



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΟΙΚΟΝΟΜΙΑΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ
Κουντουριώτου 41, Χίος, 82121

ΝΟΗΜΟΝΕΣ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΕΜΠΝΕΥΣΜΕΝΕΣ ΑΠΟ ΤΟΝ
ΦΥΣΙΚΟ ΚΟΣΜΟ ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΩΝ

Αλέξανδρος Τζανέτος

Χίος, Μάρτιος 2020

ΝΟΗΜΟΝΕΣ ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΕΜΠΝΕΥΣΜΕΝΕΣ ΑΠΟ ΤΟΝ ΦΥΣΙΚΟ ΚΟΣΜΟ ΓΙΑ ΤΗΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΩΝ

Αλγόριθμος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό

Διδακτορική Διατριβή

ΑΛΕΞΑΝΔΡΟΣ Κ. ΤΖΑΝΕΤΟΣ

Συμβουλευτική
επιτροπή:

Γεώργιος Δούνιας

Καθηγητής Τμήματος Μηχανικών Οικονομίας και Διοίκησης,
Πολυτεχνικής Σχολής Πανεπιστημίου Αιγαίου

Αγάπιος Πλατής

Καθηγητής Τμήματος Μηχανικών Οικονομίας και Διοίκησης,
Πολυτεχνικής Σχολής Πανεπιστημίου Αιγαίου

Δημήτριος Κουλουριώτης

Καθηγητής Τμήματος Μηχανικών Παραγωγής και Διοίκησης,
Δημοκρίτειου Πανεπιστημίου Θράκης

Μέλη
εξεταστικής
επιτροπής:

Μιχαήλ Δούμπος

Καθηγητής Τμήματος Μηχανικών Παραγωγής και Διοίκησης,
Πολυτεχνείου Κρήτης

Ιωάννης Μαρινάκης

Αναπληρωτής Καθηγητής Τμήματος Μηχανικών Παραγωγής και
Διοίκησης, Πολυτεχνείου Κρήτης

Βασίλειος Ζεϊμπέκης

Επίκουρος Καθηγητής Τμήματος Μηχανικών Οικονομίας και
Διοίκησης, Πολυτεχνικής Σχολής Πανεπιστημίου Αιγαίου

Βασίλειος Κούτρας

Επίκουρος Καθηγητής Τμήματος Μηχανικών Οικονομίας και
Διοίκησης, Πολυτεχνικής Σχολής Πανεπιστημίου Αιγαίου

Περίληψη

Η εποχή που διανύουμε φέρνει στο προσκήνιο την Τεχνητή Νοημοσύνη, καθότι διάφοροι κλάδοι ενσωματώνουν εντυπωσιακές εφαρμογές της. Πολλές αναφορές γίνονται στην 4^η Βιομηχανική Επανάσταση, η οποία έχει πλέον την Τεχνητή Νοημοσύνη ως κύριο πυλώνα της. Ήδη από τη δεκαετία του 1950 αναπτύσσονταν νοήμονα συστήματα λήψης απόφασης, τα οποία συνήθως είχαν υποστηρικτικό ρόλο για έναν λήπτη απόφασης. Προβλήματα στα οποία έπρεπε να βρεθεί μια βέλτιστη στρατηγική, τα επονομαζόμενα προβλήματα βελτιστοποίησης, εμφανίζονται σε διάφορα πεδία όπως η Χρηματοοικονομική Μηχανική (Financial Engineering), η Μηχανική της Διοίκησης (Management Engineering) και η Επιχειρησιακή Έρευνα (Operational Research). Πολλά εκ των προβλημάτων που καλείται να λύσει ένας Μηχανικός ανήκουν στα προβλήματα βελτιστοποίησης. Και καθώς τα προβλήματα που αντιμετωπίζει η βιομηχανία γίνονται ολοένα και πιο δύσκολα, αυξάνεται η χρησιμότητα των Νοημόνων Υπολογιστικών Μεθόδων και η ενσωμάτωσή τους στους σχετικούς κλάδους της βιομηχανίας.

Το φάσμα των Νοημόνων Υπολογιστικών Μεθόδων είναι ευρύ και εκτείνεται από τις μεθόδους Μηχανικής Μάθησης (Machine Learning) μέχρι και τις Νοήμονες Μεθόδους που είναι Εμπνευσμένες από τη Φύση (Nature-inspired Intelligent Methods). Οι τελευταίες, παρόλο που άρχισαν να αναπτύσσονται από τα μέσα της δεκαετίας του '80, αποτέλεσαν ξεχωριστή κατηγορία Νοημόνων Υπολογιστικών Μεθόδων στη βιβλιογραφία στις αρχές του 2000. Λόγω της ικανότητάς τους να επιλύουν δύσκολα προβλήματα βελτιστοποίησης πολλών διαστάσεων, οι Νοήμονες Υπολογιστικές Μέθοδοι Εμπνευσμένες από τον Φυσικό Κόσμο έχουν προσελκύσει το ενδιαφέρον των ερευνητών διεθνώς ήδη από την προηγούμενη δεκαετία.

Στην παρούσα διατριβή συλλέχθηκαν και μελετήθηκαν όλες οι Νοήμονες Υπολογιστικές Μέθοδοι που είναι Εμπνευσμένες από τον Φυσικό Κόσμο με σκοπό να εντοπιστούν τα χαρακτηριστικά που μπορούν να βελτιώσουν την απόδοσή τους. Επιπρόθετα, η μελέτη εστίασε και στα προβλήματα (χώρους εφαρμογής) που εφαρμόζονται οι παραπάνω μέθοδοι. Από αυτή την εκτενή μελέτη της βιβλιογραφίας προέκυψαν κάποιες παρατηρήσεις, οι οποίες αποτυπώθηκαν σε ένα πλαίσιο εργασίας (framework), το οποίο μπορεί να ακολουθήσει ο ενδιαφερόμενος ερευνητής, ώστε να αναπτύξει ή να βελτιώσει κάποια υπάρχουσα Νοήμονα Υπολογιστική Μέθοδος Εμπνευσμένη από τον Φυσικό Κόσμο.

Επιπλέον, σε συμφωνία με το προτεινόμενο πλαίσιο εργασίας και με βάση τα προαναφερθέντα χαρακτηριστικά, αναπτύχθηκε μια νέα νοήμων υπολογιστική μέθοδος εμπνευσμένη από τους νόμους της υποθαλάσσιας ακουστικής, η οποία αποδεικνύεται αποτελεσματική στην επίλυση διαφόρων πρακτικών εφαρμογών από το χώρο της Μηχανικής των Αποφάσεων (Βελτιστοποίηση Χαρτοφυλακίου, Βέλτιστη Εξισορρόπηση Πόρων στη Διοίκηση Έργων, Οικονομική Κατανομή Φορτίου,

Βελτιστοποίηση Γραμμών Παραγωγής). Μάλιστα, τα αποτελέσματα της προτεινόμενης μεθόδου συγκρίνονται με αυτά άλλων κλασικών μεθόδων της βιβλιογραφίας ή μεθόδων που ανήκουν εξίσου στην κατηγορία των Υπολογιστικών Μεθόδων που είναι Εμπνευσμένες από τον Φυσικό Κόσμο, τις οποίες και ξεπερνάει σε απόδοση.

Κατά τη διάρκεια της έρευνας, εφαρμόζοντας την προτεινόμενη μέθοδο σε προβλήματα των χώρων εφαρμογής που αναφέρονται παραπάνω, βελτιώθηκαν μηχανισμοί της μεθόδου, ενώ αναπτύχθηκαν μηχανισμοί που βοήθησαν στην αντιμετώπιση διάφορων προβλημάτων που αντιμετωπίζουν οι Νοήμονες Υπολογιστικές Μέθοδοι που είναι Εμπνευσμένες από τον Φυσικό Κόσμο. Μάλιστα, οι μηχανισμοί που αναπτύχθηκαν αποτελούν και μέρος της συνεισφοράς της παρούσας διατριβής, καθώς μπορούν να υιοθετηθούν και από άλλες μεθόδους Υπολογιστικής Νοημοσύνης.

Τέλος, σε ορισμένες περιπτώσεις αναπτύχθηκαν και υβριδικά σχήματα που περιέχουν την υπολογιστική μέθοδο που προτείνεται στην παρούσα διατριβή με σκοπό την βελτίωση της απόδοσής της.

Λέξεις Κλειδιά: Νοήμονες Αλγόριθμοι Εμπνευσμένοι από τη Φύση, Νοημοσύνη Εμπνευσμένη από τη Φύση, Βελτιστοποίηση

Nature Inspired Intelligence in Optimization

Sonar Inspired Optimization

Summary

The current era brings Artificial Intelligence to the forefront, as various industries incorporate impressive applications. Many references are made to the 4th Industrial Revolution, which now has Artificial Intelligence (AI) as its main pillar. As early as the 1950s, intelligent decision-making systems were being developed, which usually played a supporting role for a decision-maker. Problems in which an optimal strategy had to be found, the so-called optimization problems, arise in a variety of areas such as Financial Engineering, Management Engineering and Operational Research. Many of the problems that an engineer has to solve are optimization problems. While the problems that the industry faces become more and more difficult, the usefulness of Intelligent Computing and its integration into the relevant industry sectors is increasing.

The range of Computational Intelligence oriented methods is wide, ranging from Machine Learning to Nature-inspired Intelligent (NII) Methods. Although NII methods initially appeared back in the mid-80s, they constituted a separate category of Computational Intelligent (CI) only in the early 2000's. Due to their ability to solve difficult multi-dimensional optimization problems, Computational Intelligence oriented methods have attracted the interest of researchers over the past decade.

In this thesis, all Computational Intelligence oriented methods inspired by nature have been collected and studied in order to identify the features that can improve their performance. Moreover, this study focuses also on the application areas that these methods are used. As a result from this comprehensive literature review, some conclusions have been drawn. These conclusions constitute a framework that can be carried out for the development or improvement of Nature-inspired Intelligent Method.

In addition, based on this framework and the aforementioned features, a new Nature-Inspired Intelligent approach inspired by the laws of underwater acoustics, has been developed. This approach has been proved to be effective in various application domains (Portfolio Optimization, Resource Leveling in Project Management, Economic Load Dispatch, Production Lines Optimization). What is more, compared with other state-of-the-art methods of literature and other Nature-Inspired Algorithms, the proposed approach obtains a superior performance.

Throughout the conducted research, the main mechanisms of the proposed algorithm have been improved, based on the experimentation in the aforementioned application areas. Furthermore, some novel mechanisms (in respect of autotuning and balancing exploration and exploitation) have been implemented in the proposed

approach. These mechanisms are part of the contribution of this thesis, since they can be implemented also in several Computational Intelligence oriented methods, in order to overcome problems that these methods usually face off.

Furthermore, in some cases, hybrid intelligent schemes have been developed containing the proposed approach, in order to obtain improved performance.

Keywords: Nature Inspired Intelligent Algorithms, Nature Inspired Intelligence, Optimization

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τα μέλη της συμβουλευτικής επιτροπής, τους κ.κ. Καθηγητές Γεώργιο Δούνια, Ιωάννη Γκιάλα και Δημήτριο Κουλουριώτη για τις συμβουλές τους και την γενικότερη υποστήριξή τους καθ' όλη τη διάρκεια της διατριβής, καθώς και τον Καθηγητή κο. Αγάπιο Πλατή που αντικατέστησε τον κ. Γκιάλα κατά τους τελευταίους μήνες της έρευνάς μου όπου ο δεύτερος πήρε ακαδημαϊκή άδεια. Ιδιαίτερη μνεία θα ήθελα να κάνω στον κ. Γ. Δούνια, ο οποίος με εμπιστεύθηκε και όντας επιβλέπων καθηγητής της διπλωματικής μου, με παρότρυνε για το επόμενο βήμα, αυτό της εκπόνησης της παρούσας διδακτορικής διατριβής.

Επιπλέον, ευχαριστώ πολύ τον Jan Jantzen για τις κομβικές του υποδείξεις, ώστε να πάρει σάρκα και οστά η αρχική μορφή του αλγορίθμου, αλλά και για την εποικοδομητική κριτική του, όταν βρισκόμουν κοντά στην ολοκλήρωση της διατριβής.

Σημαντική, επίσης, θεωρώ και την συμβολή του διδάσκοντα κ. Παναγιώτη Βασιλάκη, του οποίου η εργασιακή εμπειρία αποτυπώθηκε στις συζητήσεις μας και στην βοήθεια που πρόσφερε με σχόλια επί της αποτελεσματικότητας της δουλειάς μιας διατριβής. Επιπλέον, όντας στενός συνεργάτης του επιβλέποντά μου και κατ' επέκταση έχοντας συχνή επικοινωνία μαζί μου, με συμβούλεψε σε διάφορα στάδια της έρευνάς μου.

Ευχαριστώ πολύ όλους όσους συνεργαστήκαμε στις δημοσιεύσεις που συντέλεσαν στην ολοκλήρωση της παρούσας διατριβής. Τόσο τα μέλη του Εργαστηρίου ΔΕΛΑΠ, Κώστα Μπουλά, Βασίλη Βασιλειάδη, Ευάγγελο Καραμπότση, Δημήτριο Ντάρδα και Χρήστο Κυρικλίδη, όσο και τους φοιτητές Απόστολο Δημουλάκη και Αναστασία Παπαμιχαήλ.

Συντελεστές υπήρξαν επίσης και οι προπτυχιακοί φοιτητές που εκπόνησαν διπλωματικές με συνεπιβλέποντα εμένα: Βασίλης Βουτσαδάκης, Χρήστος Κωνσταντίνου, Κοσμάς Ρουμेलιώτης, Κωνσταντίνος Σηφάκης, Θεοφάνης Σχετάκης, Αλεξάνδρα Ζαρογιάννη και Πέτρος Δεμέστιχας. Τα μικρά κομμάτια βιβλιογραφικής έρευνας που τους ανέθεσα, καθώς και τα πειράματα που εκτέλεσαν στις εργασίες τους συμπλήρωσαν την έρευνά μου σημαντικά. Παρόμοια συνεισφορά πρέπει να αναγνωρίσω στον Νικόλαο-Αλέξανδρο Σωτηρίου (μεταπτυχιακό φοιτητή του ΟΔΙΜ) του οποίου συνεπίβλεψα τη διπλωματική εργασία.

Επιπρόσθετα, χρίζουν αναφοράς οι προπτυχιακοί φοιτητές Εύα Δοβλέτογλου, Βασιλική Μαλλιοπούλου, Γεώργιος Τεπτερής και Ελένη Χατζηλαζάρου, οι οποίοι παρακολούθησαν τις προσφερόμενες διαλέξεις που παρέδωσα και οι οποίες ήταν πέραν του προγράμματος σπουδών τους.

Θα ήθελα να ευχαριστήσω επίσης τους ερευνητές του Ινστιτούτου Πληροφορικής και Τηλεπικοινωνιών του Εθνικού Κέντρου Έρευνας Φυσικών

Επιστημών «Δημόκριτος», στους οποίους παρουσίασα μέρος της έρευνάς μου και οι υποδείξεις και τα σχόλιά τους αποτέλεσαν χρήσιμες κατευθύνσεις για την τελική υποστήριξη της παρούσας διατριβής. Επίσης, θέλω να ευχαριστήσω τον Επίκουρο Καθηγητή κο. Βασίλειο Κούτρα για τις πολύ χρήσιμες και επικοδομητικές παρατηρήσεις του επί του τελικού κειμένου.

Τέλος, ξεχωριστή αναφορά θα κάνω και στην Μαρία Παπαγεωργίου, η οποία με υποστήριξε με κάθε τρόπο καθ' όλη τη διάρκεια της έρευνας που είχε ως αποτέλεσμα την παρούσα διατριβή.

Στην οικογένειά μου

Αλέξανδρος Κ. Τζανέτος

Διδάκτωρ Μηχανικός Οικονομίας και Διοίκησης της Πολυτεχνικής Σχολής του
Πανεπιστημίου Αιγαίου

Copyright © Αλέξανδρος Κ. Τζανέτος, 2020.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Αιγαίου.

