπνέουν από την επιφάνεια των ωκεανών. Στην πραγματικότητα, το ήμισυ του εγκεφάλου κοιμάται μόνο. Το ενδιαφέρον στοιχείο των φαλαινών είναι ότι θεωρούνται εξαιρετικά ευφυή ζώα με συγκινησιακό παράγοντα. Σύμφωνα με τους Hof και Van Der Gucht, οι φάλαινες, σε ορισμένες περιοχές του εγκεφάλου τους, έχουν παρόμοια κύτταρα με εκείνα των ανθρώπων. Αυτό το είδος των κυττάρων ευθύνονται για την κρίση, τα συναισθήματα και τις κοινωνικές συμπεριφορές στους ανθρώπους. Με άλλα λόγια ο τύπος αυτός του κυττάρου μας κάνουν να ξεχωρίζουμε από τα υπόλοιπα πλάσματα. Οι φάλαινες έχουν διπλάσιο αριθμό τέτοιων κυττάρων από έναν ενήλικα άνθρωπο που είναι η κύρια αιτία της εξυπνάδας. Έχει αποδειχθεί ότι η φάλαινα μπορεί να σκεφτεί, να μαθαίνει, να κρίνει, να επικοινωνήσει και να αναπτύσσει συναισθήματα όπως ο άνθρωπος, αλλά προφανώς με πολύ χαμηλότερο επίπεδο νοημοσύνης. Έχει παρατηρηθεί ότι οι φάλαινες (κυρίως φάλαινες δολοφόνοι) είναι σε θέση να αναπτύξουν τη δική τους διάλεκτο. Ένα άλλο ενδιαφέρον σημείο είναι η κοινωνική συμπεριφορά των φαλαινών. Ζουν ατομικά ή σε ομάδες. Ωστόσο, παρατηρείται κυρίως η ομαδικότητα. Μερικά από τα είδη τους, όπως οι φάλαινες δολοφόνοι, μπορούν να ζήσουν σε μια οικογένεια καθ’ όλη τη διάρκεια της ζωής τους.

Το πιο ενδιαφέρον πράγμα για τη φάλαινα που μελετάται είναι ξεχωριστή μέθοδος που τρέφονται. Οι φάλαινες καταδύονται περίπου 12 μέτρα κάτω από την επιφάνεια της θάλασσας. Στη συνέχεια αρχίζουν να δημιουργούν φυσαλίδες σε σπειροειδή μορφή (σχηματίζοντας νοητά τον αριθμό 9) γύρω από το θήραμα, δηλαδή τα ψάρια, για να τρομάξουν και να μαζευτούν όλα μαζί. Ύστερα αναδύονται στο κέντρο του κύκλου των φυσαλίδων με το στόμα ανοιχτό και καταπίνουν μία μεγάλη ποσότητα ψαριών. Σε αυτό τον αλγόριθμο, μοντελοποιείται ο ελιγμός αυτός όπως έχει περιγραφεί.

Η μεγάπτερη φάλαινα μπορεί να αναγνωρίσει τη θέση του θηράματος και να το περικυκλώσει. Δεδομένου ότι η θέση της βέλτιστης λύσης στο χώρο δεν είναι γνωστή εκ των προτέρων στον αλγόριθμο αυτό, θεωρούμε ότι η τρέχουσα καλύτερη λύση είναι η θέση της λείας ή τουλάχιστον ότι είναι κοντά στο βέλτιστο. Αυτό περιγράφεται από τις ακόλουθες εξισώσεις:

$$\vec{D} = \left|\vec{C}.\vec{x}^*(t) - \vec{x}(t)\right| \qquad 5.1$$

$$\vec{x}(t+1) = \vec{x}^*(t) - \vec{A} \cdot \vec{D} \qquad 5.2$$

όπου t είναι ο αριθμός της τρέχουσας επανάληψης, $\vec{A}$ και $\vec{C}$ είναι τα διανύσματα των συντελεστών, $\vec{x}^*$ είναι το διάνυσμα της βέλτιστης θέσης μέχρι την τρέχουσα στιγμή, $\vec{x}$ είναι το διάνυσμα θέσης, || είναι η απόλυτη τιμή και . είναι ο πολλαπλασιασμός στοιχείο με στοιχείο. Τα διανύσματα $\vec{A}$ και $\vec{C}$ υπολογίζονται ως ακολούθως :

$$\vec{A} = 2\vec{a} \cdot \vec{r} – \vec{a} \qquad 5.3$$

$$\vec{C} = 2 \cdot \vec{r} \qquad 5.4$$

όπου το α μειώνεται από το 2 στο 0 με το πέρασμα των επαναλήψεων και το r είναι ένα διάνυσμα με τυχαίες τιμές στο διάστημα [0,1].

Παρακάτω μοντελοποιείται μαθηματικά η δημιουργία των φυσαλίδων μέσω δύο προσεγγίσεων :

α) Περικύκλωση μειούμενης διαμέτρου : Αυτή η συμπεριφορά επιτυγχάνεται μειώνοντας την τιμή του $\vec{a}$ στην εξίσωση 5.3. Να σημειωθεί ότι η διακύμανση του $\vec{A}$ μειώνεται καθώς μειώνεται και το $\vec{a}$. Με άλλα λόγια, το $\vec{A}$ είναι μια τυχαία τιμή στο διάστημα [-α, α] όπου $\vec{a}$ το μειώνεται από το 2 στο 0 με το πέρασμα των επαναλήψεων. Καθορίζοντας τυχαίες τιμές για το $\vec{A}$ στο [-1,1], η νέα θέση ενός ατόμου μπορεί να οριστεί οπουδήποτε μεταξύ της αρχικής θέσης του ατόμου και της θέσης της τρέχουσας καλύτερης λύσης.

β) Σπειροειδής κίνηση : Η προσέγγιση αυτή υπολογίζει πρώτα την απόσταση μεταξύ της φάλαινας που βρίσκεται στα (X, Y) και του θηράματος που βρίσκονται στα (X *, Y *). Στη συνέχεια δημιουργείται μια σχέση μεταξύ της θέσης της φάλαινας και του θηράματος για να μοντελοποιήσει την ελικοειδή κίνηση των φαλαινών ως εξής:

$$\vec{x}(t+1) = \vec{D} \cdot e^{bl} \cdot cos(2\pi l) + \vec{x}^*(t), \qquad 5.5$$

όπου $\vec{D} = |\vec{x}^*(t) - \vec{x}(t)|$ είναι η απόσταση της τρέχουσας φάλαινας από το θήραμα (βέλτιστη λύση), b είναι μια σταθερά, l είναι ένας τυχαίος αριθμός στο [-1,1] και . είναι ο πολλαπλασιασμός στοιχείο με στοιχείο.

Πρέπει να αναφερθεί ότι οι φάλαινες κολυμπούν γύρω από το θήραμα μέσα σε έναν κύκλο όπου η διάμετρος του μειώνεται συνεχώς και κατά μήκος μιας σπειροειδούς διαδρομής ταυτόχρονα. Για να μοντελοποιηθεί αυτός ο παράλληλος ελιγμός, υποθέτουμε ότι υπάρχει πιθανότητα 50% να επιλέξει είτε την προσέγγιση της περικύκλωσης της μειούμενης διαμέτρου είτε την προσέγγιση του σπειροειδούς μοντέλου για την ενημέρωση των νέων διανυσμάτων θέσεων των φαλαινών κατά τη διάρκεια του αλγόριθμου.

$$\vec{x}(t+1) = \begin{cases} \vec{x}^*(t) - \vec{A} \cdot \vec{D}, αν\ p < 0.5 \\ \vec{D} \cdot e^{bl} \cdot cos(2\pi l) + \vec{x}^*(t), αν\ p \geq 0.5 \end{cases} \qquad 5.6$$

όπου p είναι τυχαίος αριθμός στο διάστημα [0,1]. Η ίδια προσέγγιση που βασίζεται στην παραλλαγή του διανύσματος $\vec{A}$ μπορεί να χρησιμοποιηθεί για την αναζήτηση της λείας. Στην πραγματικότητα, οι φάλαινες ψάχνουν τυχαία ανάλογα με τη θέση των υπολοίπων. Επομένως, χρησιμοποιούμε το $\vec{A}$ με τις τυχαίες τιμές μεγαλύτερες του 1 ή μικρότερες του -1 για να αναγκάσουμε τα άτομα να κινηθούν μακριά από μια φάλαινα αναφοράς. Σε αντίθεση με την προηγούμενη φάση (της επίθεσης στο θήραμα), στην αναζήτηση του θηράματος ενημερώνουμε τη θέση ενός ατόμου σύμφωνα με ένα τυχαία επιλεγμένο άτομο, αντί του καλύτερου που βρέθηκε μέχρι στιγμής. Αυτός ο μηχανισμός καθώς και η απόλυτη τιμή του A η οποία είναι μεγαλύτερη του ενός, επιτρέπουν στον αλγόριθμο να πραγματοποιήσει μια όχι περιορισμένη αναζήτηση. Το μαθηματικό μοντέλο έχει ως εξής:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{x_{rand}} - \vec{x}\right| \tag{5.7}$$

$$\vec{x}(t+1) = \overrightarrow{x_{rand}} - \vec{A} \cdot \vec{D} \tag{5.8}$$

όπου $\overrightarrow{x_{rand}}$ είναι μία τυχαία επιλεγμένη φάλαινα.

Παρακάτω παρουσιάζονται τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα:

Αλγόριθμος της φάλαινας

```
Αρχικοποίηση του πληθυσμού
Υπολογισμός της αντικειμενικής συνάρτησης για κάθε άτομο του πληθυσμού
Όρισε τη βέλτιστη λύση
for i = 1,
        for agent = 1,
                Ανανέωση των a, A, C, l και p
                if1 (p < 0.5)
                        if2  (|A| < 1)
                                Ανανέωσε το διάνυσμα θέσης κάθε ατόμου μέσω της
                                σχέσης 5.1
                        elseif2 (|A| ≥ 1)
                                Επίλεξε ένα τυχαίο άτομο
                                Ανανέωσε το διάνυσμα θέσης του τρέχοντος ατόμου
                                μέσω της σχέσης 5.8
                        endif2
                elseif1  (p ≥ 0.5)
                        Ανανέωσε το διάνυσμα θέσης μέσω της σχέσης 5.5
                endif1
        end
        Υπολογισμός των νέων τιμών της αντικειμενικής συνάρτησης
        Ανανέωση τη βέλτιστης λύσης
end
```

2.3.6 Ο αλγόριθμος της λιβελούλας (Dragonfly Optimization Algorithm, DA)

Οι λιβελούλες είναι φανταχτερά έντομα. Υπάρχουν σχεδόν 3000 διαφορετικά είδη αυτού του εντόμου σε όλο τον κόσμο. Ο κύκλος ζωής των λιβελούλων περιλαμβάνει δύο σημεία ορόσημα: από το στάδιο της νύμφης στην ενηλικίωση. Ξοδεύουν το κύριο μέρος της ζωής τους στο στάδιο της νύμφης κι έπειτα μεταμορφώνονται σε ενήλικες. Οι λιβελούλες θεωρούνται ως μικρά αρπακτικά που κυνηγούν σχεδόν όλα τα άλλα μικρά έντομα στη φύση. Οι νύμφες κυνηγούν επίσης θαλάσσια έντομα και μικρά ψάρια. Η συμπεριφορά τους ως σμήνος παρουσιάζει ενδιαφέρον διότι έχει δύο μόνο σκοπούς, τη θήρα (στατικό σμήνος) και τη μετανάστευση (δυναμικό σμήνος). Στο στατικό σμήνος, οι λιβελούλες δημιουργούν μικρές ομάδες και πετούν μπρος και πίσω σε μια μικρή περιοχή για να θηρεύσουν άλλα ιπτάμενα έντομα όπως πεταλούδες και κουνούπια. Οι τοπικές μετακινήσεις και οι απότομες αλλαγές στην πτήση τους αποτελούν τα κύρια χαρακτηριστικά ενός στατικού σμήνους. Σε δυναμικά σμήνη, ωστόσο, ένας τεράστιος αριθμός λιβελούλων προκαλεί στο σμήνος τη μετανάστευση χρησιμοποιώντας μια κατεύθυνση διανύοντας μεγάλες αποστάσεις. Η έμπνευση του αλγόριθμου προέρχεται από τις στατικές και δυναμικές συμπεριφορές ενός σμήνους. Αυτές οι δυο συμπεριφορές είναι παρόμοιες με τις δύο κύριες φάσεις βελτιστοποίησης των μεθευρετικών αλγορίθμων. Οι λιβελούλες δημιουργούν μικρότερα σμήνη (υποομάδες) και πετούν σε διαφορετικές περιοχές ως ένα στατικό σμήνος, που είναι το κύριο μέρος της εξερεύνησης. Ωστόσο, οι λιβελούλες πετούν σε μεγαλύτερα σμήνη κατά μήκος μιας κατεύθυνσης, η οποία είναι ευνοϊκή κατά τη δεύτερη φάση. Στόχος του αλγορίθμου είναι η μαθηματική μοντελοποίηση των δύο φάσεων. Σύμφωνα με το Reynolds, οι τρεις πρωταρχικές αρχές στη συμπεριφορά ενός σμήνους είναι:

- Ο διαχωρισμός, που αναφέρεται στην αποφυγή της στατικής σύγκρουσης των ατόμων κοινής γειτονιάς μεταξύ τους.
- Ευθυγράμμιση, που δείχνει συγχρονισμό της ταχύτητας των ατόμων στη γειτονιά.
- Συνοχή, η οποία αναφέρεται στην τάση των ατόμων προς το κέντρο της γειτονιάς.