Περιεχόμενα

1.	ΕΙΣΑΓΩΓΗ	22
1.1	Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση ως Μέθοδοι Βελτιστοποίησης	23
1.2	Στόχοι.....	24
1.3	Συνεισφορά.....	26
1.4	Δημοσιευμένη Έρευνα.....	27
1.5	Οργάνωση.....	30
2.	ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΕΜΠΝΕΥΣΜΕΝΕΣ ΑΠΟ ΤΗ ΦΥΣΗ.....	32
2.1	Υπολογιστική Νοημοσύνη.....	33
2.2	Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση	34
2.2.1	Ποιες Μέθοδοι μπορούν να θεωρηθούν Εμπνευσμένες από τη Φύση;	36
2.2.2	Προκαταρκτική ανάλυση της βιβλιογραφίας	37
2.3	Νοημοσύνη Σμήνους (ΝΣ).....	39
2.3.1	Μέθοδοι που βασίζονται στον τρόπο εύρεσης της τροφής	40
2.3.2	Μέθοδοι κοινωνικής συμπεριφοράς εντός σμήνους	42
2.3.3	Άλλες συμπεριφορές σμήνους	43
2.4	Νοημοσύνη Εμπνευσμένη από Οργανισμούς (ΝΕΟ)	45
2.4.1	Μέθοδοι βασισμένες στην πανίδα	46
2.4.2	Μέθοδοι βασισμένες στην χλωρίδα	50
2.4.3	Άλλα παραδείγματα	51
2.5	Νοημοσύνη Εμπνευσμένη από Φυσικά Φαινόμενα και Νόμους των Επιστημών (ΝΕΦΦΝΕ)	52
2.5.1	Αλγόριθμοι Εμπνευσμένοι από Φαινόμενα του Σύμπαντος	53
2.5.2	Αλγόριθμοι Εμπνευσμένοι από Νόμους των (Θετικών) Επιστημών	55
2.5.3	Αλγόριθμοι Εμπνευσμένοι από Φυσικά Φαινόμενα	57
2.6	Εφαρμογές Μεθόδων Εμπνευσμένων από τη Φύση	60
2.6.1	Εφαρμογές Αλγορίθμων Εμπνευσμένων από τη Νοημοσύνη Σμήνους	62
2.6.2	Εφαρμογές Αλγορίθμων Εμπνευσμένων από Οργανισμούς	64
2.6.3	Εφαρμογές Αλγορίθμων Εμπνευσμένων από Φυσικά Φαινόμενα και Νόμους των Επιστημών	65
2.7	Συμπεράσματα	66
2.7.1	Συμπεράσματα επί των εφαρμογών των Αλγορίθμων Εμπνευσμένων από τη Φύση.....	66
2.7.2	Αλγόριθμοι με κοινή πηγή έμπνευσης.....	68
2.7.3	Συνοπτικά συμπεράσματα.....	72
3.	ΝΟΗΜΟΝΕΣ ΑΛΓΟΡΙΘΜΟΙ ΕΜΠΝΕΥΣΜΕΝΟΙ ΑΠΟ ΤΗ ΦΥΣΗ ΣΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ	73

3.1	Προβλήματα Βελτιστοποίησης	74
3.1.1	Προβλήματα Βελτιστοποίησης υπό Περιορισμούς (Constrained Optimization Problems)	75
3.1.2	Προβλήματα Βελτιστοποίησης Πολλαπλών Στόχων (Multi-objective Optimization Problems)	75
3.1.3	Γιατί η βελτιστοποίηση είναι μια δύσκολη διαδικασία;	75
3.2	Πολυπλοκότητα Χώρου και Χρόνου	76
3.2.1	Πολυπλοκότητα Χρόνου	76
3.2.2	Κλάση P	80
3.2.3	Κλάση NP	81
3.2.4	Κλάση NP-πληρότητας	83
3.2.5	Πολυπλοκότητα Χώρου	84
3.2.6	Πολυπλοκότητα προβλημάτων βελτιστοποίησης	85
3.3	Εξελικτικοί και Νοήμονες Μετα-Ευρετικοί Αλγόριθμοι	86
3.3.1	Φυσικό Ανάλογο	87
3.3.2	Πληθυσμός και Γενιές	87
3.4	Εξερεύνηση και Εκμετάλλευση	87
3.5	Μηχανισμοί Νοημόνων Αλγορίθμων Εμπνευσμένων από τη Φύση	90
3.5.1	Αναζητητές και Επαναληπτική Διαδικασία	91
3.5.2	Αρχικοποίηση	91
3.5.3	Αναζήτηση στον Χώρο Λύσεων και Εξέλιξη της Λύσης	92
3.5.4	Αξιολόγηση Λύσεων και Ενημέρωση Καλύτερης Λύσης	93
3.5.5	Ψευδοκώδικας και Διάγραμμα Ροής Νοημόνων Αλγορίθμων Εμπνευσμένων από τη Φύση	93
3.6	Προτεινόμενο Πλαίσιο Ανάπτυξης και Βελτίωσης Αλγορίθμων Εμπνευσμένων από τη Φύση	95
3.6.1	Αναζήτηση πηγής έμπνευσης	95
3.6.2	Μέτρηση αποδοτικότητας	97
3.6.3	Αυτορρύθμιση και προσαρμογή	99
3.6.4	Βελτίωση απόδοσης	99
3.7	Συμπεράσματα	100
3.7.1	Σε ποιες περιπτώσεις είναι χρήσιμοι οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση;	100
3.7.2	Υπάρχει ένας αλγόριθμος που υπερνικά τους υπόλοιπους σε απόδοση;	102
3.7.3	Υπάρχει ανάγκη για νέες μεθόδους που να είναι εμπνευσμένες από τη φύση;	102
4.	ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΕΜΠΝΕΥΣΜΕΝΗ ΑΠΟ ΤΟΝ ΗΧΟΕΝΤΟΠΙΣΜΟ	104
4.1	Φυσικό Ανάλογο	105
4.2	Αλγοριθμικό Μοντέλο	106
4.3	Βασικοί Μηχανισμοί Προτεινόμενης Μεθόδου	109

4.3.1	Παράμετρος της Έντασης (Intensity)	109
4.3.2	Ενεργός Ακτίνα (Effective radius)	111
4.3.3	Σάρωση Πλήρους Κύκλου (Full Scan Loop)	111
4.3.4	Μηχανισμός Διόρθωσης Λύσεων (Correction mechanisms)	113
4.4	Παρατηρήσεις	114
4.4.1	Εξερεύνηση και Εκμετάλλευση της ΒΕΗ	114
4.4.2	Πολυπλοκότητα χρόνου προτεινόμενης μεθόδου	115
4.4.3	Ανοικτές παράμετροι και αυτο-ρύθμιση ΒΕΗ	119
4.4.4	Είδη προβλημάτων που λύνει η ΒΕΗ	120
5.	ΥΒΡΙΔΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ	121
5.1	Ανάπτυξη Υβριδικών Σχημάτων	122
5.1.1	Υβριδικά Μοντέλα συνεργασίας μεταξύ των επιμέρους μεθόδων	122
5.1.2	Υβριδικά Μοντέλα βελτιστοποίησης άλλων μεθόδων Υπολογιστικής Νοημοσύνης	125
5.2	Υβριδικά Μοντέλα που περιέχουν τη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό	126
5.2.1	Υβριδικό Σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό – Προσομοιωμένης Ανόπτησης	126
5.2.2	Υβριδικό Σχήμα Γενετικού Αλγορίθμου – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό	129
5.2.3	Υβριδικό Σχήμα Γενετικού Προγραμματισμού – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό	131
5.3	Συμπεράσματα	133
6.	ΓΕΝΙΚΕΣ ΕΦΑΡΜΟΓΕΣ	134
6.1	Συναρτήσεις Ελέγχου Αποδοτικότητας	135
6.1.1	Επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας	135
6.1.2	Αποτελέσματα	140
6.2	Πρότυπα Προβλήματα Μηχανικού	141
6.2.1	Σχεδιασμός Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/compression Spring Design)	141
6.2.2	Σχεδιασμός Συγκολλημένης Δοκού (Welded Beam Design)	142
6.2.3	Αποτελέσματα	143
7.	ΕΦΑΡΜΟΓΕΣ ΣΕ ΠΡΑΓΜΑΤΙΚΑ ΠΡΟΒΛΗΜΑΤΑ.....	146
7.1	Βελτιστοποίηση Χαρτοφυλακίου	147
7.1.1	Προτεινόμενο υβριδικό σχήμα	149
7.1.2	Εισαγωγή Ευρετικής Πληροφορίας	153
7.1.3	Αποτελέσματα Πειραματικής Διαδικασίας	153
7.1.4	Ανάλυση Παραμέτρων της Προτεινόμενης Μεθόδου στο Πρόβλημα Βελτιστοποίησης Χαρτοφυλακίου	155
7.2	Βέλτιστη Διαχείριση Έργων	156

7.2.1	Αντικειμενικές Συναρτήσεις.....	157
7.2.2	Τροποποιήσεις του Αλγορίθμου	159
7.2.3	Επίλυση τεχνητών έργων της βιβλιογραφίας	160
7.2.4	Βέλτιστη Κατανομή Πόρων σε έργο κατασκευής Υποσταθμών Υψηλών Τάσεων	164
7.2.5	Ανάλυση Παραμέτρων της Προτεινόμενης Μεθόδου στο Πρόβλημα Κατανομής Πόρων.....	170
7.3	Βελτιστοποίηση Γραμμών Παραγωγής	171
7.3.1	Μέθοδος Επίλυσης	173
7.3.2	Περίπτωση 1	174
7.3.3	Περίπτωση 2.....	174
7.3.4	Περίπτωση 3.....	175
7.3.5	Αποτελέσματα.....	175
7.4	Βέλτιστη Οικονομική Κατανομή Ενέργειας.....	177
7.4.1	Αντικειμενικοί Στόχοι του Προβλήματος.....	178
7.4.2	Περιορισμοί του Προβλήματος.....	179
7.4.3	Επίδραση του Σημείου Βαλβίδας (Valve-point effect)	180
7.4.4	Λειτουργία πολλαπλών καυσίμων (multi-fuel operation).....	181
7.4.5	Οικονομική Κατανομής Ενέργειας σε ένα τεχνητό σύστημα 10 γεννητριών	182
7.5	Συμπεράσματα	187
8.	ΑΝΑΚΕΦΑΛΑΙΩΣΗ - ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ	189
8.1	Σύνοψη Έρευνας	190
8.2	Κύρια σημεία πρωτοτυπίας της διατριβής	190
8.3	Μελλοντική Έρευνα	191
9.	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	194
ΠΑΡΑΡΤΗΜΑ Ι ΥΠΟΚΑΤΗΓΟΡΙΕΣ ΑΛΓΟΡΙΘΜΩΝ ΕΜΠΝΕΥΣΜΕΝΩΝ ΑΠΟ ΤΗ ΦΥΣΗ		228
ΠΑΡΑΡΤΗΜΑ ΙΙ ΣΥΓΚΕΝΤΡΩΤΙΚΟΙ ΠΙΝΑΚΕΣ ΕΦΑΡΜΟΓΩΝ ΑΛΓΟΡΙΘΜΩΝ ΕΜΠΝΕΥΣΜΕΝΩΝ ΑΠΟ ΤΗ ΦΥΣΗ		244

Κατάλογος Γραφημάτων

Γράφημα 2-1: Πλήθος δημοσιευμένων μεθόδων Εμπνευσμένων από τη Φύση ανά έτος	36
Γράφημα 2-2: Πλήθος δημοσιευμένων μεθόδων σε περιοδικά και συνέδρια σε ετήσια βάση.....	37
Γράφημα 2-3: Δημοσιευμένοι αλγόριθμοι σε περιοδικά αναγνωρισμένων οίκων ανά έτος	38
Γράφημα 2-4: Κυκλικό διάγραμμα αναπαράστασης της ύπαρξης ή μη πραγματικής εφαρμογής στην αρχική εργασία παρουσίασης των μεθόδων	60
Γράφημα 2-5: Εφαρμογή στην αρχική εργασία που παρουσιάζεται μια μέθοδος ανά έτος	61
Γράφημα 2-6: Έτος εμφάνισης των μεθόδων που δεν έχουν εφαρμογές στη βιβλιογραφία	67
Γράφημα 2-7: Ομαδοποίηση των αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση με βάση το πλήθος των δημοσιεύσεων που επιλύουν πραγματικά προβλήματα.....	68
Γράφημα 7-1: Κατανομές ανεξάρτητων πειραμάτων για διάφορα σετ παραμέτρων	155
Γράφημα 7-2: Συγκριτικές κατανομές αποτελεσμάτων απλής και υβριδικής μεθόδου για το κριτήριο G_f	170
Γράφημα 7-3: Συγκριτικές κατανομές αποτελεσμάτων απλής και υβριδικής μεθόδου για το κριτήριο RLI	171

Κατάλογος Εικόνων

Εικόνα 3.1: Χώρος λύσεων της συνάρτησης σφαίρας (Sphere function) για δύο μεταβλητές απόφασης.....	76
Εικόνα 3.2: Παράδειγμα χώρου λύσεων.....	88
Εικόνα 3.3: Παράδειγμα γραφήματος ποιότητας λύσεων	89
Εικόνα 3.4: Χώρος λύσεων παραδείγματος.....	89
Εικόνα 3.5: Συσσώρευση αναζητητών γύρω από τοπικά βέλτιστα.....	90
Εικόνα 4.1: Μηχανική του SONAR	105
Εικόνα 4.2: Η Ένταση του Ήχου συναρτήσει του Εκπέμποντος Σήματος και του Εμβαδού	106
Εικόνα 4.3: Γραφική παράσταση της συνάρτησης $y=e^x$	110
Εικόνα 4.4: Παράδειγμα υπολογισμού του βήματος movement ^d	112
Εικόνα 6.1: Παραδείγματα χώρου λύσεων με ένα και πολλαπλά τοπικά βέλτιστα ..	136
Εικόνα 6.2: Παραδείγματα χώρου λύσεων κυρτής (convex) και μη-κυρτής (non-convex) συνάρτησης	136
Εικόνα 6.3: Σχεδιασμός Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/compression Spring Design).....	141
Εικόνα 6.4: Σχεδιασμός Συγκολλημένης Δοκού (Welded Beam Design).....	142
Εικόνα 7.1: Προφίλ πόρων για το Πρόβλημα 1 (Leu et al., 2000)	161
Εικόνα 7.2: Προφίλ πόρων για το Πρόβλημα 2 (Leu and Hung, 2002)	162
Εικόνα 7.3: Σύγκριση του καλύτερου προφίλ πόρων για το κριτήριο RLI με τα προφίλ της Ενωρίτερης και Αργότερης Έναρξης.....	168
Εικόνα 7.4: Σύγκριση του καλύτερου προφίλ πόρων για το κριτήριο G_f με τα προφίλ της Ενωρίτερης και Αργότερης Έναρξης.....	169
Εικόνα 7.5: Παράδειγμα σειριακής γραμμής παραγωγής (serial production line)	172
Εικόνα 7.6: Η μέθοδος επίλυσης του προβλήματος	173
Εικόνα 7.7: Δοκιμαστικό σύστημα 14 ζυγών του Ινστιτούτου Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE)	177
Εικόνα 7.8: Η επίδραση του φαινομένου της βαλβίδας στη συνάρτηση κόστους	180
Εικόνα 7.9: Η συνάρτηση κόστους του προβλήματος στην περίπτωση πολλαπλών καυσίμων.....	181

Κατάλογος Πινάκων

Πίνακας 2.1: Σύνολο και ποσοστιαία κάλυψη δημοσιεύσεων που προτείνουν Αλγορίθμους Εμπνευσμένους από τη Φύση σε περιοδικά	38
Πίνακας 2.2: Αλγόριθμοι που θεωρούνται ότι τείνουν να καθιερωθούν.....	61
Πίνακας 2.3: Δημοσιευμένες εργασίες Αλγορίθμων Νοημοσύνης Σμήνους ανά κατηγορία προβλημάτων.....	63
Πίνακας 2.4: Δημοσιευμένες εργασίες Αλγορίθμων Εμπνευσμένων από Οργανισμούς ανά κατηγορία προβλημάτων.....	64
Πίνακας 2.5: Δημοσιευμένες εργασίες Αλγορίθμων Εμπνευσμένων από Φυσικά Φαινόμενα και Νόμους των Επιστημών ανά κατηγορία προβλημάτων	65
Πίνακας 3.1: Απαιτούμενος χρόνος εκτέλεσης κάθε βήματος του Ψευδοκώδικα 3.1	78
Πίνακας 3.2: Γνωστές πολυπλοκότητες χρόνου (time complexities).....	78
Πίνακας 4.1: Απαιτούμενος χρόνος εκτέλεσης κάθε βήματος της ΒΕΗ.....	115
Πίνακας 6.1: Ολικά βέλτιστα και πεδία ορισμού των συναρτήσεων ελέγχου αποδοτικότητας	137
Πίνακας 6.2: Απόδοση της προτεινόμενης μεθόδου στις επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας	140
Πίνακας 6.3: Παράμετροι της ΒΕΗ για όλα τα πειράματα.....	143
Πίνακας 6.4: Σύγκριση αποτελεσμάτων για το πρόβλημα του Σχεδιασμού Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη	144
Πίνακας 6.5: Σύγκριση αποτελεσμάτων για το πρόβλημα του Σχεδιασμού Συγκολλημένης Δοκού	145
Πίνακας 7.1: Τιμές παραμέτρων των αλγορίθμων που απαρτίζουν το προτεινόμενο υβριδικό σχήμα.....	154
Πίνακας 7.2: Σύγκριση αποτελεσμάτων προτεινόμενης μεθόδου με άλλες μεθόδους, χωρίς και με ευρετική πληροφορία	154
Πίνακας 7.3: Παράμετροι ΒΕΗ σε κάθε σετ πειραμάτων	156
Πίνακας 7.4: Ενωρίτερη και Αργότερη Έναρξη ενός έργου 13 δραστηριοτήτων.....	157
Πίνακας 7.5: Διαθέσιμες Χρονικές Ενάρξεις κάθε μη-κρίσιμης δραστηριότητας d .	157
Πίνακας 7.6: Αποτελέσματα σε μικρά έργα με γνωστές λύσεις.....	161
Πίνακας 7.7: Αποτελέσματα στα τεχνητά προβλήματα της PSPLib (90 δραστηριοτήτων).....	162