Ο κύριος στόχος κάθε σμήνους είναι η επιβίωση. Γι' αυτό όλα τα άτομα πρέπει να προσελκύονται από πηγές τροφίμων και να αποσπώνται από τους εχθρούς. Λαμβάνοντας υπόψη αυτά τα δύο, υπάρχουν πέντε παράγοντες που συμβάλουν στην ενημέρωση της θέσης των ατόμων. Αυτοί οι παράγοντες υπολογίζονται μέσω των σχέσεων παρακάτω:

- Διαχωρισμός:

$$S_i = -\sum_{j=1}^{N} x - x_j \qquad 6.1$$

όπου x είναι η θέση του τρέχοντος ατόμου, x_j είναι η θέση του j γειτονικού ατόμου και N είναι ο αριθμός των γειτονικών ατόμων.

- Ευθυγράμμιση:

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \qquad 6.2$$

όπου x_j είναι η ταχύτητα του j γειτονικού ατόμου.

- Συνοχή:

$$C_i = \frac{\sum_{j=1}^{N} x_j}{N} - x \qquad 6.3$$

όπου x είναι η θέση του τρέχοντος ατόμου, N είναι ο αριθμός των γειτονιών (υποομάδων) και x_j είναι η θέση του j γειτονικού ατόμου.

- Έλξη προς την τροφή:

$$F_i = X^+ - X \qquad 6.4$$

όπου το X είναι η θέση του τρέχοντος ατόμου και X^+ είναι η θέση της πηγής τροφής.

- Αντιπερισπασμός προς τους εχθρούς:

$$E_i = X^- + X \qquad 6.5$$

όπου το X είναι η θέση του τρέχοντος ατόμου και X^- είναι η θέση του εχθρού.Για να υπολογίσουμε το διάνυσμα θέσης της λιβελούλας στο χώρο αναζήτησης και να προσομοιώσουμε τις κινήσεις τους χρειαζόμαστε δύο διανύσματα. Το βήμα, ΔΧ, και το διάνυσμα θέσης, Χ. Στον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων το ΔΧ είναι ανάλογο του διανύσματος της ταχύτητας. Έτσι κι ο αλγόριθμος της λιβελούλας βασίζεται σε αυτήν την αναλογία. Το ΔΧ δείχνει την κατεύθυνση της κίνησης τον λιβελούλων (ταχύτητα) και ορίζεται ως ακολούθως:

$$\Delta X_{t+1} = (sS_i + \alpha A_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad 6.6$$

όπου s, α, c, f, e και w είναι τα βάρη διαχωρισμού, ευθυγράμμισης, συνοχής, έλξης της τροφής, αντιπερισπασμού προς τον εχθρό και του βήματος αντίστοιχα. Το S_i είναι ο διαχωρισμός, A_i είναι η ευθυγράμμιση, C_i είναι η συνοχή, F_i είναι η πηγή τροφής και E_i είναι η θέση του εχθρού του i ατόμου. Τέλος, t είναι η τρέχουσα επανάληψη. Αφού υπολογιστεί το διάνυσμα ΔX τα διανύσματα θέσης υπολογίζονται ως ακολούθως:

$$X_{t+1} = X_t + \Delta X_{t+1} \quad 6.7$$

όπου t είναι η τρέχουσα επανάληψη.

Παρακάτω παρουσιάζονται τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα:

Αλγόριθμος της λιβελούλας

Αρχικοποίηση αρχικού πληθυσμού
Αρχικοποίηση διανύσματος ΔX
for i=1,
    Υπολογισμός της αντικειμενικής συνάρτησης όλων των ατόμων του πληθυσμού
    Ανανέωση του διανύσματος θέσης της τροφής και του εχθρού
    Ανανέωση s, α, c, f, e και w
    Υπολογισμός των S, A, C, F *και* E μέσω των εξισώσεων 6.1 ως 6.5
    Ανανέωση του διανύσματος ταχύτητας μέσω της σχέσης 6.6
    Ανανέωση του διανύσματος θέσης μέσω της σχέσης 6.7
end

2.3.7 Ο αλγόριθμος των μυρμηλεοντίδων (Antlion Optimizer, ALO)

Οι μυρμηλέοντες ανήκουν στην τάξη των νευρόπτερων. Ο κύκλος ζωής τους περιλαμβάνει δύο κύριες φάσεις, της κάμπιας και του ενήλικα, διάρκειας δύο-τριών ετών. Ωστόσο, χρονικά υπερτερεί η φάση της κάμπιας και η ενηλικίωση διαρκεί μόνο τρεις με πέντε βδομάδες. Οι ενήλικοι μυρμηλέοντες που είναι λιγότερο γνωστοί έχουν την ικανότητα να πετούν και αναγνωρίζονται λανθασμένα ως λιβελούλες. Η ονομασία τους, μυρμηλέοντες, περιγράφει ακριβώς την θηρευτική συμπεριφορά τους προς τα μυρμήγκια. Οι κάμπιες μυρμηλέοντες σκάβουν κωνικές κοιλότητες διαφορετικού βάθους και περιμένουν στο κάτω μέρος να γλιστρήσει η λεία τους και να πέσει μέσα. Όταν ένα έντομο πέσει μέσα ο μυρμηλέων θα προσπαθήσει να το πιάσει καθώς αυτό θα επιχειρεί να δραπετεύσει. Ο μυρμηλέων προσπαθεί με έξυπνο τρόπο να παρασύρει τη λεία του ρίχνοντας άμμο προς την άκρη του λάκκου. Αφού συλληφθεί το θήραμα ο μυρμηλέων το τραβά και το τρώει, κι όταν ολοκληρώσει το γεύμα του αποσύρει τα απομεινάρια προετοιμάζοντας το λάκκο για το επόμενο θήραμα. Αξίζει να σημειωθεί ότι το μέγεθος της παγίδας εξαρτάται από την ένταση της πείνας και από το σχήμα του φεγγαριού. Όταν έχει πανσέληνο κι όταν είναι πολύ πεινασμένοι οι λάκκοι είναι μεγαλύτεροι. Το μέγεθος της παγίδας είναι ανάλογο με το ποσοστό επιτυχίας της θήρευσης. Ο αλγόριθμος αυτός είναι εμπνευσμένος από τη θηρευτική συμπεριφορά τους.

Στον αλγόριθμο αυτό τα μυρμήγκια αποτελούν πράκτορες αναζήτησης και μετακινούνται στο χώρο αναζήτησης ενώ στους μυρμηλέοντες επιτρέπεται να τα θηρεύσουν και να γίνουν καλύτερα άτομα (λύσεις). Σε κάθε επανάληψη το διάνυσμα θέσης κάθε μυρμηγκιού ανανεώνεται σε σχέση με έναν επιλεγμένο μυρμηλέων, ο οποίος έχει επιλεχθεί μέσω του roulette wheel selection. Με τη χρήση αυτού, καλύτερες λύσεις έχουν περισσότερες πιθανότητες να επιλεχθούν καθώς οι μυρμηλεοντίδες με μεγαλύτερες παγίδες έχουν περισσότερες πιθανότητες να κυνηγάν περισσότερη λεία. Η αρχικοποίηση των διανυσμάτων θέσεων των μυρμηγκιών και των μυρμηλεοντίδων γίνεται με μία τυχαία γεννήτρια αριθμών και στη συνέχεια ακολουθεί ο υπολογισμός της αντικειμενικής συνάρτησης κάθε ατόμου. Έπειτα επιλέγεται η βέλτιστη λύση μυρμηλέοντος. Σε κάθε επανάληψη για κάθε ένα μυρμήγκι επιλέγεται ένας μυρμηλέων με χρήση του roulette wheel selection, του οποίου το διάνυσμα θέσης ανανεώνεται με χρήση δύο τυχαίων διαδρομών επιλεγμένων με όμοιο τρόπο δηλαδή με χρήση του roulette wheel selection. Έπειτα γίνεται υπολογισμός της αντικειμενικής συνάρτησης κάθε νέας λύσης μυρμηγκιού και γίνεται σύγκριση με αυτής του μυρμηλέοντος. Αν ένα μυρμήγκι είναι καλύτερο από το μυρμηλέων που του αντιστοιχεί τότε το διάνυσμα θέσης του αποτελεί στην επόμενη επανάληψη το διάνυσμα θέσης του μυρμηλέοντος του. Επίσης η βέλτιστη λύση αντικαθίσταται μόνο αν βρεθεί νέα καλύτερη λύση στην επόμενη επανάληψη. Στον αλγόριθμο αυτό, υπάρχουν δύο είδη πληθυσμών. Τα μυρμήγκια και οι θηρευτές τους, οι μυρμηλεοντίδες. Τα διανύσματα θέσης των πληθυσμών αυτών αποτελούν τις N μεταβλητές απόφασης και εκφράζονται ως ακολούθως :

$$Ant's\ location = [A_1, A_2, \dots, A_n, \dots, A_N] \quad 7.1$$

$$Antlion's\ location = [AL_1, AL_2, \dots, AL_n, \dots, AL_N] \quad 7.2$$

όπου A_n και AL_n η μεταβλητή απόφασης και N ο αριθμός των μεταβλητών απόφασης. Όπως προαναφέρθηκε μέσω του roulette wheel για κάθε μυρμήγκι επιλέγεται ένας μυρμηλέων. Πρέπει να σημειωθεί ότι κάποιο μυρμήγκι μπορεί να πέσει σε μία μόνο παγίδα ανά επανάληψη. Ο επιλεγμένος μυρμηλέων για κάθε μυρμήγκι είναι και αυτός ο οποίος το παγιδεύει. Χρησιμοποιώντας το μηχανισμό αυτό, καλύτερες λύσεις έχουν περισσότερες πιθανότητες να επιλεχθούν αφού μυρμηλεοντίδες με μεγαλύτερες παγίδες θηρεύουν περισσότερα μυρμήγκια.

Όταν ένα μυρμήγκι πέφτει στην παγίδα, ο μυρμηλέων αρχίζει να ρίχνει άμμο στο κέντρο του το λάκκου για να γλιστρήσει το μυρμήγκι που προσπαθεί να ξεφύγει. Αυτή η συμπεριφορά περιγράφεται μαθηματικά με τη μείωση της ακτίνας του τυχαίου περιπάτου του μυρμηγκιού. Έτσι το εύρος των ορίων των μεταβλητών απόφασης μειώνεται και ανανεώνεται, όπως εκφράζεται ακολούθως:

$$\gamma(It) = \frac{c(It)}{R} \quad 7.3$$

$$\delta(It) = \frac{d(It)}{R} \quad 7.4$$

όπου $\gamma(It)$ τροποποιημένο διάνυσμα που περιλαμβάνει το ελάχιστο όλων των μεταβλητών απόφασης της i επανάληψης, $c(It)$ διάνυσμα που περιλαμβάνει το ελάχιστο όλων των μεταβλητών απόφασης της i επανάληψης, R η αναλογία που δίνεται στη σχέση 7.5, $\delta(It)$ τροποποιημένο διάνυσμα που περιλαμβάνει το μέγιστο όλων των μεταβλητών απόφασης της i επανάληψης και $d(It)$ διάνυσμα που περιλαμβάνει το μέγιστο όλων των μεταβλητών απόφασης της i επανάληψης.

όπου

$$R = 10^w \frac{It}{IT} \tag{7.5}$$

όπου w μία σταθερά που βασίζεται στον αριθμό της τρέχουσας επανάληψης και ορίζεται ως εξής:

$$w = \begin{cases} 2\ if\ It > 0.1IT \\ 3\ if\ It > 0.5IT \\ 4\ if\ It > 0.75IT \\ 5\ if\ It > 0.9IT \\ 6\ if\ It > 0.95IT \end{cases} \tag{7.6}$$

Όταν ο αριθμός της επανάληψης στην σχέση 7.6 αυξάνεται, η ακτίνα του τυχαίου περιπάτου μειώνεται, γεγονός που εγγυάται τη σύγκλιση του αλγορίθμου.

Οι παγίδες επηρεάζουν τον τυχαίο περίπατο των μυρμηγκιών. Προκειμένου να περιγραφεί μαθηματικά αυτό, τα όρια του περιπάτου των μυρμηγκιών ρυθμίζονται σε κάθε επανάληψη έτσι ώστε το μυρμήγκι να κινείται σε μια υπερακτίνα γύρω από την επιλεγμένη παγίδα. Τα κάτω και τα πάνω όρια του περιπάτου των σε κάθε επανάληψη υπολογίζονται από τις ακόλουθες εξισώσεις:

$$\gamma_m(It) = Antlion_l(It) + \gamma(It) \tag{7.7}$$

$$\delta_m(It) = Antlion_l(It) + \delta(It) \tag{7.8}$$

όπου $\gamma_m(It)$ διάνυσμα που περιλαμβάνει το ελάχιστο όλων των μεταβλητών απόφασης για το m μυρμήγκι στην i επανάληψη, $Antlion_l(It)$ το διάνυσμα θέσης του επιλεγμένου μυρμηλέοντος και $\delta_m(It)$ διάνυσμα που περιλαμβάνει το ελάχιστο όλων των μεταβλητών απόφασης για το m μυρμήγκι. Οι παραπάνω εξισώσεις δείχνουν ότι ο τυχαία διαδρομή των μυρμηγκιών ανήκουν στην υπερακτίνα.