Πίνακας 7.8: Αποτελέσματα στα τεχνητά προβλήματα της PSPLib (120 δραστηριοτήτων).....	163
Πίνακας 7.9: Δεδομένα έργου κατασκευής Υποσταθμών Υψηλών Τάσεων	165
Πίνακας 7.10: Τιμές των παραμέτρων της BEH και του υβριδικού σχήματος BEH-ΠΑ	166
Πίνακας 7.11: Αποτελέσματα BEH και BEH-ΠΑ για το κριτήριο RLI.....	167
Πίνακας 7.12: Αποτελέσματα BEH και BEH-ΠΑ για το κριτήριο G_f	169
Πίνακας 7.13: Αποτελέσματα του Γενετικού Προγραμματισμού	176
Πίνακας 7.14: Αποτελέσματα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό	176
Πίνακας 7.15: Δεδομένα συστήματος 10 γεννητριών	182
Πίνακας 7.16: Συντελεστές του κόστους καυσίμου (fuel cost coefficients) και εκπομπών (emission coefficients) για κάθε γεννήτρια, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας	182
Πίνακας 7.17: Συντελεστές του κόστους καυσίμου (fuel cost coefficients) ανά τύπο καυσίμου, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας	183
Πίνακας 7.18: Περιορισμοί και εκδοχές του προβλήματος Οικονομικής Κατανομής Ενέργειας για το σύστημα 10 γεννητριών	184
Πίνακας 7.19: Κατανομή Ενέργειας σύμφωνα με τα αποτελέσματα της μεθόδου για κάθε πρόβλημα	185
Πίνακας 7.20: Συγκριτικά αποτελέσματα στο πρόβλημα της Οικονομικής Κατανομής Φορτίου.....	186
Πίνακας 7.21: Τιμές των παραμέτρων της BEH και των συγκρινόμενων μεθόδων .	187

Κατάλογος Σχημάτων

Σχήμα 3-1: Διαγραμματική απεικόνιση ντετερμινιστικών και στοχαστικών μεθόδων	79
Σχήμα 3-2: Γράφος Προβλήματος Χαμιλτόνιου Μονοπατιού	81
Σχήμα 3-3: Σχέση μεταξύ των κλάσεων πολυπλοκότητας χρόνου και πολυπλοκότητας χώρου	85
Σχήμα 3-4: Μοτίβο μεθόδων εμπνευσμένων από Φαινόμενα της Φύσης	94
Σχήμα 5-1: Υβριδικό μοντέλο όπου οι επιμέρους αλγόριθμοι εκτελούν διαφορετικές διεργασίες.....	123
Σχήμα 5-2: Υβριδικό μοντέλο όπου οι επιμέρους αλγόριθμοι εκτελούν διεργασίες του κύριου αλγόριθμου.....	124
Σχήμα 5-3: Υβριδικό μοντέλο επίλυσης πολλών στόχων.....	125
Σχήμα 5-4: Δενδρική Δομή Γενετικού Προγραμματισμού	132
Σχήμα 7-1: Διαγραμματική απεικόνιση προσέγγισης βελτιστοποίησης χαρτοφυλακίου με υβριδικό σχήμα αλγορίθμων	150
Σχήμα 7-2: Προτεινόμενο υβριδικό σχήμα για Βελτιστοποίηση Χαρτοφυλακίου....	151

Κατάλογος Ψευδοκώδικων

Ψευδοκώδικας 3.1: Διεργασία με βρόγχο «Όσο».....	77
Ψευδοκώδικας 3.2: Αλγόριθμος διερεύνησης ύπαρξης μονοπατιού σύνδεσης αρχικού και τελικού κόμβου σε έναν γράφο G, a, g	81
Ψευδοκώδικας 3.3: Αλγόριθμος διερεύνησης Χαμιλτόνιου Μονοπατιού.....	82
Ψευδοκώδικας 3.4: Γενική Μορφή Αλγορίθμων Εμπνευσμένων από τη Φύση	93
Ψευδοκώδικας 4.1: Αλγόριθμος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization).....	107
Ψευδοκώδικας 5.1: Προσομοιωμένη Ανόπτηση (Simulated Annealing).....	127
Ψευδοκώδικας 5.2: Υβριδικό Σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization) με Προσομοιωμένη Ανόπτηση (Simulated Annealing)	128
Ψευδοκώδικας 5.3: Υβριδικό Σχήμα Γενετικού Αλγορίθμου - Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization).....	131

Συντομογραφίες

Συντομογραφία	Πλήρες κείμενο
ΑΕΦ	Αλγόριθμοι Εμπνευσμένοι από τη Φύση
ΒΕΗ	Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό
ΒΣΣ	Βελτιστοποίηση Σμήνους Σωματιδίων
ΓΑ	Γενετικός Αλγόριθμος
ΝΕΟ	Νοημοσύνη Εμπνευσμένη από Οργανισμούς
ΝΕΦΦΝΕ	Νοημοσύνη Εμπνευσμένη από Φυσικά Φαινόμενα και Νόμους των Επιστημών
ΝΣ	Νοημοσύνη Σμήνους

Χρήσιμη Ορολογία

Αγγλική	Ελληνική
Agent	Αναζητητής
Artificial Intelligence	Τεχνητή Νοημοσύνη
Computational Intelligence	Υπολογιστική Νοημοσύνη
Economic Load Dispatch	Οικονομική Κατανομή Φορτίου
Evolutionary Algorithms	Εξελικτικοί Αλγόριθμοι
Generations	Γενιές
Hybrid Scheme	Υβριδικό Σχήμα
Nature-inspired Algorithms	Αλγόριθμοι Εμπνευσμένοι από τη Φύση
NP-complete	NP-πλήρης
NP-hard	NP-δυσμενής
Population	Πληθυσμός
Swarm Intelligence	Νοημοσύνη Σμήνους

1. Εισαγωγή

1.1 Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση ως Μέθοδοι Βελτιστοποίησης

Ο χώρος της Τεχνητής Νοημοσύνης (Artificial Intelligence) παρουσιάζει ερευνητικά ένα συνεχώς αυξανόμενο ενδιαφέρον, μετρώντας ήδη πάνω από 60 χρόνια σταδιακής ανάπτυξης (McCarthy, Minsky and Rochester, 1955). Ο στόχος της ήταν ανέκαθεν να προγραμματίσει με τέτοιο τρόπο τις μηχανές, ώστε να μπορούν να προβούν σε ανθρώπινες λειτουργίες. Από την πρώτη εφαρμογή, όπου ο Alan Turing προσπάθησε να αποκρυπτογραφήσει τα μηνύματα των Γερμανών κατά τον Β' Παγκόσμιο Πόλεμο, μέχρι και σήμερα, ο χώρος της Τεχνητής Νοημοσύνης έχει επεκταθεί σε πολλούς τομείς και έχει προσφέρει λύσεις σε προβλήματα της καθημερινότητας του ανθρώπου.

Λόγω της ικανότητας των υπολογιστών να επεξεργάζονται συντριπτικά περισσότερες πληροφορίες απ' ότι ο άνθρωπος σε μια μονάδα χρόνου, η Τεχνητή Νοημοσύνη κερδίζει συνεχώς έδαφος και πλέον θεωρείται μια από τις σημαντικότερες γνωστικές περιοχές. Αρκετές πτυχές της Τεχνητής Νοημοσύνης έχουν εμφανιστεί όλες αυτές τις δεκαετίες, όπου μερικές εξ αυτών είναι η Ρομποτική, η Επεξεργασία της Φυσικής Γλώσσας (Natural Language Process), τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks), η Μηχανική Μάθηση (Machine Learning), η Βαθιά Μάθηση (Deep Learning) κ.ά. Γνωστές εφαρμογές της Τεχνητής Νοημοσύνης είναι η Αναγνώριση Προτύπων (Pattern Recognition), η Πρόβλεψη (Prediction) και η Κατηγοριοποίηση (Classification). Ωστόσο, τα τελευταία χρόνια, μια υποκατηγορία μεθόδων της Τεχνητής Νοημοσύνης, η επονομαζόμενη Υπολογιστική Νοημοσύνη (Computational Intelligence) έχει χρησιμοποιηθεί ευρέως στην επίλυση περίπλοκων προβλημάτων βελτιστοποίησης.

Γνωστά προβλήματα βελτιστοποίησης είναι τα προβλήματα Λήψης Αποφάσεων (Decision Making), στα οποία στόχος είναι η εύρεσης της βέλτιστης απόφασης ή της σειράς βέλτιστων αποφάσεων που πρέπει να ακολουθήσει ένας λήπτης απόφασης. Τέτοια προβλήματα εμφανίζονται σε πολλά πεδία εφαρμογών που συγκαταλέγονται σε εκείνα που θα κληθεί να ασχοληθεί ένας Μηχανικός: μερικά παραδείγματα είναι η Διοίκηση Έργων (Project Management), η Επιχειρησιακή Έρευνα (Operational Research), η Χρηματοοικονομική Μηχανική (Financial Engineering) και τα Ενεργειακά Συστήματα (Energy Systems).

Τα παραπάνω πεδία εφαρμογών εμφανίζουν δεκάδες προβλήματα βελτιστοποίησης, τα οποία έχουν επιλυθεί κατά καιρούς από μεθόδους Μαθηματικού Προγραμματισμού. Ωστόσο, με την ανάπτυξη της Τεχνητής Νοημοσύνης, εμφανίστηκαν μέθοδοι που μπόρεσαν να παρέχουν καλύτερες λύσεις από τις μεθόδους Μαθηματικού Προγραμματισμού. Τεχνητά Νευρωνικά Δίκτυα, Γενετικός Προγραμματισμός και Εξελικτικοί Αλγόριθμοι χρησιμοποιήθηκαν και χρησιμοποιούνται για την επίλυση προβλημάτων βελτιστοποίησης.

Τα τελευταία χρόνια έχουν κάνει την εμφάνισή τους κάποιες μέθοδοι Εξελικτικών Αλγορίθμων, οι οποίες είναι εμπνευσμένες από στρατηγικές που ακολουθούν οργανισμοί στη φύση για την επίλυση βασικών προβλημάτων τους, όπως η εύρεση τροφής και η επιλογή βιότοπων. Γι' αυτό έχουν ονομαστεί Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση (Nature Inspired Computational Methods). Στόχος των μεθόδων αυτών είναι η προσομοίωση της στρατηγικής από την οποία έχουν εμπνευστεί με τέτοιο τρόπο που να δημιουργηθεί μια αλγοριθμική (βηματική δηλαδή) διαδικασία που να επιλύει προβλήματα βελτιστοποίησης.

Ένα καίριο ζήτημα που προκύπτει είναι το γεγονός ότι συνεχώς εμφανίζονται νέες τέτοιες μέθοδοι, η πληθώρα των οποίων δείχνει να μοιάζει σε μεγάλο βαθμό με την μεθοδολογία κάποιων καθιερωμένων (established) μεθόδων αντί να προτείνονται νέες τεχνικές. Κατ' επέκταση δεν προσφέρουν βελτίωση στον τρόπο προσέγγισης των προβλημάτων ή γρηγορότερες και αυτοματοποιημένες μεθόδους, απλά κακές αναπαραγωγές παλαιότερων κουβαλώντας μαζί και τις ατέλειές (drawbacks) τους. Η παραγωγή μιας νέας μεθόδου πρέπει να εστιάζει στο να ελαχιστοποιεί την παρέμβαση του χρήστη και να έχει μελετηθεί με τέτοιο τρόπο που να μπορεί να προσεγγίσει διαφορετικά είδη προβλημάτων, χωρίς να απαιτεί τεράστια παραμετροποίηση ή εκ νέου προγραμματισμό.

Στην παρούσα διατριβή έχουν μελετηθεί σε βάθος τα μοντέλα των Εμπνευσμένων από τη Φύση μεθόδων. Επίσης, έχουν μελετηθεί και επιλυθεί διάφορα προβλήματα βελτιστοποίησης και προτείνεται μια νέα μέθοδος εμπνευσμένη από ένα φαινόμενο που παρατηρείται στον φυσικό κόσμο.

1.2 Στόχοι

Οι στόχοι της διατριβής παρουσιάζονται με τη μορφή ερευνητικών ερωτημάτων, όπως αυτά τέθηκαν στα διάφορα στάδια της έρευνας.

Ερώτημα 1. Πόσοι και ποιοι είναι οι αλγόριθμοι που είναι εμπνευσμένοι από τη φύση; Τι είδους προβλήματα μπορούν να λύσουν;

Αρχικός στόχος της διατριβής ήταν η καθολική χαρτογράφηση του χώρου των μεθόδων που είναι Εμπνευσμένες από τη Φύση. Μεμονωμένα, υπάρχουν κάποιες εργασίες που έχουν συλλέξει ένα μέρος των μεθόδων, αλλά πουθενά δεν έχει παρουσιασθεί το σύνολο αυτών.

Παράλληλα, στόχος αυτής της βιβλιογραφικής έρευνας ήταν η εύρεση των προβλημάτων που μπορούν να λύσουν οι μέθοδοι που είναι εμπνευσμένες από τη Φύση, ενώ εξίσου σημαντικός στόχος ήταν και η διερεύνηση του πλήθους εξ αυτών των μεθόδων, οι οποίες επιλύουν πραγματικά προβλήματα. Τέτοιου τύπου παρατηρήσεις απουσιάζουν από τη βιβλιογραφία, καθώς έχουν γίνει μόνο περιορισμένες έρευνες που παρουσιάζουν κάποια από τα προβλήματα που έχουν λύσει οι πιο γνωστοί αλγόριθμοι εξ όσων είναι Εμπνευσμένοι από τη Φύση.

Ερώτημα 2. Ποια είναι εκείνα τα χαρακτηριστικά που καθιστούν έναν Αλγόριθμο Εμπνευσμένο από τη Φύση αποδοτικό και ποιος είναι κατά περίπτωση ο καλύτερος όλων από πλευράς απόδοσης;

Το επόμενο βήμα που κλήθηκε να απαντήσει η έρευνα στα πλαίσια της διατριβής είναι αν υπάρχει κάποιος αλγόριθμος Εμπνευσμένος από τη Φύση που είναι αποδοτικότερος όλων για όλα τα είδη προβλημάτων. Από την άλλη, εάν δεν υπάρχει ένας αλγόριθμος αυτής της κατηγορίας Υπολογιστικής Νοημοσύνης που να υπερσχύει όλων των υπολοίπων, στόχος ήταν να καθοριστούν εκείνα τα χαρακτηριστικά που καθιστούν αποδοτικό έναν τέτοιο αλγόριθμο σε σύγκριση με άλλες συμβατικές μεθόδους βελτιστοποίησης.

Ερώτημα 3. Υπάρχει ένα πλαίσιο στο οποίο μπορεί να στραφεί κανείς, εφόσον θέλει να αναπτύξει έναν Αλγόριθμο που είναι Εμπνευσμένος από τη Φύση;

Κύριος στόχος εδώ ήταν να δομηθεί ένα γενικό πλαίσιο αρχών για τον χώρο της Νοημοσύνης που είναι Εμπνευσμένη από τη Φύση (Nature Inspired Intelligence). Επιπρόσθετα, ένας εκ των πρωταρχικών στόχων ήταν να προταθεί ένα πλαίσιο εργασίας (framework) για την ανάπτυξη και την βελτίωση των μεθόδων που είναι Εμπνευσμένες από τη Φύση. Στη βιβλιογραφία δεν υπάρχει διατυπωμένο ένα σχετικό πλαίσιο.

Ερώτημα 4. Μπορεί να προταθεί ένας νέος αλγόριθμος εμπνευσμένος από τη φύση, ικανός να επιλύει προβλήματα βελτιστοποίησης υψηλής πολυπλοκότητας;

Με δεδομένο τις παρατηρήσεις που σημειώθηκαν δουλεύοντας το προαναφερθέν πλαίσιο, κύριος στόχος της παρούσας διατριβής ήταν να αναπτυχθεί μια νέα μέθοδος, η οποία να έχει υψηλές και ανταγωνιστικές επιδόσεις, ελαχιστοποιώντας παράλληλα την ανθρώπινη παρέμβαση, η οποία θα απαιτηθεί για παραμετροποίηση της μεθόδου.

Για να διερευνηθεί η αποτελεσματικότητα της προτεινόμενης μεθοδολογίας σε πραγματικά προβλήματα Βελτιστοποίησης και Λήψης Αποφάσεων, τα οποία αντιμετωπίζει ένας Μηχανικός, μελετήθηκαν διάφορα υψηλής πολυπλοκότητας προβλήματα βελτιστοποίησης, τα οποία διαφέρουν ως προς τον τρόπο επίλυσής τους και ως προς τους περιορισμούς που τα κάνουν δύσκολα να επιλυθούν από συμβατικές μεθόδους.

Παράλληλα, ένας άλλος στόχος της εφαρμογής της προτεινόμενης μεθόδου σε προβλήματα βελτιστοποίησης υψηλής πολυπλοκότητας υπήρξε η καταγραφή των προβλημάτων που αντιμετωπίζουν οι μέθοδοι που είναι εμπνευσμένες από τη φύση, αλλά και η αντιμετώπισή τους.

Ερώτημα 5. Πώς θα επηρεαστεί η απόδοση του προτεινόμενου αλγόριθμου εάν αναπτυχθούν υβριδικές μέθοδοι, στις οποίες θα συνεργάζεται με άλλους αλγόριθμους εμπνευσμένους από τη φύση ή άλλες μεθόδους Υπολογιστικής Νοημοσύνης;

Αναπτύχθηκαν υβριδικά σχήματα (hybrid schemes) τα οποία βασίζονται στην προτεινόμενη μέθοδο, που παρουσιάζεται παρακάτω στη διατριβή, τα οποία να βελτιώνουν την απόδοση της. Αυτή η διαδικασία αποσκοπεί στην ανάδειξη της σημασίας της δημιουργίας υβριδικών μεθόδων έναντι της συνεχούς εμφάνισης νέων μεθόδων, οι οποίες δεν λύνουν πραγματικά προβλήματα.

1.3 Συνεισφορά

Στα πλαίσια της διατριβής, η έρευνα που πραγματοποιήθηκε, κατέληξε στα κάτωθι σημαντικότερα σημεία συνεισφοράς:

1. Έγινε για πρώτη φορά μια πλήρης χαρτογράφηση του χώρου των μεθόδων που είναι εμπνευσμένες από τη φύση (Κεφάλαιο 2). Επιπλέον, προτάθηκε ένας τρόπος κατηγοριοποίησης για αυτές τις μεθόδους, η οποία βασίστηκε σε κοινά χαρακτηριστικά των αλγορίθμων, αλλά και του πεδίου από το οποίο αντλούν την έμπνευσή τους.
2. Μελετήθηκε η χρησιμότητα των αλγορίθμων των Εμπνευσμένων από τη Φύση (Nature Inspired) τόσο ανά υπο-κατηγορία βάσει πεδίου έμπνευσης, όσο και συνολικά (Κεφάλαιο 2). Επιπλέον, αναλύθηκε το είδος των προβλημάτων στο οποίο εφαρμόζονται οι αλγόριθμοι αυτοί σε συνάρτηση με τα ιδιαίτερα χαρακτηριστικά που έχουν ανά υποκατηγορία. Σκοπός της μελέτης ήταν να απαντηθεί το δεύτερο σκέλος του πρώτου ερευνητικού ερωτήματος που διατυπώνεται παραπάνω: τι είδους προβλήματα μπορούν να λύσουν αυτοί οι αλγόριθμοι;
3. Δομήθηκε ένα πλαίσιο εργασίας (framework) στην ανάπτυξη, την βελτίωση και στην αξιολόγηση Νοημόνων Μεθόδων που είναι Εμπνευσμένες από τη Φύση (Κεφάλαιο 3), καθώς επίσης δόθηκε η γενική διαγραμματική απεικόνιση και ο γενικός ψευδοκώδικας των αλγορίθμων αυτών. Το πλαίσιο εργασίας εστιάζει κατά κύριο λόγο στα χαρακτηριστικά εκείνα που καθιστούν έναν αλγόριθμο αποδοτικό, με στόχο να απαντάει στο τρίτο ερευνητικό ερώτημα που διατυπώθηκε παραπάνω.
4. Παρουσιάζεται μια νέα μέθοδος βελτιστοποίησης (Κεφάλαιο 4), η οποία ανήκει στην ευρύτερη κατηγορία των Νοημόνων Μεθόδων που είναι Εμπνευσμένες από τη Φύση (Nature Inspired Intelligent Methods). Η μέθοδος αυτή δομήθηκε με στόχο την ελαχιστοποίηση παραμετροποίησης από τον χρήστη, αλλά και με σκοπό την αντιμετώπιση του προβλήματος του εγκλωβισμού σε τοπικά βέλτιστα (local optima trapping). Αυτός ο στόχος επιτεύχθηκε μέσα από την εφαρμογή μηχανισμών αυτο-ρύθμισης (auto-tuning) που εισήχθησαν στη διαδικασία της μεθόδου.
5. Οι κύριοι μηχανισμοί της προτεινόμενης μεθόδου αποτελούν σημαντική συνεισφορά στο πεδίο της Επιστήμης Υπολογιστών (Computer Science) και της

Υπολογιστικής Νοημοσύνης (Computational Intelligence), καθώς μπορούν να χρησιμοποιηθούν σε άλλες νοήμονες μεθόδους, ώστε να αντισταθμιστεί το πρόβλημα της πρόωρης σύγκλισης (premature convergence) και να επιτευχθεί η κατάλληλη ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation).

6. Επιπρόσθετα, μελετήθηκαν και αναπτύχθηκαν υβριδικές μέθοδοι που περιέχουν τον προτεινόμενο αλγόριθμο βελτιστοποίησης. Η σύμπραξη με άλλες μεθόδους, όπου δημιούργησε υβριδικά σχήματα, είχε βασικό σκοπό να αυξηθεί η απόδοση της προτεινόμενης μεθόδου, καθώς επίσης να επιλυθούν αποδοτικότερα ορισμένες εφαρμογές, οι οποίες έχουν υψηλό βαθμό δυσκολίας.
7. Τέλος, μελετήθηκαν διάφορα προβλήματα Λήψης Απόφασης από τον ευρύτερο χώρο των προβλημάτων Μηχανικού, ώστε να προταθούν τροποποιημένες εκδοχές της προτεινόμενης μεθόδου για διάφορα είδη προβλημάτων (Κεφάλαια 6 και 7).

1.4 Δημοσιευμένη Έρευνα

Συνοπτικά, στα πλαίσια της διατριβής δημοσιεύθηκαν ή έχουν υποβληθεί προς δημοσίευση οι παρακάτω εργασίες:

σε περιοδικά:

1. Tzanetos, A., & Dounias, G. (2017). Nature Inspired Optimization Algorithms Related to Physical Phenomena and Laws of Science: A Survey. *International Journal on Artificial Intelligence Tools*, 26(06), 1750022.
2. Tzanetos, A., & Dounias, G. (2018). Sonar inspired optimization (SIO) in engineering applications. *Evolving Systems*, 1-9.
3. Tzanetos, A., Vassiliadis, V., & Dounias, G. (2018). Boosting the performance of hybrid Nature-Inspired algorithms: Application from the financial optimization domain. *Logic Journal of the IGPL*.
4. Ntardas, D., Tzanetos, A., & Dounias, G. (2020). Resource Leveling Optimization in Construction Projects of High Voltage Substations Using Nature-Inspired Intelligent Evolutionary Algorithms. *International Journal of Electrical and Computer Engineering*, 14(1), 6-13.
5. Konstantinou, Ch., Tzanetos, A., & Dounias, G. (submitted). Tackling Portfolio Optimization with a Hybrid Scheme of Genetic Algorithm and Sonar Inspired Optimization Application.
6. Tzanetos, A., & Dounias, G. (submitted). Nature Inspired Optimization Algorithms or Simply Variations of Metaheuristics?

7. Tzanetos, A., & Dounias, G. (submitted). A comprehensive analysis of the applications and variations of Sonar Inspired Optimization.

σε συνέδρια με κριτές:

8. Tzanetos, A., & Dounias, G. (2017, August). A New Metaheuristic Method for Optimization: Sonar Inspired Optimization. In International Conference on Engineering Applications of Neural Networks (pp. 417-428). Springer, Cham.
9. Tzanetos, A., Kyriklidis, C., Papamichail, A., Dimoulakis, A., & Dounias, G. (2018, July). A Nature Inspired metaheuristic for Optimal Leveling of Resources in Project Management. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (p. 17). ACM.
10. Boulas, K., Tzanetos, A., & Dounias, G. (2018, July). Acquisition of approximate throughput formulas for serial production lines with parallel machines using intelligent techniques. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (p. 18). ACM.
11. Tzanetos, A., & Dounias, G. (2019, May). Sonar Inspired Optimization in Energy Problems Related to Load and Emission Dispatch. In International Conference on Learning and Intelligent Optimization (pp. 268-283). Springer, Cham.

και ως κεφάλαια βιβλίων:

12. Tzanetos, A., & Dounias, G. (2020, to appear). A comprehensive survey on the applications of Swarm Intelligence and Bio-inspired evolutionary strategies (to appear). In: G.A. Tsihrintzis and L.C. Jain, eds. Machine Learning Paradigms: Advances in Theory and Applications of Deep Learning, Intelligent Systems Reference Library. Springer, Cham.

Ενώ αναρτήθηκε και μια Ερευνητική Αναφορά (Research Report) στον ιστοχώρο του Εργαστηρίου Διοίκησης Επιχειρήσεων και Λήψης Αποφάσεων:

- Tzanetos, A. and Dounias, G., 2019. *Observations on Nature Inspired Intelligent Techniques*. [Research Report] Chios, Greece: Management and Decision Engineering Laboratory (MDE-Lab) University of the Aegean, School of Engineering, Dept. of Financial and Management Engineering. Available at: <http://mde-lab.aegean.gr/images/stories/docs/reportnii2019.pdf>.

Άλλες εργασίες που έγιναν στα πλαίσια της γενικότερης ερευνητικής δραστηριότητας του Εργαστηρίου Διοίκησης Επιχειρήσεων και Λήψης Αποφάσεων (ΔΕΛΑΠ):

1. Tzanetos, A., Vassiliadis, V., & Dounias, G. (2017, June). A Novel Hybrid Nature-Inspired Scheme for Solving a Financial Optimization Problem. In International Conference on Hybrid Artificial Intelligence Systems (pp. 161-172). Springer, Cham.

2. Karampotsis, E., Boulas, K., Tzanetos, A., Androvitsaneas, V. P., Gonos, I. F., Dounias, G., & Stathopoulos, I. A. (2017, June). Computational Intelligence Techniques for Modelling the Critical Flashover Voltage of Insulators: From Accuracy to Comprehensibility. In International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (pp. 295-301). Springer, Cham.

Επιπλέον, πραγματοποιήθηκε η συνεπίβλεψη των κάτωθι διπλωματικών εργασιών προπτυχιακών φοιτητών του Π.Π.Σ. του τμήματος Μηχανικών Οικονομίας και Διοίκησης:

Φοιτητής	Τίτλος Διπλωματικής	Έτος ολοκλήρωσης
Βουτσαδάκης Βασίλειος	Νοήμονες Μέθοδοι Εμπνευσμένες από το Φυσικό Κόσμο για την Επίλυση Προβλημάτων Βελτιστοποίησης από το Χώρο Οικονομίας και Διοίκησης	2017
Κωνσταντίνου Χρήστος	Εφαρμογή της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό στη Διαχείριση Οικονομικού Χαρτοφυλακίου	2018
Κοσμάς Ρουμελιώτης	Ανάλυση Παραμέτρων της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization): Μελέτη περίπτωσης στο πρόβλημα της Οικονομικής Κατανομής Φορτίου (Economic Load Dispatch)	2020
Κωνσταντίνος Σηφάκης	Εφαρμογή βασισμένη στην Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (Sonar Inspired Optimization) για την Βέλτιστη Κατανομή Πόρων σε Μεγάλα Έργα	2020
Φανούρης Σχετάκης	Βελτιστοποίηση Χαρτοφυλακίου με Χρήση Υβριδικών Νοημόνων Αλγορίθμων Εμπνευσμένων από τη Φύση	υπό εξέλιξη
Αλεξάνδρα Ζαρογιάννη	Αντιμετώπιση Προβλημάτων Εξισορρόπησης Πόρων Τηλεπικοινωνιακού Έργου με Μεθόδους Υπολογιστικής Νοημοσύνης	υπό εξέλιξη
Πέτρος Δεμέστιχας	Εφαρμογή του Αλγορίθμου Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό στα Προβλήματα Χρονοπρογραμματισμού	υπό εξέλιξη
Ανδρέας Βουβαλίδης	Εφαρμογή Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό σε προβλήματα ταξινόμησης (Classification) από τον χώρο της υγείας	υπό εξέλιξη

αλλά και του μεταπτυχιακού φοιτητή:

Νικόλαος – Επίλυση Προβλημάτων από τον Χώρο της
Αλέξανδρος Διοίκησης Έργων με τη Χρήση Νοημόνων 2020
Σωτηρίου Μεθόδων

Τέλος, υπήρξε συνεργασία με τον μεταπτυχιακό φοιτητή Δημήτριο Ντάρδα, ο οποίος εκπόνησε την έρευνά του για το Μεταπτυχιακό Δίπλωμα Μηχανικού Οικονομίας και Διοίκησης μέσω Έρευνας με τίτλο «Διοίκηση Έργων και Νοήμονες Υπολογιστικές Τεχνικές». Για τις ανάγκες του διπλώματός του χρησιμοποίησε ως μέθοδο βελτιστοποίησης την μέθοδο που προτείνεται στην παρούσα διατριβή.

1.5 Οργάνωση

Η Διατριβή διαρθρώνεται ως ακολούθως:

Το Κεφάλαιο 1 έχει μια σύντομη εισαγωγή αναφορικά με την χρήση των Υπολογιστικών Μεθόδων που είναι Εμπνευσμένες από τη Φύση σε προβλήματα Βελτιστοποίησης και Λήψης Αποφάσεων. Επιπλέον, αναφέρονται επιγραμματικά οι στόχοι, αλλά και η συνεισφορά της διατριβής στον ευρύτερο τομέα της Τεχνητής Νοημοσύνης. Τέλος, αναφέρονται οι δημοσιευμένες εργασίες που έγιναν στα πλαίσια της διατριβής, καθώς επίσης και οι διπλωματικές που συνεπίβλεψε ο υποψήφιος.

Στο Κεφάλαιο 2 γίνεται μια εκτενής βιβλιογραφική επισκόπηση των μεθόδων Υπολογιστικής Νοημοσύνης και ειδικότερα αυτών που είναι Εμπνευσμένες από τη Φύση. Αφού επιμεριστούν οι μέθοδοι σε υποκατηγορίες βάσει των χαρακτηριστικών τους, αναλύονται τα κύρια χαρακτηριστικά κάθε υποκατηγορίας και παρουσιάζονται τα αποτελέσματα της έρευνας που έγινε για τις εφαρμογές στις οποίες έχουν χρησιμοποιηθεί αυτές οι υπολογιστικές μέθοδοι.

Στο Κεφάλαιο 3 επεξηγούνται βασικές έννοιες που διέπουν τη λειτουργία των νοημόνων αλγορίθμων και ειδικότερες περιπτώσεις που αφορούν τους μετα-ευρετικούς αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση.

Η μέθοδος που προτείνεται στην παρούσα διατριβή, παρουσιάζεται λεπτομερώς στο Κεφάλαιο 4. Όλοι οι μηχανισμοί που έχουν εφαρμοσθεί στο κύριο μέρος του αλγόριθμου, καθώς και οι τροποποιήσεις (modifications) που έγιναν κατά τη διάρκεια της έρευνας στα πλαίσια της διατριβής αναλύονται και επεξηγούνται σε ξεχωριστές υποενότητες.

Το Κεφάλαιο 5 κάνει μια αναφορά στους υβριδικούς αλγόριθμους και γενικότερα στις υβριδικές μεθόδους Υπολογιστικής Νοημοσύνης, καθώς εισάγει τον αναγνώστη στη λειτουργία και την επιχειρηματολογία γύρω από την συνεργασία της προτεινόμενης μεθόδου με άλλες μεθόδους Υπολογιστικής Νοημοσύνης με απώτερο σκοπό την περαιτέρω βελτίωση των λύσεων που παράγονται. Επιπλέον, τα υβριδικά σχήματα που αναπτύχθηκαν στα πλαίσια της διατριβής παρουσιάζονται και αναλύεται ο ρόλος των επιμέρους μεθόδων που τα απαρτίζουν.

Στο 6^ο Κεφάλαιο παρουσιάζεται η πειραματική διαδικασία του ελέγχου λειτουργικότητας του αλγορίθμου σε γνωστές μαθηματικές συναρτήσεις, μαζί με τα αποτελέσματά της. Επιπλέον, υποστηρίζεται η απόδοση της μεθόδου με Ανάλυση Ευαισθησίας και επίλυση απλών προβλημάτων βελτιστοποίησης, που μπορούν να λυθούν με ακρίβεια από μαθηματικές μεθόδους, όπως ο Μαθηματικός Προγραμματισμός.