Τα μυρμήγκια κινούνται τυχαία αναζητώντας τροφή. Για να μοντελοποιηθεί αυτή η κίνηση χρησιμοποιείται η τυχαία διαδρομή και εκφράζεται ως εξής:

$$X = [0, cumsum(2r(1)-1), cumsum(2r(2)-1), \dots, cumsum(2r(It)-1, \dots, cumsum(2r(IT)-1)], \quad 7.9$$

όπου X το διάνυσμα της τυχαίας διαδρομής, $cumsum$ το συσσωρευτικό άθροισμα και $r(It)$ στοχαστική συνάρτηση που υπολογίζεται ως ακολούθως:

$$r(It) = \begin{cases} 1 \; if \; rand > 0.5 \\ 0 \; otherwise \end{cases} \quad 7.10$$

όπου $rand$ τυχαία τιμή που δίνει μια γεννήτρια τυχαίων αριθμών ομοιόμορφης κατανομής στο διάστημα [0,1].

Προκειμένου η τυχαία διαδρομή να είναι εντός των ορίων των μεταβλητών απόφασης κανονικοποιούνται ως εξής:

$$Z_n(It) = \frac{(X_n(It) - \alpha_n) + (\delta(It) - \gamma_n(It))}{(\beta_n - \alpha_n)} \quad 7.11$$

όπου $Z_n(It)$ κανονικοποιημένο διάνυσμα θέσης της διαδρομής της n μεταβλητής απόφασης της i επανάληψης, $X_n(It)$ διάνυσμα θέσης της τυχαίας διαδρομής της n μεταβλητής απόφασης της i επανάληψης πριν την κανονικοποίηση, α_n η ελάχιστη τιμή της τυχαίας διαδρομής της n μεταβλητής απόφασης, β_n μέγιστη τιμή της τυχαίας διαδρομής της n μεταβλητής απόφασης, $\gamma_n(It)$ ελάχιστη τιμή της n μεταβλητής απόφασης της i επανάληψης και $\delta(It)$ μέγιστη τιμή της n μεταβλητής απόφασης της i επανάληψης.

Η ελιτιστική επιλογή εγγυάται ότι η ποιότητα που λαμβάνεται δεν θα μειωθεί από τη μια γενιά στην επόμενη. Στον αλγόριθμο αυτό, η καλύτερη λύση από τον πληθυσμό των μυρμηλεοντίδων αποθηκεύεται ως η βέλτιστη. Τα νέα διανύσματα θέσης των μυρμηγκιών ορίζονται ως ο μέσος όρος των τυχαίων περιπάτων του μυρμηλέοντος που έχει επιλεχθεί μέσω του roulette wheel selection και του βέλτιστου, και εκφράζονται μέσω της σχέσης παρακάτω:

$$Ant_m^{It} = \frac{R_l^{It} + R_e^{It}}{2} \quad 7.12$$

όπου Ant_m^{It} είναι το διάνυσμα θέσης του m μυρμηγκιού της It επανάληψης, R_l^{It} είναι ο τυχαίος περίπατος τριγύρω από τον l μυρμηλέων που έχει επιλεγεί με χρήση του roulette wheel selection στην It επανάληψη και R_e^{It} είναι ο τυχαίος περίπατος τριγύρω του βέλτιστου μυρμηλέοντος της It επανάληψης.

Στο τελικό στάδιο της θήρευσης των μυρμηλεοντίδων καθώς ένα μυρμήγκι πέφτει στην παγίδα και ο μυρμηλέων το πιάνει με το σαγόνι του και το ωθεί μέσα στο χώμα ώστε να το καταναλώσει. Στον αλγόριθμο ένας μυρμηλέων πιάνει ένα θήραμα όταν η τιμή της αντικειμενικής συνάρτησης ενός μυρμηγκιού είναι καλύτερη από την αντίστοιχη του μυρμηλέοντος. Σε αυτή την περίπτωση το διάνυσμα θέσης του μυρμηλέοντος αντικαθίσταται από αυτό του θηράματος μυρμηγκιού. Μαθηματικά εκφράζεται η παραπάνω διαδικασία ως εξής:

$$Antlion_l^{It} = Ant_m^{It} \; if \; f(Ant_m^{It}) < f(Antlion_l^{It}) \quad 7.13$$

Παρακάτω παρουσιάζονται τα βήματα του αλγορίθμου σε μορφή ψευδοκώδικα:

Αλγόριθμος των μυρμηλεοντίδων

Αρχικοποίηση αρχικού πληθυσμού

for iter=1**,**

for i=1,

Υπολογισμός της αντικειμενικής συνάρτησης όλων των ατόμων του πληθυσμού (μυρμηγκιών και μυρμηλεοντίδων)

Εύρεση καλύτερου ατόμου από τον πληθυσμό των μυρμηλεοντίδων

Επίλεξε έναν μυρμηλέων χρησιμοποιώντας roulette wheel selection

Ανανέωσε τα άνω και κάτω όρια χρησιμοποιώντας τις εξ. 11.7 και 11.8

Ανανέωσε τα γ και δ χρησιμοποιώντας τις εξ. 11.11 και 11.12

Δημιούργησε δυο τυχαίες διαδρομές μεταξύ του επιλεγμένου και του καλύτερου μυρμηλέοντος μέσω της εξ. 11.15

Ανανέωσε τα διανύσματα θέσεις των μυρμηγκιών μέσω της εξ. 11.16

end

Υπολογισμός της αντικειμενικής συνάρτησης για ολόκληρο τον πληθυσμό

Αντικατάσταση του μυρμηλέοντος με του αντίστοιχου μυρμηγκιού σε περίπτωση εύρεσης καλύτερης λύσης