Το Κεφάλαιο 7 παρουσιάζει τα αποτελέσματα του προτεινόμενου αλγόριθμου σε πραγματικά προβλήματα, τα οποία ανήκουν στον ευρύτερο χώρο της Μηχανικής της Διοίκησης.

Στο Κεφάλαιο 8 συνοψίζονται τα συμπεράσματα της διατριβής. Επίσης, προτείνονται μελλοντικές ερευνητικές κατευθύνσεις που θα μπορούσε να ακολουθήσει η περαιτέρω ενασχόληση με την προτεινόμενη μέθοδο, αλλά και γενικότερα η έρευνα πάνω στη λειτουργικότητα των αλγορίθμων αυτής της λογικής.

Τέλος, στο Κεφάλαιο 9 παρατίθεται η βιβλιογραφία που χρησιμοποιήθηκε στην παρούσα διατριβή.

2. Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση

2.1 Υπολογιστική Νοημοσύνη

Η ικανότητα των υπολογιστών να επεξεργαστούν συντριπτικά περισσότερες πληροφορίες απ' ότι ο άνθρωπος σε μια μονάδα χρόνου, οδήγησε τον κλάδο της Τεχνητής Νοημοσύνης στην ανάπτυξη υπολογιστικών συστημάτων με νοημοσύνη παρόμοια με την ανθρώπινη. Εστίασε, δηλαδή, στην δημιουργία κανόνων μέσα στα υπολογιστικά συστήματα που αναφέρονται παραπάνω, ώστε να μπορούν να ληφθούν σχετικές αποφάσεις. Αυτός ο κλάδος της Τεχνητής Νοημοσύνης ονομάστηκε Υπολογιστική Νοημοσύνη (Computational Intelligence) (Pedrycz, 1997; Chen, 1999).

Αρχικά, εμφανίστηκε η Μηχανική Μάθηση (Machine Learning), στα πλαίσια της οποίας αναπτύχθηκαν νοήμονες μέθοδοι που κατέστησαν δυνατό να μοντελοποιηθεί μια σχέση μεταξύ μεταβλητών απόφασης και μεταβλητής στόχου. Η Μηχανική Μάθηση (Michalski, Carbonell and Mitchell, 1983, 1986; Kodratoff and Michalski, 1990; Mitchell, 1997) ήρθε να συμπληρώσει την αδυναμία των μοντέλων πρόβλεψης της Στατιστικής, καθώς γενίκευσε τις πληροφορίες που παρείχαν ιστορικά δεδομένα και παρείχε αξιόπιστα προβλεπτικά μοντέλα.

Τα Συστήματα Ασαφών Κανόνων (Fuzzy rule based systems) υπήρξαν μια από τις πρώτες απόπειρες εισαγωγής νοημοσύνης σε υπολογιστικά συστήματα. Η συγκεκριμένη κατηγορία Υπολογιστικής Νοημοσύνης μπορεί να περιγραφεί σαν ένα είδος λεξιλογίου με συντακτικό και σημασιολογία, το οποίο μπορεί να αποτυπώσει ένα είδος της πληροφορίας που χρειάζεται για να λυθεί ένα πρόβλημα. Συνήθως είναι απαραίτητη η συμβολή ενός ειδικού, που κατανοεί την πληροφορία αυτή.

Στην προσπάθεια των ερευνητών να αποδώσουν την σχέση μεταξύ κανόνων και απόφασης, μελετήθηκε ο τρόπος λειτουργίας του ανθρώπινου εγκεφάλου και ειδικότερα η διαδρομή που ακολουθούσε μια πληροφορία στο νευρωνικό δίκτυο του ανθρώπινου εγκεφάλου. Έτσι προτάθηκαν τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) (Rosenblatt, 1959; Widrow and Hoff, 1960; Werbos, 1974; Rojas, 2013).

Παρόλο που τα Τεχνητά Νευρωνικά Δίκτυα αποτελούν την πρώτη μέθοδο Υπολογιστικής Νοημοσύνης που αντλούσε την έμπνευσή της από τη Φύση, ο Holland (1975) με την ομάδα του, που πρότειναν τον πρώτο Γενετικό Αλγόριθμο (Genetic Algorithm), ο οποίος και θεωρείται η απαρχή των Μεθόδων που είναι Εμπνευσμένες από τη Φύση (Nature-inspired Methods). Επόμενο ήταν να δημιουργηθεί το ερώτημα: μπορούν άλλα φαινόμενα να αποτελέσουν έμπνευση για νέα Νοήμονα Υπολογιστικά Συστήματα;

Αρχικά εμφανίστηκε ο όρος Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms) και Εξελικτικός Υπολογισμός (Evolutionary Computing) στη βιβλιογραφία (Back, Fogel and Michalewicz, 1997; Eiben and Smith, 2003), όπου αναπτύχθηκαν μέθοδοι οι οποίες είναι εμπνευσμένες από την θεωρία της εξέλιξης του Δαρβίνου, όπως ο

Γενετικός Αλγόριθμος (Holland, 1992b) και ο αλγόριθμος της Διαφορικής Εξέλιξης (Differential Evolution) (Storn and Price, 1997). Γνωστή μέθοδος Εξελικτικού Υπολογισμού είναι ο Γενετικός Προγραμματισμός (Genetic Programming) (Koza, 1992), ο οποίος βασίστηκε στην θεωρία των Γενετικών Αλγορίθμων.

Σε αντίθεση με τις υπόλοιπες μεθόδους Υπολογιστικής Νοημοσύνης, οι οποίες εφαρμόζονται κατά κύριο λόγο σε προβλήματα κατηγοριοποίησης (classification), συσταδοποίησης (clustering) και πρόβλεψης (forecasting), οι Εξελικτικοί Αλγόριθμοι χρησιμοποιήθηκαν ευρέως για την επίλυση προβλημάτων βελτιστοποίησης, των οποίων η λύση δεν ήταν προφανής ή τα προβλήματα ήταν δυναμικά με αποτέλεσμα να αλλάζει συνεχώς η βέλτιστη λύση. Η διαφοροποίηση στα πεδία εφαρμογών κάθε είδους μεθόδων Υπολογιστικής Νοημοσύνης δημιούργησε την ανάγκη ανάπτυξης τεχνικών που διάφορες μέθοδοι Υπολογιστικής Νοημοσύνης συνεργάζονταν. Οι τεχνικές αυτές ονομάζονται Υβριδικά Σχήματα (Hybrid Schemes) και αποτελούνται από δύο ή περισσότερες μεθόδους Υπολογιστικής Νοημοσύνης. Τέλος, έχουν αναπτυχθεί εξίσου αρκετά Προσαρμοστικά Νοήμονα Συστήματα (Adaptive Intelligent Systems) (Hayes-Roth, 1995).

2.2 Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση

Η επιτυχία των Εξελικτικών Αλγορίθμων δημιούργησε πρόσφορο έδαφος για την ανάπτυξη της Προσομοιωμένης Ανόπτησης (Simulated Annealing) (Kirkpatrick, Gelatt and Vecchi, 1983), της Αναζήτησης Στοχαστικής Διάχυσης (Stochastic Diffusion Search) (J. M. Bishop, 1989), της Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization) (Eberhart and Kennedy, 1995) και του Αλγόριθμου Επιλογής Κλώνου (Clonal Selection Algorithm) (de Castro and von Zuben, 2000). Η εμφάνιση αυτών των αλγορίθμων, οι οποίοι εμπνεύστηκαν από φυσικά φαινόμενα (ανόπτηση), συμπεριφορές εντόμων (μυρμήγκια) και εξελικτικές θεωρίες (κλώνοι), συντέλεσε ώστε από τις αρχές του 2000 να χρησιμοποιηθεί ευρέως στη βιβλιογραφία ο όρος Νοήμονες Μέθοδοι Εμπνευσμένες από Φαινόμενα της Φύσης (Nature Inspired Intelligence methods). Από τις αρχές του 2000 ξεκίνησαν να εμφανίζονται σταδιακά Αλγόριθμοι Εμπνευσμένοι από τη Φύση (Nature Inspired Algorithms), με τον αριθμό τους να αυξάνεται εκθετικά μέχρι και σήμερα.

Μια πρώτη συλλογή των δημοσιευμένων μεθόδων που ήταν Εμπνευσμένες από τη Φύση (Nature Inspired) παρουσιάστηκε στην εργασία των Fister Jr. et al. (2013). Οι συγγραφείς χώρισαν τους αλγόριθμους σε τέσσερις (4) μεγάλες κατηγορίες βάσει της πηγής έμπνευσης κάθε αλγόριθμου:

1. Νοημοσύνη Σμήνους (Swarm Intelligence)
2. Βιο-εμπνευσμένα σχήματα (Bio-inspired)
3. Εμπνευσμένα από Φυσική και Χημεία (Physics and Chemistry) και
4. Διάφορα (Others)

Η πρώτη κατηγορία περιείχε τις μεθόδους που ανήκουν στην επονομαζόμενη Νοημοσύνη Σμήνους (Swarm Intelligence), η οποία είναι ευρέως γνωστή για μεθόδους που προσομοιώνουν τη συλλογική συμπεριφορά ομάδων έμβιων όντων, όπως η Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization) και η Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization).

Στη δεύτερη κατηγορία είχαν συμπεριλάβει μεθόδους που ήταν εμπνευσμένες από συμπεριφορές ζώων, πτηνών, ψαριών, αλλά και φυτών, που ωστόσο δεν εμφανίζουν κάποια συλλογική συμπεριφορά.

Η τρίτη κατηγορία περιείχε εκείνες τις μεθόδους, των οποίων η έμπνευση στηρίζεται σε φαινόμενα που περιγράφονται από τις θετικές επιστήμες (Φυσική, Χημεία κλπ).

Στην τελευταία κατηγορία, οι συγγραφείς της προαναφερόμενης εργασίας είχαν συλλέξει εκείνες τις μεθόδους που δεν μπορούσαν να κατηγοριοποιηθούν σε μια εκ των τριών προηγούμενων κατηγοριών.

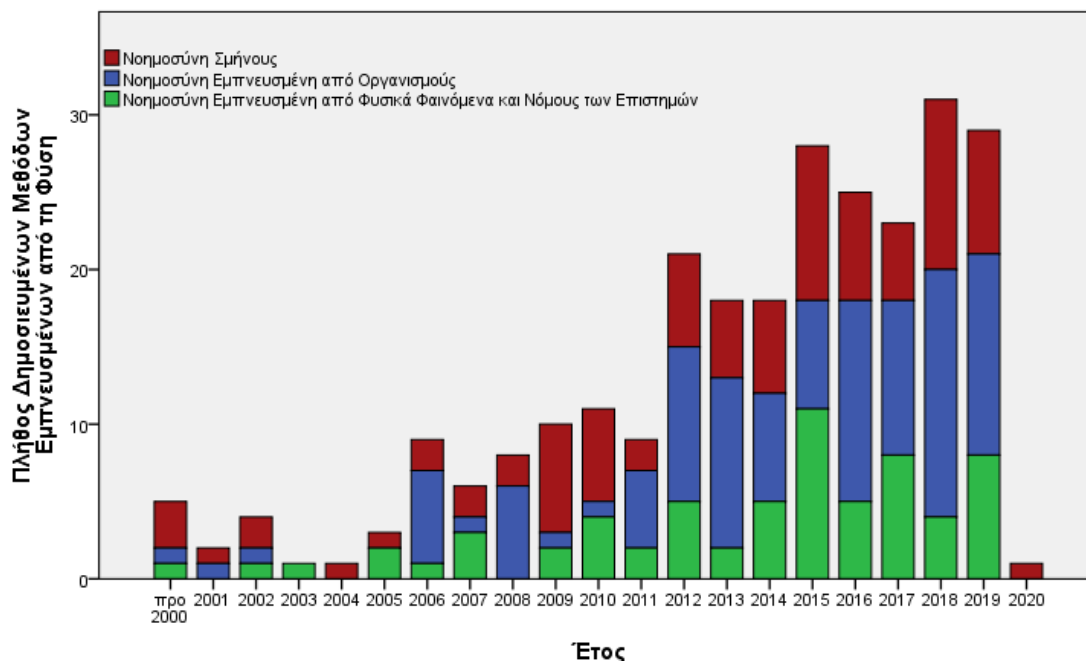
Μελετώντας κανείς τον χώρο των υπολογιστικών μεθόδων που είναι εμπνευσμένες από τη φύση, θα παρατηρούσε ότι η συνεχής εμφάνιση νέων μεθόδων καθιστά δύσκολο τον προσδιορισμό του πλήθους των αλγορίθμων που απαρτίζουν αυτή την περιοχή των νοημόνων μεθόδων. Επίσης, πουθενά στη βιβλιογραφία δεν υπήρχαν συγκεντρωμένοι όλοι οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση. Ωστόσο, πρωταρχικής σημασίας είναι να προσδιοριστεί ποιοι αλγόριθμοι μπορούν να θεωρηθούν εμπνευσμένοι από τη φύση.

Στα πλαίσια της παρούσας διατριβής έγινε μια διεξοδική έρευνα των μεθόδων που συντελούν τον χώρο των προσεγγίσεων Εμπνευσμένων από τη Φύση, ώστε να συλλεχθούν όλοι οι αλγόριθμοι του εν λόγω χώρου και να μελετηθούν οι εφαρμογές τους. Η έρευνα αυτή είχε σαν αποτέλεσμα δύο δημοσιεύσεις βιβλιογραφικού χαρακτήρα (Tzanetos and Dounias, 2017b, 2020). Στις δημοσιεύσεις αυτές αναδιαμορφώθηκε η κατηγοριοποίηση των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση (Nature Inspired Algorithms) ως ακολούθως:

1. Νοημοσύνη Σμήνους (Swarm Intelligence)
2. Αλγόριθμοι Εμπνευσμένοι από Οργανισμούς (Organisms-based)
3. Αλγόριθμοι Εμπνευσμένοι από Φυσικά Φαινόμενα και Νόμους των Επιστημών (Physical Phenomena and Laws of Science)

Στην εργασία (Tzanetos and Dounias, 2020) διαχωρίστηκαν οι μέθοδοι Νοημοσύνης Σμήνους από όσες δεν πληρούσαν τα χαρακτηριστικά που θέτει η βιβλιογραφία για έναν αλγόριθμο αυτής της κατηγορίας. Επιπλέον, οι μέθοδοι που αφαιρέθηκαν από την παλαιότερη κατηγοριοποίηση των Fister Jr. et al. (2013) μαζί με όσες μεθόδους αντλούν έμπνευση από οργανισμούς (όπως ζώα, φυτά, έντομα, μικρο-οργανισμούς κλπ) συνέθεσαν την κατηγορία των Αλγορίθμων που είναι

Εμπνευσμένοι από Οργανισμούς. Το όνομα Βιο-εμπνευσμένα σχήματα (Bio-inspired) δεν διατηρήθηκε, διότι στην βιβλιογραφία παρατηρείται η τάση να συγχέεται ο όρος nature-inspired με τον όρο bio-inspired, αναφερόμενοι και οι δύο στους Αλγορίθμους που είναι Εμπνευσμένοι από τη Φύση.



Γράφημα 2-1: Πλήθος δημοσιευμένων μεθόδων Εμπνευσμένων από τη Φύση ανά έτος

Οι δημοσιευμένες μέθοδοι κάθε κατηγορίας ανά έτος παρουσιάζονται στο Γράφημα 2-1 παραπάνω. Όπως φαίνεται από το εν λόγω γράφημα, ένα μεγάλο ποσοστό των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση προέρχεται από την κατηγορία εκείνων που είναι Εμπνευσμένοι από Οργανισμούς. Ωστόσο, αξιοσημείωτη είναι η εκθετική τάση που παρατηρείται στον αριθμό των μεθόδων. Την τελευταία πενταετία εμφανίζονται σχεδόν 25 νέοι αλγόριθμοι ανά έτος.

2.2.1 Ποιες Μέθοδοι μπορούν να θεωρηθούν Εμπνευσμένες από τη Φύση;

Η Νοημοσύνη που είναι Εμπνευσμένη από τη Φύση (Nature Inspired Intelligence - NII) περιγράφει αλγόριθμους και μεθόδους, που μοντελοποιούν φαινόμενα της φύσης, τον τρόπο εύρεσης τροφής και την κοινωνική συμπεριφορά μεταξύ μελών σμήνους (swarm) ή στρατηγικές που έχουν αναπτύξει ζώα, έντομα και φυτά με σκοπό να επιλύσουν προβλήματα επιβίωσης και διαβίωσης. Ένας πιο περιεκτικός ορισμός για το τι θεωρείται Νοημοσύνη Εμπνευσμένη από τη Φύση δίδεται από τους Steer, Wirth and Halgamuge (2009), οι οποίοι ορίζουν ότι ο όρος φύση (nature) αναφέρεται “στο κομμάτι του φυσικού κόσμου (physical universe) το οποίο δεν είναι αποτέλεσμα εκούσιας παρέμβασης του ανθρώπου”.

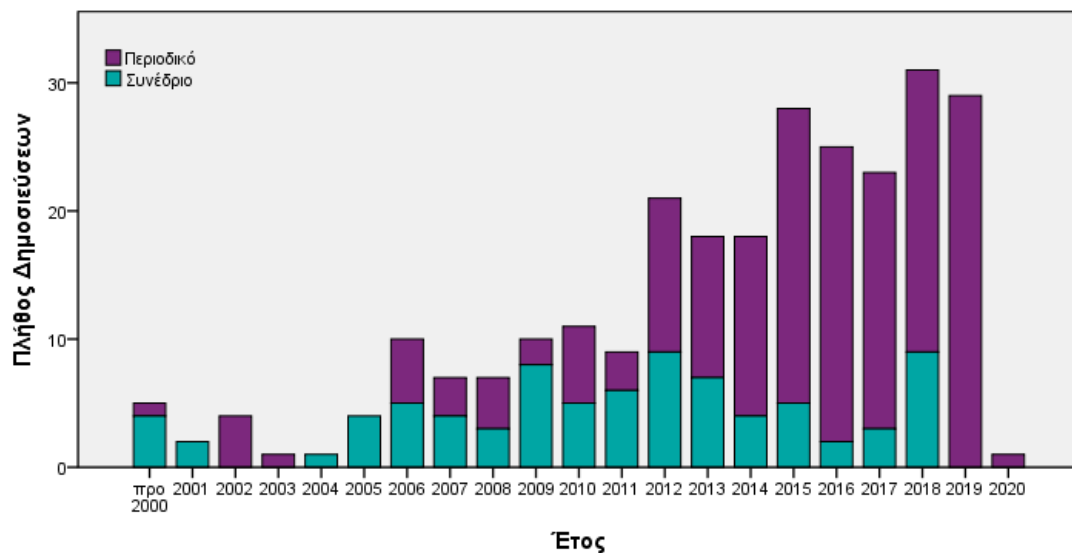
Κατ’ επέκταση, η τέταρτη κατηγορία αλγορίθμων σύμφωνα με τους Fister Jr. et al. (2013), η οποία περιέχει μεθόδους όπως ο Αλγόριθμος Ιμπεριαλιστικού Ανταγωνισμού (Imperialist Competitive Algorithm) (Atashpaz-Gargari and Lucas,

2007) και ο Αλγόριθμος Ανώτερης Λίγκας Πετοσφαίρισης (Volleyball League Premier Algorithm) (Moghdani and Salimifard, 2018), δεν συμπεριλαμβάνεται στην ευρύτερη κατηγορία των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση. Πρόσφατες εργασίες συμφωνούν προς αυτή την κατεύθυνση (Neme and Hernández, 2009; Yang, 2017; Alatas, 2017; Kumar and Kulkarni, 2019).

2.2.2 Προκαταρκτική ανάλυση της βιβλιογραφίας

Μερικά ενδιαφέροντα συμπεράσματα μπορούν να εξαχθούν μελετώντας κάποια βασικά ερωτήματα, όπως:

- α) πού δημοσιεύονται οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση
- β) ποια περιοδικά με κριτές και ποια διεθνή συνέδρια δημοσιεύουν εργασίες σχετικές με τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση



Γράφημα 2-2: Πλήθος δημοσιευμένων μεθόδων σε περιοδικά και συνέδρια σε ετήσια βάση

Στο Γράφημα 2-2 φαίνεται το πλήθος των δημοσιεύσεων που παρουσιάστηκαν Αλγόριθμοι Εμπνευσμένοι από τη Φύση σε περιοδικά και σε συνέδρια, σε ετήσια βάση. Παρατηρείται ότι η συντριπτική πλειοψηφία των αλγορίθμων, ειδικά μετά το 2012, πρωτοεμφανίζεται σε περιοδικά. Ωστόσο, σκόπιμο είναι να ληφθεί υπόψη και η αναγνωσιμότητα του περιοδικού, ώστε να είναι πιο ολοκληρωμένη η εικόνα.

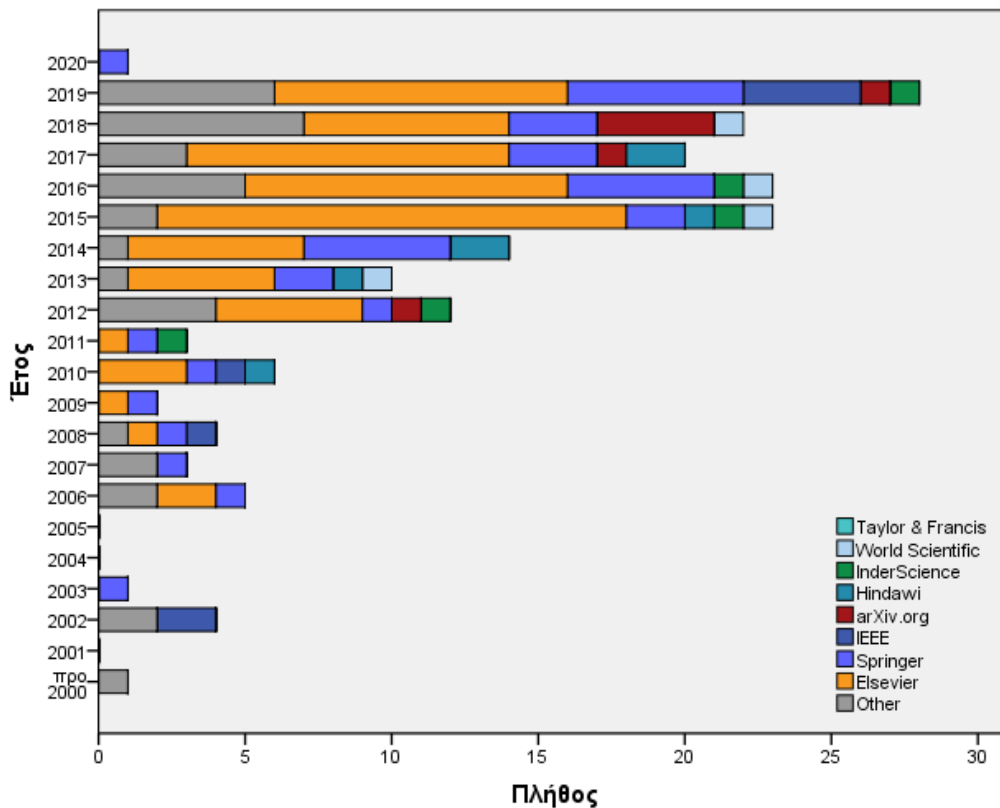
Όπως φαίνεται στον Πίνακα 2.1, σχεδόν το 60% των αλγορίθμων που εξετάζονται στην παρούσα διατριβή έχει δημοσιευθεί σε περιοδικά των γνωστών οίκων Elsevier και Springer, με την IEEE να έρχεται τρίτη με μεγάλη απόκλιση από τους δύο προαναφερόμενους οίκους. Σχεδόν 20% των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση έχουν εμφανιστεί σε περιοδικά μικρότερων οίκων.

Στο Γράφημα 2-3 φαίνεται το πλήθος των δημοσιευμένων αλγορίθμων στα περιοδικά των οίκων που αναφέρονται στον Πίνακα 2.1 ανά έτος. Σημαντική παρατήρηση είναι ότι τα τελευταία χρόνια αυξάνεται σταδιακά ο αριθμός των μεθόδων που εμφανίζονται σε περιοδικά χαμηλότερης αναγνωσιμότητας.

Πίνακας 2.1: Σύνολο και ποσοστιαία κάλυψη δημοσιεύσεων που προτείνουν Αλγορίθμους Εμπνευσμένους από τη Φύση σε περιοδικά

Εκδοτικός Οίκος	Σύνολο Δημοσιεύσεων	Ποσοστό (%)
Elsevier	79	42,9
Springer	35	19,0
IEEE	8	4,3
Hindawi	7	3,8
arXiv.org	7	3,8
InderScience	5	2,7
World Scientific	4	2,2
Wiley	3	1,6
Άλλος Οίκος	34	18,68

Στις επόμενες ενότητες του παρόντος κεφαλαίου ακολουθεί μια αναλυτική παρουσίαση και χαρτογράφηση του χώρου των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση. Εφόσον δεν εμπίπτει στα ερευνητικά ενδιαφέροντα του αναγνώστη, μπορεί να μεταβεί στην ενότητα 2.6 (σελ. 60), που παρουσιάζονται συνοπτικά οι εφαρμογές των Μεθόδων που είναι Εμπνευσμένες από τη Φύση, ή στην Ενότητα 2.7 (σελ. 66), που βρίσκονται τα συμπεράσματα του παρόντος κεφαλαίου.



Γράφημα 2-3: Δημοσιευμένοι αλγόριθμοι σε περιοδικά αναγνωρισμένων οίκων ανά έτος

2.3 Νοημοσύνη Σμήνους (ΝΣ)

Ο όρος Νοημοσύνη Σμήνους (Swarm Intelligence ή SI) προτάθηκε από τους Beni and Wang (1993) με σκοπό να αποδώσουν την συλλογική συμπεριφορά μεταξύ ρομποτικών συστημάτων. Παρόλα αυτά, ο ίδιος όρος χρησιμοποιείται για να περιγράψει τους αλγόριθμους που οι υποψήφιες λύσεις (ή αναζητητές, όπως αναφέρονται παρακάτω) συνεργάζονται με σκοπό την εύρεση της καλύτερης λύσης του προβλήματος που καλείται ο αλγόριθμος να επιλύσει.

Ο πρώτος αλγόριθμος Νοημοσύνης Σμήνους ήταν η Αναζήτηση Στοχαστικής Διάχυσης (Stochastic Diffusion Search), η οποία παρουσιάστηκε από τον J. M. Bishop (1989) και είναι εμπνευσμένη από την επικοινωνία ενός-προς-ένα (one-by-one) ενός είδους μυρμηγκιών που ονομάζονται *Leptothorax acervorum*. Το 1992, ο Marco Dorigo πρότεινε την Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization) με το όνομα Σύστημα Μυρμηγκιών (Ant System) στην Διδακτορική του Διατριβή (Dorigo, 1992). Τρία χρόνια μετά, οι Eberhart and Kennedy (1995) πρότειναν την Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization ή PSO) για την επίλυση προβλημάτων συνεχούς χώρου (continuous space optimization problems). Οι τρεις αυτές μέθοδοι είναι ευρέως γνωστές και έχουν εμπνεύσει σε μεγάλο βαθμό τους ερευνητές να δοκιμάσουν και να αναπτύξουν νέες μεθόδους που να είναι εμπνευσμένες από τη φύση.

Στις υποενότητες που ακολουθούν γίνεται διαχωρισμός των μεθόδων ΝΣ σε τρεις υποκατηγορίες με βάση τα χαρακτηριστικά τους: τροφοληψία (Foraging), κοινωνική συμπεριφορά (Social Behavior) και άλλες συμπεριφορές σμήνους (Other Swarm Behavior). Οι μέθοδοι Τροφοληψίας (Foraging) περιγράφουν στρατηγικές που ακολουθούν ομάδες ζώων για να βρουν σημεία τροφής. Στην υποκατηγορία των Μεθόδων Κοινωνικής Συμπεριφοράς (Social Behavior) οι αλγόριθμοι μοντελοποιούν κοινωνικά χαρακτηριστικά, όπως η ιεραρχία εντός αγέλης. Οποιαδήποτε άλλη συμπεριφορά παρατηρείται σε μεγάλες ομάδες (σμήνη, αγέλες, κοπάδια κλπ.) και δεν μπορεί να συμπεριληφθεί στις δύο προαναφερόμενες υποκατηγορίες συμπεριλαμβάνεται στην υποκατηγορία των Άλλων Συμπεριφορών Σμήνους (Other Swarm Behavior).

Στην εργασία των Fister Jr. et al. (2013), τόσο ο Αλγόριθμος Εικονικών Μυρμηγκιών (Virtual Ant Algorithm) (Yang, Lees and Morley, 2006) όσο και ο αλγόριθμος Εικονικών Μελισσών (Virtual Bees) (Yang, 2005) συμπεριλαμβάνονται στις μεθόδους ΝΣ. Ωστόσο, βάσει του ορισμού του τι μπορεί να θεωρηθεί Αλγόριθμος Εμπνευσμένος από τη Φύση που δίνεται στην υποενότητα 2.2.1 παραπάνω, στην παρούσα διατριβή παραλείπονται και οι δύο αυτές μέθοδοι από οποιαδήποτε κατηγορία Αλγορίθμων Εμπνευσμένων από τη Φύση παρουσιάζεται παρακάτω.

2.3.1 Μέθοδοι που βασίζονται στον τρόπο εύρεσης της τροφής

Πολλές μέθοδοι Νοημοσύνης Σμήνους (ΝΣ) είναι εμπνευσμένες από τον τρόπο εύρεσης τροφής (foraging strategies) ζώων, πουλιών, εντόμων ή ψαριών. Η Βελτιστοποίηση Αποικίας Μυρμηγκιών είναι ο πρώτος αλγόριθμος ΝΣ, ο οποίος βασίστηκε στη συνεργατική στρατηγική οργανισμών (μυρμηγκιών στην συγκεκριμένη περίπτωση) με σκοπό την εύρεση τροφής. Ο επόμενος αλγόριθμος με παρόμοια στρατηγική παρουσιάστηκε το 2002 και είναι βασισμένος στην τροφοληψία (foraging) των βακτηρίων (Passino, 2002).

Το 2004, ο αλγόριθμος της Κυψέλης (Beehive algorithm) προτάθηκε από τους Wedde, Farooq and Zhang (2004). Ένα χρόνο αργότερα, οι Teodorovic and Dell'Orco (2005) παρουσίασαν τη Βελτιστοποίηση Αποικίας Μελισσών (Bee Colony Optimization) με σκοπό να λύσουν Προβλήματα Πλανόδιου Πωλητή (TSP). Ένας ακόμη αλγόριθμος που εμπνεύστηκε από τη συλλογική συμπεριφορά των μελισσών παρουσιάστηκε από τους Pham et al. (2006), ενώ την επόμενη χρονιά οι Karaboga and Basturk (2007) παρουσίασαν έναν εκ των πιο γνωστών αλγορίθμων: την Τεχνητή Αποικία Μελισσών (Artificial Bee Colony).

Ένας εξίσου ευρέως γνωστός αλγόριθμος με εφαρμογές σε πολλά πεδία προτάθηκε από τον Yang (2010), επονομαζόμενος Αλγόριθμος Νυχτερίδας (Bat algorithm). Την ίδια χρονιά, παρουσιάστηκε ένας ακόμη αλγόριθμος εμπνευσμένος από τη συμπεριφορά των μελισσών (Akbari, Mohammadi and Ziarati, 2010). Μια ακόμη μέθοδος που επίσης παρουσιάστηκε το 2010 εμπνεύστηκε από τον τρόπο που κυνηγούν τα ζώα και πήρε το σχετικό όνομα Αναζήτηση Κυνηγιού (Hunting Search) (Oftadeh, Mahjoob and Shariatpanahi, 2010).

Παρόλο που η Βελτιστοποίηση Σμήνους Κατσαρίδων (Cockroach Swarm Optimization) παρουσιάστηκε για πρώτη φορά στο 2^ο Διεθνές Συνέδριο Μηχανικής Υπολογιστών και Τεχνολογίας (2nd International Conference on Computer Engineering and Technology) το 2010, η εργασία ανακλήθηκε και η επίσημη πρώτη εμφάνιση του εν λόγω αλγορίθμου θεωρείται η εργασία του 2011 (Zhaohui and Haiyan, 2011).

Το 2012, ο αλγόριθμος του Τυφλού, Γυμνού Τυφλοπόντικα (Blind, naked mole-rat algorithm) προτάθηκε από τους Shirzadi and Bagheri (2012). Εμπνεόμενοι από τον τρόπο εύρεσης τροφής της βρωμούσας (eurygaster), οι Ahmadi, Salehi and Karimi (2012) πρότειναν τον ομώνυμο αλγόριθμο, τον οποίο και εφάρμοσαν σε ένα πρόβλημα διαχωρισμού γραφημάτων (graph partitioning problem). Μια ακόμη τεχνική εμπνευσμένη από τις μέλισσες παρουσίασαν οι R. D. Maia, L. N. de Castro and W. M. Caminhas (2012), ενώ το ίδιο έτος προτάθηκε η Αναζήτηση Λύκου (Wolf Search) από τους Tang et al. (2012).

Το 2013, οι Subramanian, Sekar and Subramanian (2013) πρότειναν τον Αλγόριθμο του Αφρικανικού Αγριόσκυλου (African Wild Dog Algorithm) ως μια μέθοδο βελτιστοποίησης προβλημάτων μηχανικού. Την ίδια χρονιά, οι Gheraibia and

Moussaoui (2013) παρουσίασαν τον Αλγόριθμο Βελτιστοποίησης Αναζήτησης Πιγκουίνων (Penguins Search Optimization Algorithm) και οι Wang, Zhu and Huang (2013) ανέπτυξαν τον αλγόριθμο Βελτιστοποίησης της Πασχαλίτσας (Seven-spot Ladybird Optimization).