Εύρεση καλύτερου ατόμου από τον πληθυσμό των μυρμηλεοντίδων

end

Κεφάλαιο 5ο

Αποτελέσματα

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των αποτελεσμάτων των αλγορίθμων που υλοποιήθηκαν GSA, BB-BC, GWO, SMO, WO, DA και ALO καθώς και σύγκριση αυτών με τις τιμές των βέλτιστων. Η σύγκριση θα γίνει σε σχέση με την ελάχιστη τιμή που βρέθηκε σε κάθε αλγόριθμο κι έπειτα θα σχολιαστεί η αποτελεσματικότητα τους σύμφωνα με τα αποτελέσματα που εξήχθησαν με κριτήριο το ποσοστό απόκλισης της κάθε λύσης από το βέλτιστο της βιβλιογραφίας. Όπως αναφέρθηκε και στο 1ο κεφάλαιο η βέλτιστη τιμή των συναρτήσεων δοκιμής όπου υλοποιήθηκαν είναι το μηδέν και σκοπός της παρούσας εργασίας ήταν να το προσεγγίσουμε ή ακόμα καλύτερα να το βρούμε. Δεδομένων λοιπόν των αλγορίθμων που χρησιμοποιήθηκαν και των συναρτήσεων δοκιμής που ήταν προς επίλυση δοκιμάστηκαν αριθμοί μεταβλητών απόφασης από τέσσερα έως και δέκα. Έτσι λοιπόν μετά από σειρά εκτελέσεων σε κάθε συνάρτηση και αλγόριθμο σημειώθηκαν οι βέλτιστες τιμές και παρουσιάζονται ακολούθως.

3.1 Παρουσίαση αποτελεσμάτων

Στη συνέχεια παρουσιάζονται πίνακες που περιέχουν τα αποτελέσματα κάθε αλγόριθμου σε κάθε πρόβλημα που επιλύθηκε. Στην πρώτη στήλη περιέχεται η ελάχιστη τιμή που έχει βρει ο αλγόριθμος, στη δεύτερη στήλη παρουσιάζεται ο μέσος όρος των λύσεων που βρήκαμε, και τέλος στην τρίτη, η απόκλιση από το βέλτιστο.

Antlion Optimizer

n=4

	min	average	ω
rosenbrock	1,53	2,88	1,53
sphere	0,04	0,11	0,04
rastrigin	3,65	4,10	3,65
griewank	2,47	2,67	2,47

n=5

	min	average	ω
rosenbrock	2,66	4,43	2,66
sphere	0,17	0,18	0,17
rastrigin	5,69	7,68	5,69
griewank	1,87	2,79	1,87

n=6

	min	average	ω
rosenbrock	6,29	6,75	6,29
sphere	0,22	0,35	0,22
rastrigin	11,35	12,22	11,35
griewank	3,86	4,26	3,86

n=7

	min	average	ω
rosenbrock	4,77	7,56	4,77
sphere	0,26	0,44	0,26
rastrigin	19,21	20,44	19,21
griewank	4,12	5,30	4,12

n=8

	min	average	ω
rosenbrock	26,15	26,66	26,15
sphere	0,55	0,59	0,55
rastrigin	20,42	21,81	20,42
griewank	4,65	5,83	4,65

n=9

	min	average	ω
rosenbrock	22,02	27,24	22,02
sphere	0,37	0,72	0,37
rastrigin	25,01	25,93	25,01
griewank	7,68	9,26	7,68

n=10

	min	average	ω
rosenbrock	26,74	29,96	26,74
sphere	0,91	1,15	0,91
rastrigin	24,24	28,50	24,24
griewank	6,43	7,53	6,43

Στον αλγόριθμο αυτό παρατηρούνται ικανοποιητικά αποτελέσματα καθώς όμως ο αριθμός n αυξάνεται παρουσιάζεται χειροτέρευση των αποτελεσμάτων ειδικά στις συναρτήσεις rosenbrock και rastrigin. Η συνάρτηση sphere φαίνεται να μην επηρεάζεται από τον αριθμό αυτό αφού ο αλγόριθμος έχει αποδώσει πολύ καλά αποτελέσματα αρκετά κοντά στο βέλτιστο.

Dragonfly Algorithm

n=4

	min	average	ω
rosenbrock	0,15	0,28	0,15
sphere	0,00	0,00	0,00
rastrigin	2,01	2,20	2,01
griewank	1,00	1,00	1,00

n=5

	min	average	ω
rosenbrock	0,06	0,31	0,06
sphere	0,00	0,00	0,00
rastrigin	2,51	3,00	2,51
griewank	1,00	1,00	1,00

n=6

	min	average	ω
rosenbrock	0,10	0,13	0,10
sphere	0,00	0,00	0,00
rastrigin	2,32	3,08	2,32
griewank	1,00	1,00	1,00

n=7

	min	average	ω
rosenbrock	3,20	3,89	3,20
sphere	0,00	0,01	0,00
rastrigin	4,06	4,07	4,06
griewank	1,01	1,08	1,01

n=8

	min	average	ω
rosenbrock	4,01	4,03	4,01
sphere	0,00	0,00	0,00
rastrigin	7,69	8,37	7,69
griewank	1,07	1,49	1,07

n=9

	min	average	ω
rosenbrock	3,21	3,21	3,21
sphere	0,03	0,03	0,03
rastrigin	18,74	20,81	18,74
griewank	1,15	1,26	1,15

n=10

	min	average	ω
rosenbrock	8,26	8,60	8,26
sphere	0,09	0,09	0,09
rastrigin	13,07	13,27	13,07
griewank	2,07	2,40	2,07

Στον αλγόριθμο της λιβελούλας παρατηρείται το ίδιο φαινόμενο που είδαμε στον προηγούμενο αλγόριθμο, όμως σε αρκετά μικρότερο βαθμό. Εδώ τα αποτελέσματα της συνάρτησης rosenbrock είναι αρκετά βελτιωμένα συγκρίσει με τα προηγούμενα. Τα αποτελέσματα της συνάρτησης rastrigin έχουν επηρεαστεί περισσότερο στον τρέχων αλγόριθμο. Ομοίως με τα προηγούμενα, τα αποτελέσματα της sphere είναι πολύ καλύτερα, αυτή τη φορά όμως ο αλγόριθμος σχεδόν τα κατάφερε και βρήκε το βέλτιστο.

Gravitational Search Algorithm

n=4

	min	average	ω
rosenbrock	1,20	1,85	1,20
sphere	0,00	0,13	0,00
rastrigin	2,13	2,73	2,13
griewank	2,25	2,75	2,25

n=5

	min	average	ω
rosenbrock	1,97	2,97	1,97
sphere	0,03	0,31	0,03
rastrigin	2,79	4,53	2,79
griewank	3,28	3,79	3,28

n=6

	min	average	ω
rosenbrock	2,40	5,55	2,40
sphere	0,18	0,43	0,18
rastrigin	4,02	5,35	4,02
griewank	6,13	7,78	6,13

n=7

	min	average	ω
rosenbrock	8,71	11,18	8,71
sphere	0,17	0,66	0,17
rastrigin	9,16	9,48	9,16
griewank	11,70	13,51	11,70

n=8

	min	average	ω
rosenbrock	16,65	19,70	16,65
sphere	0,79	0,93	0,79
rastrigin	7,82	10,38	7,82
griewank	11,76	14,45	11,76

n=9

	min	average	ω
rosenbrock	26,67	29,31	26,67
sphere	1,10	1,30	1,10
rastrigin	10,28	14,10	10,28
griewank	16,31	17,15	16,31

n=10

	min	average	ω
rosenbrock	31,67	34,96	31,67
sphere	1,02	1,46	1,02
rastrigin	25,94	27,40	25,94
griewank	18,25	19,04	18,25

Ο αλγόριθμος αναζήτησης της βαρυτικής έλξης δεν έχει καταφέρει να εξάγει καλά αποτελέσματα κι αυτό φαίνεται από τον πίνακα παραπάνω. Στις συναρτήσεις rosenbrock, rastrigin και griewank βλέπουμε ότι έχει δυσκολευτεί αρκετά. Εδώ πάλι τα αποτελέσματα της συνάρτησης sphere είναι πολύ καλύτερα των υπολοίπων.

Grey Wolf Optimization Algorithm

n=4

	min	average	ω
rosenbrock	1,05	2,09	1,05
sphere	0,41	0,49	0,41
rastrigin	2,85	3,82	2,85
griewank	10,82	10,83	10,82

n=5

	min	average	ω
rosenbrock	3,79	7,91	3,79
sphere	0,32	0,34	0,32
rastrigin	8,04	8,66	8,04
griewank	13,29	13,30	13,29

n=6

	min	average	ω
rosenbrock	11,88	18,82	11,88
sphere	0,54	0,74	0,54
rastrigin	11,57	12,65	11,57
griewank	15,76	15,77	15,76

n=7

	min	average	ω
rosenbrock	19,02	23,25	19,02
sphere	0,56	0,78	0,56
rastrigin	21,24	24,25	21,24
griewank	18,22	18,22	18,22

n=8

	min	average	ω
rosenbrock	28,68	32,11	28,68
sphere	0,54	1,04	0,54
rastrigin	28,80	30,36	28,80
griewank	20,69	20,70	20,69

n=9

	min	average	ω
rosenbrock	15,98	30,80	15,98
sphere	1,03	1,34	1,03
rastrigin	32,73	33,50	32,73
griewank	23,14	23,16	23,14

n=10

	min	average	ω
rosenbrock	30,44	33,33	30,44
sphere	1,03	1,24	1,03
rastrigin	39,13	43,20	39,13
griewank	25,63	25,64	25,63

Ο πίνακας παραπάνω εμφανίζει τα αποτελέσματα του αλγόριθμου των γκρίζων λύκων. Για ακόμα μία φορά παρατηρείται ότι όσο ο αριθμός των μεταβλητών απόφασης αυξάνεται, δυστυχώς η ποιότητα των αποτελεσμάτων φθίνει. Ενώ σε μικρό αριθμό n τα αποτελέσματα είναι σχετικά καλά πλην κάποιων εξαιρέσεων βλέπουμε ότι η ποιότητα αυτή χαλάει. Τα αποτελέσματα της sphere δεν φαίνεται να έχουν επηρεαστεί καθόλου. Παρατηρούμε ότι διατηρούν την ποιότητα τους αναλλοίωτη.

Spider Monkey Optimization

n=4

	min	average	ω
rosenbrock	0,00	0,00	0,00
sphere	0,00	0,00	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=5

	min	average	ω
rosenbrock	0,00	0,49	0,00
sphere	0,00	0,00	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=6

	min	average	ω
rosenbrock	0,00	0,92	0,00
sphere	0,00	0,00	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=7

	min	average	ω
rosenbrock	0,29	2,49	0,29
sphere	0,00	0,00	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=8

	min	average	ω
rosenbrock	1,54	3,02	1,54
sphere	0,00	0,17	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=9

	min	average	ω
rosenbrock	3,66	4,81	3,66
sphere	0,00	0,09	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

n=10

	min	average	ω
rosenbrock	1,21	3,23	1,21
sphere	0,00	0,08	0,00
rastrigin	0,00	0,00	0,00
griewank	1,00	1,00	1,00

Ο αλγόριθμος των μαϊμούδων όπως μπορεί να δει κανείς έχει εξάγει πάρα πολύ καλά αποτελέσματα σε όλα τα προβλήματα αφού σε αρκετές περιπτώσεις έχει βρει και τις βέλτιστες τιμές. Μέχρι στιγμής έχει αποδώσει τα καλύτερα αποτελέσματα από όλους τους αλγόριθμους που είδαμε.

Big Bang – Big Crunch Algorithm

n=4

	min	average	ω
rosenbrock	0,55	0,72	0,55
sphere	0,19	0,23	0,19
rastrigin	7,14	7,50	7,14
griewank	10,70	10,70	10,70

n=5

	min	average	ω
rosenbrock	2,00	3,24	2,00
sphere	0,30	0,37	0,30
rastrigin	6,19	10,45	6,19
griewank	13,14	13,14	13,14

n=6

	min	average	ω
rosenbrock	4,11	7,15	4,11
sphere	0,37	0,43	0,37
rastrigin	21,96	23,68	21,96
griewank	15,55	15,56	15,55

n=7

	min	average	ω
rosenbrock	1,69	5,77	1,69
sphere	0,58	0,64	0,58
rastrigin	16,51	18,64	16,51
griewank	18,00	18,00	18,00

n=8

	min	average	ω
rosenbrock	10,24	16,75	10,24
sphere	0,50	0,68	0,50
rastrigin	40,34	42,16	40,34
griewank	20,42	20,44	20,42

n=9

	min	average	ω
rosenbrock	8,10	15,80	8,10
sphere	0,74	0,81	0,74
rastrigin	24,63	32,82	24,63
griewank	22,86	22,87	22,86

n=10

	min	average	ω
rosenbrock	20,70	21,02	20,70
sphere	1,08	1,14	1,08
rastrigin	48,48	49,06	48,48
griewank	25,29	25,30	25,29

Ο αλγόριθμος της μεγάλης έκρηξης δεν μας έχει δώσει ικανοποιητικά αποτελέσματα. Θα μπορούσε να πει κανείς ότι είναι παρόμοια με αυτά του αλγορίθμου αναζήτησης της βαρυτικής έλξης. Βέβαια, όπως σε όλες τις περιπτώσεις τα αποτελέσματα της sphere είναι για ακόμα μια φορά πολύ καλά.