Μια ακόμη μέθοδος βελτιστοποίησης με πολύ καλά αποτελέσματα παρουσιάστηκε από τους Mirjalili, Mirjalili and Lewis (2014), γνωστή ως Βελτιστοποιητής Γκριζού Λύκου (Grey Wolf Optimizer). Την ίδια χρονιά, οι Meng et al. (2014) παρουσίασαν την Βελτιστοποίηση Σμήνους από Κότες (Chicken Swarm Optimization). Και οι δύο προαναφερθείσες τεχνικές εφαρμόστηκαν σε πρότυπα προβλήματα μηχανικού στην αρχική τους εργασία. Το 2015, οι Wu and Wu (2015) προτείνουν την Εξέλιξη Σμήνους Κατσαρίδων (Cockroach Swarm Evolution) η οποία είχε πρωτοεμφανιστεί με το όνομα Γενετικός Αλγόριθμος Κατσαρίδας (Cockroach Genetic Algorithm) από τους ίδιους συγγραφείς (Wu and Wu, 2013). Επίσης το 2015, οι S. Arora and S. Singh (2015) πρότειναν τον Αλγόριθμο Βελτιστοποίησης Πεταλούδας (Butterfly Optimization Algorithm) υπό το όνομα Αλγόριθμος Πεταλούδας (Butterfly Algorithm), δίνοντας του το τελικό του όνομα μερικά χρόνια αργότερα (Arora and Singh, 2019).

Το 2016, οι Yong et al. (2016) παρουσιάζουν τον Αλγόριθμο Βελτιστοποίησης Σμήνους Δελφινιών (Dolphin Swarm Optimization Algorithm) και οι Meng et al. (2016) προτείνουν τη Βελτιστοποίηση Σμήνους Πουλιών (Bird Swarm Optimization). Εμπνεόμενοι από τα κοράκια, οι Brabazon, Cui and O'Neill (2016) προτείνουν τον Αλγόριθμο Βελτιστοποίησης Κουρνιάσματος Κόρακα (Raven Roosting Optimization Algorithm). Οι Mirjalili and Lewis (2016) προτείνουν έναν ακόμη αλγόριθμο, γνωστό ως Αλγόριθμο Βελτιστοποίησης Φάλαινας (Whale Optimization Algorithm).

Το 2017, οι Dhiman and Kumar (2017) προτείνουν τον Βελτιστοποιητή της Ύαινας με Βούλες (Spotted Hyena Optimizer) εφαρμόζοντάς τον σε πρότυπα προβλήματα μηχανικού. Την ίδια χρονιά, ο Αλγόριθμος Σμήνους Φαλαινών (Whale Swarm Algorithm) παρουσιάστηκε από τους Zeng, Gao and Li (2017), ενώ ένα ακόμη ψάρι έγινε πηγή έμπνευσης αλγορίθμου, όπου οι Zaldívar et al. (2018) πρότειναν τον αλγόριθμο Μπαρμπουνιού με Κίτρινη Ράχη (Yellow Saddle Goatfish algorithm) την επόμενη χρονιά. Ο τρόπος που βρίσκουν την τροφή τους οι σουρικάτες (meerkats) ενέπνευσε τους Al-Obaidi, Abdullah and Ahmed (2018) να παρουσιάσουν τον Αλγόριθμο Φυλής Σουρικατών (Meerkat Clan Algorithm) ως μια μέθοδο επίλυσης Προβλημάτων Πλανόδιου Πωλητή (TSP).

Η πιο πρόσφατη μέθοδος που έχει παρουσιαστεί ονομάζεται Βελτιστοποιητής Ιστιοφόρου Ξιφία (Sailfish Optimizer) και προτάθηκε από τους Shadravan, Najji and Bardsiri (2019), όπου εφαρμόστηκε σε πρότυπα προβλήματα μηχανικού.

2.3.2 Μέθοδοι κοινωνικής συμπεριφοράς εντός σμήνους

Μια ακόμη πηγή έμπνευσης για τους ερευνητές αποτέλεσε η κοινωνική δομή και συμπεριφορά μεταξύ των μελών του κοπαδιού ή της αγέλης. Η κοινωνική συμπεριφορά των ψαριών ενέπνευσε τους Li, Shao and Qian (2002) και πρότειναν τον αλγόριθμο Σμήνους Ψαριών (Fish-swarm algorithm). Τέσσερα χρόνια αργότερα, οι Krishnanand and Ghose (2006) πρότειναν τη Βελτιστοποίηση Βασισμένη σε Σμήνος Λαμπυρίδων (Glowworm Swarm-based Optimization). Η κοινωνική συμπεριφορά των κατσαρίδων αποτέλεσε έμπνευση για τους T. C. Havens et al. (2008), οι οποίοι πρότειναν την Βελτιστοποίηση Επιδρομής Κατσαρίδων (Roach Infestation Optimization).

Παρόλο που η πλειοψηφία των μεθόδων που είναι εμπνευσμένες από μέλισσες βρίσκεται στην προηγούμενη υποκατηγορία των αλγορίθμων ΝΣ, το 2009, οι Comellas and Martinez-Navarro (2009) μοντελοποίησαν την κοινωνική δομή ενός είδους μέλισσας, γνωστό και ως Βομβίνος (Bumblebees algorithm). Την ίδια χρονιά προτάθηκαν δύο εκ των πιο γνωστών Αλγορίθμων Εμπνευσμένων από τη Φύση: η Αναζήτηση Κούκου (Cuckoo Search) (X. Yang and Suash Deb, 2009) και ο Αλγόριθμος Πυγολαμπίδας (Firefly Algorithm) (Yang, 2009). Επίσης το 2009, προτάθηκε η Βελτιστοποίηση Συντρόφου Δελφινιού (Dolphin Partner Optimization) από τους Y. Shiqin, J. Jianjun and Y. Guangxing (2009), ενώ ο Chen (2009a) πρότεινε έναν αλγόριθμο ΝΣ εμπνεόμενος από τις ακρίδες (locust).

Το 2010, οι Hedayatzadeh et al. (2010) πρότειναν τη Βελτιστοποίηση Αποικίας Τερμιτών (Termite Colony Optimization). Η δημιουργία κοπαδιών (herding behavior) των Krill (γαριδόμορφων ζώων) ενέπνευσε τους Gandomi and Alavi (2012) στη δημιουργία του ομώνυμου αλγόριθμου.

Εμπνεόμενοι από τις αράχνες, οι Cuevas et al. (2013) πρότειναν τη Βελτιστοποίηση Κοινωνικής Αράχνης (Social Spider Optimization). Το ίδιο έτος, προτάθηκε ο Αλγόριθμος Βελτιστοποίησης Σμήνους Χελιδονιών (Swallow Swarm Optimization Algorithm) από τους Neshat, Sepidnam and Sargolzaei (2013). Ένα είδος πιθήκου ενέπνευσε τους Bansal et al. (2014) να αναπτύξουν τον Αλγόριθμο Βελτιστοποίησης του Πιθήκου Αράχνης (Spider Monkey Optimization Algorithm).

Το 2015 παρουσιάστηκε η Βελτιστοποίηση Αφρικανικού Βούβαλου (African Buffalo Optimization) από τους Odili and Kahar (2015) ως μια μέθοδος βελτιστοποίησης για Προβλήματα Πλανόδιου Πωλητή. Αξιοπερίεργο είναι ότι την ίδια χρονιά παρουσιάστηκαν δύο διαφορετικοί αλγόριθμοι που βασίζονται στην κοινωνική συμπεριφορά των ελεφάντων (S. Deb, S. Fong and Z. Tian, 2015; Wang, Deb and Coelho, 2015a) καθώς επίσης ότι και στις δύο συμμετέχει ο Καθηγητής S. Deb. Επίσης το 2015, οι R. Omidvar, H. Parvin and F. Rad (2015) πρότειναν τη Βελτιστοποίηση Κλωσόπουλων Πέρδικας See-See (See-See Partridge Chicks Optimization). Την ίδια χρονιά, οι Wang, Deb and Coelho (2015b) προτείνουν τον Αλγόριθμο Βελτιστοποίησης Γαιοσκώληκα (Earthworm Optimization Algorithm),

τον οποίο ανέπτυξαν ως μέθοδο επίλυσης του προβλήματος Σύνθεσης Θεμάτων Εξέτασης (Test-Sheet Composition ή TSC), όπου ο στόχος είναι να βρεθεί ένας συνδυασμός ερωτήσεων από μια τράπεζα θεμάτων (item bank) που θα συνθέσουν τα θέματα εξέτασης.

Ο Αλγόριθμος Βελτιστοποίησης Λέοντα (Lion Optimization Algorithm) προτάθηκε από τους Yazdani and Jolai (2016), ενώ την ίδια χρονιά ο τρόπος επικοινωνίας και η κοινωνική συμπεριφορά των δελφινιών ενέπνευσε τους Wu, Yao and Yang (2016) να προτείνουν τον Αλγόριθμο Σμήνους Δελφινιών (Dolphin Swarm Algorithm). Τα κογιότ (coyotes) αποτέλεσαν πηγή έμπνευσης για τους J. Pierezan and L. Dos Santos Coelho (2018) οι οποίοι πρότειναν τον ομώνυμο αλγόριθμο.

Τέλος, ο Αυτοκρατορικός Πιγκουίνος ενέπνευσε τους Dhiman and Kumar (2018) να προτείνουν τον Βελτιστοποιητή Αυτοκρατορικού Πιγκουίνου (Emperor Penguin Optimizer), αλλά και τους Harifi et al. (2019) που παρουσίασαν την Αποικία Αυτοκρατορικών Πιγκουίνων (Emperor Penguins Colony). Επιπλέον, ο Μπλε Πίθηκος αποτέλεσε πηγή έμπνευσης για τον ομώνυμο αλγόριθμο (Mahmood and Al-Khateeb, 2019). Ο πιο πρόσφατος αλγόριθμος σε αυτή την υποκατηγορία ονομάζεται Αλγόριθμος Κυριαρχίας (Dominion Algorithm) και είναι εμπνευσμένος από την επικράτηση διαφορετικών αγελών λύκων σε μια έκταση.

2.3.3 Άλλες συμπεριφορές σμήνους

Σε αυτή την υποκατηγορία ανήκει και η Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization - PSO), η οποία είναι βασισμένη στον τρόπο που διατηρούν το σχηματισμό τους μεγάλα σμήνη πουλιών όταν πετάνε. Βάσει του φυσικού αναλόγου της, και η Αναζήτηση Στοχαστικής Διάχυσης (Stochastic Diffusion Search) συγκαταλέγεται στην παρούσα υποκατηγορία. Γενικότερα, ως «Άλλες συμπεριφορές σμήνους» (Other Swarm Behaviors) χαρακτηρίζονται εκείνες οι αλγοριθμικές μέθοδοι που μοντελοποιούν ταυτόχρονη ή συγχρονισμένη κίνηση μεγάλων ομάδων ζώων.

Παρόλο που εμφανίζει πολύ λίγες εφαρμογές, ο αλγόριθμος του Μοντέλου Κληρονομικότητας Κοπαδιού Προβάτων (Sheep Flocks Heredity Model) πρωτοπαρουσιάστηκε περίπου πριν είκοσι χρόνια (Hyunchul Kim and Byungchul Ahn, 2001). Το 2007 παρουσιάστηκε ένας ακόμη αλγόριθμος που είναι εμπνευσμένος από τη συμπεριφορά των βακτηρίων (Tang, Wu and Saunders, 2007), ονομαζόμενος Αλγόριθμος Συρροής Βακτηρίων (Bacterial Swarming Algorithm).

Οι Garcia and Pérez (2008) πρότειναν τη Βελτιστοποίηση Αλμάτων Βατράχων (Jumping Frogs Optimization), την οποία και εφάρμοσαν σε ένα πρόβλημα p -Διάμεσου (p -Median). Έναν χρόνο αργότερα, ο Chen (2009b) προτείνει τον Αλγόριθμο Τεχνητής Αναζήτησης Σμήνους (Artificial Searching Swarm Algorithm). Οι Filho et al. (2009) πρότειναν την Αναζήτηση Κοπαδιού Ψαριών (Fish School Search) εμπνεόμενοι από τη συλλογική συμπεριφορά των ψαριών, όπου το κοπάδι

κινείται σαν να μια μονάδα στην προσπάθεια εύρεσης τροφής, αλλά και στο ενδεχόμενο αποφυγής κινδύνων.

Το Ιεραρχικό Μοντέλο Σμήνους (Hierarchical Swarm Model) παρουσιάστηκε το 2010 από τους Chen et al. (2010). Βασιζόμενοι στη μεταναστευτική συμπεριφορά των πτηνών, οι Duman, Uysal and Alkaya (2011) πρότειναν τη Βελτιστοποίηση Μετανάστευσης Πτηνών (Migrating Birds Optimization). Οι Marinakis and Marinaki (2011) παρουσίασαν τη Βελτιστοποίηση Ζευγαρώματος Βομβίνων (Bumble Bees Mating Optimization) ως μια μέθοδο επίλυσης Προβλημάτων Δρομολόγησης Οχημάτων (Vehicle Routing Problems ή VRP), που ενώ επηρεάστηκαν από τη Βελτιστοποίηση Ζευγαρώματος Μελισσών (Honey Bees Mating Optimization), οι διαφορές μεταξύ των δύο μεθόδων είναι αρκετές.

Προσπαθώντας να αναπτύξουν μια απλούστερη μορφή της Βελτιστοποίησης Σμήνους Σωματιδίων, οι Ting et al. (2012) πρότειναν τον Αλγόριθμο Αβαρούς Σμήνους (Weightless Swarm Algorithm). Τα περιστέρια αποτέλεσαν πηγή έμπνευσης για τους Qiao and Duan (2014) ώστε να αναπτύξουν μια μέθοδο σχεδιασμού διαδρομής εναέριων ρομπότ (air robot path planning). Η μετανάστευση των ζώων αποτέλεσε επίσης πηγή έμπνευσης και γέννησε τον ομώνυμο αλγόριθμο (Li, Zhang and Yin, 2014). Οι Wang, Deb and Cui (2015) πρότειναν τη Βελτιστοποίηση της Πεταλούδας Μονάρχη (Monarch Butterfly Optimization), ενώ την ίδια χρονιά οι Cheng et al. (2015) παρουσίασαν τη Βελτιστοποίηση Αποικίας Κατσαρίδων (Cockroach Colony Optimization) ως μια νέα εκδοχή της Βελτιστοποίησης Σμήνους Κατσαρίδων (Cockroach Swarm Optimization) εφαρμόζοντάς τη στην πλοήγηση των ρομπότ (robot navigation).

Ένας εκ των γνωστότερων ερευνητών που ασχολούνται με τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση είναι ο S. Mirjalili, ο οποίος έχει συμμετάσχει στις ερευνητικές ομάδες που ανέπτυξαν δύο μεθόδους για τις οποίες ήδη έγινε λόγος παραπάνω: ο Βελτιστοποιητής Γκρίζου Λύκου (Grey Wolf Optimizer) και ο Αλγόριθμος Βελτιστοποίησης Φάλαινας (Whale Optimization Algorithm). Ωστόσο, και σε αυτή την υποκατηγορία συναντάει κανείς μεθόδους που ανέπτυξε ο S. Mirjalili, όπως η Βελτιστοποίηση Νυχτοπεταλούδας και Φλόγας (Moth Flame Optimization) (Mirjalili, 2015a) και ο Αλγόριθμος Λιβελούλας (Dragonfly Algorithm) (Mirjalili, 2016a), καθώς και τον Αλγόριθμο Σμήνους Σάλπας (Salp Swarm Algorithm) (Mirjalili et al., 2017) και τον Αλγόριθμο Βελτιστοποίησης Ακρίδας (Grasshopper Optimization Algorithm) (Mirjalili et al., 2018) όπου συμμετείχε στην ομάδα εργασίας και των δύο αυτών μεθόδων. Η πλειοψηφία των μεθόδων που αναφέρονται σε αυτή την παράγραφο έχουν εφαρμοστεί σε πρότυπα προβλήματα μηχανικού.