The Whale Optimization Algorithm

n=4

	min	average	ω
rosenbrock	0,05	0,13	0,05
sphere	0,00	0,00	0,00
rastrigin	0,12	0,35	0,12
griewank	1,08	1,10	1,08

n=5

	min	average	ω
rosenbrock	0,34	0,63	0,34
sphere	0,00	0,00	0,00
rastrigin	0,51	1,17	0,51
griewank	1,15	1,38	1,15

n=6

	min	average	ω
rosenbrock	0,36	1,16	0,36
sphere	0,00	0,00	0,00
rastrigin	0,05	1,88	0,05
griewank	2,38	2,64	2,38

n=7

	min	average	ω
rosenbrock	1,40	2,92	1,40
sphere	0,01	0,01	0,01
rastrigin	5,67	7,43	5,67
griewank	2,42	2,74	2,42

n=8

	min	average	ω
rosenbrock	7,82	8,36	7,82
sphere	0,01	0,02	0,01
rastrigin	14,67	16,38	14,67
griewank	2,49	3,18	2,49

n=9

	min	average	ω
rosenbrock	2,66	12,03	2,66
sphere	0,04	0,05	0,04
rastrigin	2,74	7,24	2,74
griewank	4,21	4,41	4,21

n=10

	min	average	ω
rosenbrock	13,52	15,64	13,52
sphere	0,08	0,09	0,08
rastrigin	9,97	10,93	9,97
griewank	2,66	3,45	2,66

Στον πίνακα αυτό παρατηρούμε αρκετά καλά αποτελέσματα ειδικά μέχρι ένα συγκεκριμένο αριθμό n. Στις μεγαλύτερες τιμές του παρατηρείται να φθίνουν τα αποτελέσματα όμως σε συγκεκριμένες συναρτήσεις όπως η rosenbrock και η rastrigin. Είναι πλέον περιττό να αναφερθεί πως τα εξαγόμενα της sphere είναι πάρα πολύ κοντά στα βέλτιστα, ο αλγόριθμος το έχει βρει σχεδόν σε όλες τις περιπτώσεις.

3.2 Συμπεράσματα και σχόλια

Η παρούσα διπλωματική εργασία ασχολείται με τη σύγκριση εξελικτικών αλγορίθμων σε προβλήματα ολικής βελτιστοποίησης, δηλαδή πιο συγκεκριμένα, σε συναρτήσεις δοκιμής όπως οι rosenbrock, sphere, rastrigin και griewank. Σκοπός μας είναι η ελαχιστοποίηση των προβλημάτων συνεπώς επιδιώκουμε το κατά δύναμιν μικρότερες τιμές. Εκτελέστηκε μία σειρά εκτελέσεων σε όλα τα προβλήματα από κάθε μεθοδολογία και παρατηρήθηκε η απόδοση κάθε αλγόριθμου. Έτσι λοιπόν, σχεδόν σε όλες τις μεθοδολογίες είδαμε ότι καθώς ο αριθμός των μεταβλητών απόφασης αυξανόταν τα αποτελέσματα παρουσίαζαν αύξηση, σε κάποιες περιπτώσεις πολύ μικρή και σε άλλες αρκετά μεγάλη. Δηλαδή η ποιότητα έφθινε. Έτσι λοιπόν θα μπορούσε να πει κανείς ότι οι αλγόριθμοι παρουσίαζαν ευαισθησία σε αυτόν τον παράγοντα. Ο αλγόριθμος των μαϊμούδων έδειξε ότι επηρεάστηκε σημαντικά λιγότερο από τον αριθμό των μεταβλητών απόφασης παρουσιάζοντας τα καλύτερα αποτελέσματα όλων. Σχετικά καλή απόδοση, σε αυτόν τον παράγοντα, είχε και ο αλγόριθμος της λιβελούλας εξαιρουμένης μιας συνάρτησης, της griewank. Σε μεγάλο αριθμό n κακή απόδοση είχαν οι αλγόριθμοι BB-BC, GWO, GSA και ο ALO, αν και σε μικρό n είχαν ορατά καλύτερη, σε κάποιες περιπτώσεις κιόλας προσέγγιζαν κατά πολύ το βέλτιστο ή ακόμα το έβρισκαν. Τις καλύτερες απ’ όλους τους αλγορίθμους επιδόσεις, παρουσίασε ο αλγόριθμος των μαϊμούδων με διαφορά. Τέλος, ο αλγόριθμος της φάλαινας κι αυτός της λιβελούλας αν και βρήκε το βέλτιστο σε κάποιες περιπτώσεις, σε μεγάλο αριθμό n δεν αντεπεξήλθε το ίδιο καλά. Άρα θα μπορούσε να πει κανείς ότι είχαν λιγότερο καλή απόδοση όμως όχι τη χείριστη. Σε γενικές γραμμές κάποιες συναρτήσεις δυσκόλεψαν τους αλγόριθμους με αποτέλεσμα την όχι τόσο καλή απόδοση τους, όπως η rosenbrock και η προαναφερθείσα griewank. Παρατηρώντας τα αποτελέσματα αυτό δεν ισχύει για τη συνάρτηση sphere καθώς όλες οι μεθοδολογίες είχαν πολύ καλή απόδοση. Κάποιες εμφάνισαν πολύ καλά αποτελέσματα και κάποιες άλλες τέλεια.

Β Ι Β Λ Ι Ο Γ Ρ Α Φ Ι Α

[1] E.S. Ali, S.M. Abd Elazim, A.Y. Abdelaziz. ‘Ant Lion Optimization Algorithm for optimal location and sizing of renewable distributed generations’. Renewable Energy Volume 101, February 2017, Pages 1311-1324, Elsevier.

[2] Seyedali Mirjalili, Seyed Mohammad, Mirjalili Andrew Lewis. ‘Grey Wolf Optimizer’. Advances in Engineering Software Volume 69, March 2014, Pages 46-61, Elsevier.

[3] Seyedali Mirjalili. ‘Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems’. Neural Comput & Applic (2016) 27:1053-1073, Springer.

[4] Seyedali Mirjalili, Andrew Lewis. ‘The Whale Optimization Algorithm’. Advances in Engineering Software, Volume 95, May 2016, Pages 51-67, Elsevier.

[5] https://en.wikipedia.org/wiki/Test_functions_for_optimization

[6] Cheng He, Hongwei Mo, Linqiang Pan, Yuxin Zhao. Bio-inspired Computing: Theories and Applications: 12th International Conference, BIC-TA 2017 Harbin, China, December 1-3, 2017 Proceedings, Springer.

[7] Osyczka Andrzej, Krenich Stanislaw. ‘Evolutionary Algorithms for Global Optimization’. Global Optimization (2006), pp 267-300, Springer.

[8] Jagdish Chand Bansal, Shimpi Singh Jadon, Harish Sharma. ‘Spider Monkey Optimization algorithm for numerical optimization′, Memetic computing, March 2014, pp 31–47, Springer.