Το 2017 προτάθηκε ο Βελτιστοποιητής Εγωιστικού Κοπαδιού (Selfish Herd Optimizer) από τους Fausto et al. (2017), ενώ μέσα στο 2018 προτάθηκαν τέσσερις ακόμη αλγόριθμοι: ο Αλγόριθμος Βιολογικής Μετανάστευσης (Biology Migration

Algorithm) (Zhang et al., 2018), ο Αλγόριθμος Βελτιστοποίησης Περηφάνιας Λέοντα (Lion Pride Optimization Algorithm) (Kaveh and Mahjoubi, 2018), ο Αλγόριθμος εμπνευσμένος από Σουρικάτες (Meerkat-inspired Algorithm) (Klein and dos Santos Coelho, 2018) και η Βελτιστοποίηση Κοπαδιού Γκνου (Wildebeests Herd Optimization) (Motevali et al., 2018). Την ίδια χρονιά προστέθηκε μια ακόμη μέθοδος σε εκείνες που είναι εμπνευσμένες από τις κατσαρίδες, η οποία ονομάστηκε Τεχνητές Κοινωνικές Κατσαρίδες (Artificial Social Cockroaches) και προτάθηκε κυρίως για προβλήματα αναγνώρισης εικόνων από τους Bouarara, Hamou and Abdelmalek (2018).

Οι πιο πρόσφατες τεχνικές που ανήκουν σε αυτή την υποκατηγορία είναι ο αλγόριθμος του Βίσωνα (Bison algorithm) (Kazikova et al., 2019), ο Αλγόριθμος Βελτιστοποίησης του Γλάρου (Seagull Optimization Algorithm) (Dhiman and Kumar, 2019), η Βελτιστοποίηση Τραγουδιού της Μεγάπτερης Φάλαινας (Vocalization of Humpback Whale Optimization) (Masadeh, Sharieh and Mahafzah, 2019), ο Βελτιστοποιητής Στρατιωτικού Σκύλου (Military Dog Optimizer) (Tripathi, Sharma and Bala, 2019), ο Βελτιστοποιητής Εξαρτώμενος από την Καταλληλότητα (Fitness Dependent Optimizer) (J. M. Abdullah and T. Ahmed, 2019) και ο επερχόμενος αλγόριθμος Συλλογικής Συμπεριφοράς Ζώων (Collective Animal Behavior) (Cuevas, Fausto and González, 2020).

2.4 Νοημοσύνη Εμπνευσμένη από Οργανισμούς (NEO)

Ενώ οι αλγόριθμοι ΝΣ είναι εμπνευσμένοι από συλλογικές συμπεριφορές (collective behavior), στη βιβλιογραφία εμφανίζονται αρκετοί αλγόριθμοι που μοντελοποιούν μεμονωμένες συμπεριφορές. Ωστόσο, είτε οι μεμονωμένες (individual) συμπεριφορές προέρχονται από ζώα, ψάρια, πουλιά ή παρατηρούνται σε είδη λουλουδιών και χλόης, όλοι οι αλγόριθμοι αυτής της κατηγορίας είναι εμπνευσμένοι από οργανισμούς. Γι' αυτό και η κατηγορία αποκαλείται Νοημοσύνη Εμπνευσμένη από Οργανισμούς (NEO).

Στην υποκατηγορία των μεθόδων που είναι βασισμένες στην πανίδα (Fauna-based methods) ανήκουν οι τεχνικές που είναι εμπνευσμένες από είδη ζώων (μεγαπανίδα), πουλιών (ορνιθοπανίδα), ψαριών (ιχθυοπανίδα), αλλά και από βακτήρια (υπόβενθος και επίβενθος πανίδα), μικρο-οργανισμούς και ιούς (μικροπανίδα), συνιστώντας την μεγαλύτερη εκ των τριών υποκατηγοριών της NEO. Οι μέθοδοι που είναι βασισμένες στη χλωρίδα (Flora-based methods) περιλαμβάνουν όλους τους αλγόριθμους που είναι εμπνευσμένοι από φυτικές μορφές ζωής, όπως δέντρα και μύκητες. Η τρίτη υποκατηγορία αποτελείται από μεθόδους που μοντελοποιούν λειτουργίες οργάνων, όπως η καρδιά, ή θεραπείες ασθενειών, όπως η χημειοθεραπεία.

2.4.1 Μέθοδοι βασισμένες στην πανίδα

Ο πρώτος αλγόριθμος που συναντάται σε αυτή την υποκατηγορία ονομάζεται Γάμος Μελισσών (Marriage in Honey Bees) και προτάθηκε από τον Abbass (2001), με σκοπό να λύνει προβλήματα ικανοποιησιμότητας (SAT). Έναν χρόνο αργότερα οι Muller et al. (2002) παρουσίασαν την Χημειοταξία Βακτηρίων (Bacterial Chemotaxis) έχοντας ως στόχο την επίλυση ενός προβλήματος σχεδιασμού αεροτομής (airfoil design problem). Η πανίδα, όπως σημειώνεται και παραπάνω, περικλείει και τους ιούς με αποτέλεσμα να κατηγοριοποιούνται εδώ αλγόριθμοι που είναι εμπνευσμένοι από αυτούς, όπως τα Ιογενή Συστήματα (Viral Systems) (Cortés et al., 2008), η Βελτιστοποίηση Βασισμένη στο Μοντέλο της Γρίπης των Χοίρων (Swine Influenza Models Based Optimization) (Pattnaik et al., 2013), ο Αλγόριθμος Βελτιστοποίησης Επιδημικότητας (Virulence Optimization Algorithm) (Jaderyan and Khotanlou, 2016) και η Αναζήτηση Αποικίας Ιών (Virus Colony Search) (Li et al., 2016b).

Παρόλο που λόγω ονόματος θα περίμενε κανείς να συναντήσει την Βελτιστοποίηση Σμήνους Γατών (Cat Swarm Optimization) (Chu, Tsai and Pan, 2006) στην κατηγορία των μεθόδων ΝΣ, ταξινομείται στη ΝΕΟ και αποτελεί ιδανικό παράδειγμα για το πρόβλημα που παρουσιάζεται στη βιβλιογραφία, όπου αρκετοί ερευνητές μπερδεύουν την έννοια του σμήνους (swarm) με αυτή του πληθυσμού (population) των λύσεων. Την ίδια χρονιά εμφανίστηκαν ο Αλγόριθμος Ανακατεμένου Άλματος Βατράχων (Shuffle Frog-leaping Algorithm) από τους Eusuff, Lansley and Pasha (2006) και η Βελτιστοποίηση Ζευγαρώματος Μελισσών (Honey-bees Mating Optimization) από τους Haddad, Afshar and Mariño (2006). Οι S. He, Q. H. Wu and J. R. Saunders (2006) εμπνεόμενοι από τον τρόπο αναζήτησης (τροφής ή καταλύματος) που συναντάται στα ζώα και την συνύπαρξη σε μεγάλες ομάδες αυτών, παρουσίασαν τον Βελτιστοποιητή Αναζήτησης Ομάδας (Group Search Optimizer).

Το 2007 παρουσιάστηκε η πρώτη μέθοδος που είναι εμπνευσμένη από τους πιθήκους (Mucherino and Seref, 2007). Η δεύτερη μέθοδος με την ίδια πηγή έμπνευσης προτάθηκε έναν χρόνο αργότερα (Zhao and Tang, 2008), ενώ την ίδια χρονιά εμφανίστηκε μια ακόμη μέθοδος εμπνευσμένη από τις μέλισσες, η οποία ονομάζεται Αλγόριθμος Συλλογής Γύρης από Μέλισσες (Bee Collecting Pollen Algorithm) (Lu and Zhou, 2008) και προτάθηκε σαν τρόπος επίλυσης Προβλημάτων Πλανόδιου Πωλητή, όπως και η πλειοψηφία των μελισσο-εμπνευσμένων αλγορίθμων. Επίσης το 2008 παρουσιάζεται από τον Simon (2008) ένας εκ των γνωστότερων αλγορίθμων, η Βελτιστοποίηση βασισμένη στη Βιογεωγραφία (Biogeography-based Optimization), η οποία μοντελοποιεί την μετανάστευση (migration) και την μετάλλαξη (mutation) που παρατηρείται στα ζώα βάσει των γεωγραφικών τους ιδιοτήτων, δηλαδή των συνθηκών του περιβάλλοντος που διαβιώνουν.

Οι Hernández and Blum (2011) παρουσίασαν ένα μαθηματικό μοντέλο καλέσματος των δενδροβάτραχων (tree frogs), ονομάζοντας τον αλγόριθμο Ιαπωνικοί Δενδροβάτραχοι (Japanese Tree Frogs). Δύο αλγόριθμοι με πηγή έμπνευσης διαφορετικά έντομα παρουσιάστηκαν το 2011: η Βελτιστοποίηση Φρουτόμυγας (Fruit Fly Optimization) από τον Pan (2011) και η Ωοτοκία Κουνουπιών (Mosquitos Oniposition) από τους Minhas and Arif (2011).

Οι νυχτερίδες υπήρξαν πηγή έμπνευσης μια ακόμη φορά, καθώς ο Tawfeeq (2012) πρότεινε τον Αλγόριθμο Ηχοεντοπισμού Νυχτερίδων (Bat Sonar Algorithm). Ο τσαλαπετεινός ενέπνευσε τους El-Dosuky et al. (2012) να προτείνουν την Ευρετική Βελτιστοποίηση Τσαλαπετεινού (Hoopoe Heuristic Optimization). Την ίδια χρονιά προτάθηκε ο πρώτος αλγόριθμος που ήταν εμπνευσμένος από τα λιοντάρια (Rajakumar, 2012), ενώ έκανε την εμφάνισή του και Η Διαδρομή του Μεγάλου Σολομού (The Great Salmon Run) (Mozaffari, Fathi and Behzadipour, 2012). Επιπλέον, προτάθηκε ο Αλγόριθμος του Υπερ-βακτηρίου (Superbug Algorithm) (Anandaraman, Madurai Sankar and Natarajan, 2012) με σκοπό την επίλυση προβλημάτων χρονοδρομολόγησης (scheduling).

Οι σουπιές (cuttlefish) αποτέλεσαν έμπνευση ώστε οι Eesa, Brifcani and Orman (2013) να προτείνουν τον ομώνυμο αλγόριθμο. Ο ηχοεντοπισμός (echolocation) των δελφινιών ενέπνευσε τους Kaveh and Farhoudi (2013), οι οποίοι πρότειναν έναν αλγόριθμο για την επίλυση εφαρμογών μηχανικού. Ο Αλγόριθμος Βελτιστοποίησης Αιγυπτιακού Γύπα (Egyptian Vulture Optimization Algorithm) προτάθηκε από τους Sur, Sharma and Shukla (2013) και δοκιμάστηκε με επιτυχία στο Πρόβλημα του Σακιδίου (Knapsack problem). Την ίδια χρονιά προτείνεται από την ίδια συγγραφική ομάδα (Sur and Shukla, 2013) ο Αλγόριθμος Βελτιστοποίησης Πράσινου Ερωδιού (Green Heron Optimization Algorithm) ως μέθοδος επίλυσης προβλημάτων δρομολόγησης. Ένα είδος πάπιας, το οποίο ονομάζεται Keshtel, έδωσε το όνομά του στον αλγόριθμο που εμπνεύστηκαν και παρουσίασαν οι Hajiaghaei-Keshteli and Aminnayeri (2013). Ο Αλγόριθμος της Πυγολαμπίδας Άλτη (Jumper Firefly Algorithm) προτάθηκε από τους M. Bidar and H. Rashidy Kanan (2013) για την επίλυση προβλημάτων ανάθεσης (assignment). Οι Tilahun and Ong (2013) βασίστηκαν στη διαφορετική συμπεριφορά θύματος (prey) και θύτη (predator) και ανέπτυξαν έναν σχετικό αλγόριθμο. Τα βακτήρια έγιναν διάσημη πηγή έμπνευσης για τους ερευνητές, λόγω της καλής απόδοσης των ήδη υπαρχόντων αλγορίθμων που βασίζονται σε αυτά. Το γεγονός αυτό οδήγησε τους H. Mo and L. Xu (2013) να αναπτύξουν τον Αλγόριθμο Βελτιστοποίησης Μαγνητοτακτικών Βακτηρίων (Magnetotactic Bacteria Optimization Algorithm). Μια ακόμη τεχνική που παρουσιάστηκε το 2013 ονομάζεται Βελτιστοποίηση Περιπλάνησης Μυδιών (Mussels Wandering Optimization) και προτάθηκε από τους An et al. (2013).

Το 2014 εμφανίστηκε μια ακόμη μέθοδος βασισμένη σε συμπεριφορά των πουλιών, επονομαζόμενη Βελτιστοποιητής Ζευγαρώματος Πουλιών (Bird Mating Optimizer) (Askarzadeh, 2014). Την ίδια χρονιά, οι Salcedo-Sanz et al. (2014)

πρότειναν τον Αλγόριθμο Βελτιστοποίησης Κοραλλιογενούς Ύφαλου (Coral Reefs Optimization Algorithm) ως μέθοδο επίλυσης προβλημάτων δρομολόγησης. Ο τρόπος κατανομής της διαθέσιμης τροφής στον πληθυσμό μιας ομάδας ζώων προξένησε το ενδιαφέρον των S. Mohseni et al. (2014), οι οποίοι πρότειναν τον Ανταγωνισμό στον Πόρο (Competition over Resources), ενώ η συμβίωση διαφορετικών ειδών σε μια περιοχή ενέπνευσε τους Cheng and Prayogo (2014) να αναπτύξουν την Αναζήτηση Συμβιωτικών Οργανισμών (Symbiotic Organisms Search).

Ο Mirjalili (2015b) έχει προτείνει και σε αυτή την υποκατηγορία έναν αλγόριθμο, τον Βελτιστοποιητή Μηρυλεοντίδας (Ant Lion Optimizer). Επιπλέον, την ίδια χρονιά έκαναν την εμφάνισή τους ο Αλγόριθμος Τεχνητών Φυκιών (Artificial Algae Algorithm) (Uymaz, Tezel and Yel, 2015) και ο Αλγόριθμος Ιαγουάρου (Jaguar Algorithm) (Chen et al., 2015).

Το 2016 παρουσιάστηκαν έξι αλγόριθμοι: ο Αλγόριθμος Καμήλας (Camel Algorithm) (Ibrahim and Ali, 2016), ο Αλγόριθμος Αναζήτησης Κόρακα (Crow Search Algorithm) (Askarzadeh, 2016), ο Αλγόριθμος Κόκκινου Ελαφιού (Red Deer Algorithm) (Fard and Hajiaghahi-Keshteli, 2016), η Βελτιστοποίηση Πετάγματος Κουνουπιού (Mosquito Flying Optimization) (M. Alauddin, 2016), η Βελτιστοποίηση Οσφρησης Καρχαρία (Shark Smell Optimization) (Abedinia, Amjady and Ghasemi, 2016) και ο Αλγόριθμος Φάλαινας Φυσητήρα (Sperm Whale Algorithm) (Ebrahimi and Khamehchi, 2016).

Οι πεταλούδες (Qi, Zhu and Zhang, 2017) και τα σκαθάρια (Jiang and Li, 2017) αποτέλεσαν πηγή έμπνευσης για ομώνυμους αλγορίθμους το 2017. Η Βελτιστοποίηση εντοπισμού μέσω ηλεκτρικών πεδίων των Ψαριών (Fish Electrolocation Optimization) προτάθηκε από τους Halдар and Chakraborty (2017) για την επίλυση προβλημάτων αξιοπιστίας. Ένα ακόμη είδος φάλαινας προστέθηκε στη λίστα αυτών που αποτέλεσαν έμπνευση για μεθόδους Εμπνευσμένες από τη Φύση, καθώς το 2017 προτάθηκε ο Αλγόριθμος Φάλαινας Δολοφόνου (Killer Whale Algorithm) από τους Biyanto et al. (2017). Στη λίστα του πρόσθεσε έναν ακόμη αλγόριθμο και ο Hosseini (2017), ο οποίος πρότεινε τον Αλγόριθμο Κότας που Γεννάει Αυγά (Laying Chicken Algorithm). Οι Połap and Wozniak (2017) πρότειναν τον Αλγόριθμο Βελτιστοποίησης Πολικής Αρκούδας (Polar Bear Optimization Algorithm) ως μέθοδο επίλυσης σχεδιαστικών προβλημάτων μηχανικού. Οι Samareh Moosavi and Khatibi Bardsiri (2017) παρουσίασαν τον Βελτιστοποιητή Σατέν Πτιλονορρυγχίδας (Satin Bowerbird Optimizer), όπου η έμπνευση προήλθε από τις δομές που δημιουργούν ως σπίτι τα αρσενικά.

Όπως φαίνεται και στο Γράφημα 2-1 παραπάνω, το 2018 ήταν το έτος με το μεγαλύτερο πλήθος δημοσιευμένων αλγορίθμων από την κατηγορία της ΝΕΟ. Με πηγή έμπνευσης τα θυσανόποδα (barnacles) τα οποία προσκολλώνται σε ζώα της θάλασσας όπως οι θαλάσσιες χελώνες, οι Sulaiman et al. (2018) παρουσίασαν τον

Βελτιστοποιητή Ζευγαρώματος Θυσανόποδων (Barnacles Mating Optimizer), τον οποίο εφάρμοσαν σε προβλήματα ανάθεσης (Assignment Problem). Μια ακόμη μέθοδος εμπνευσμένη από τα δελφίνια, επονομαζόμενη Βελτιστοποίηση Δεξαμενής Δελφινιών (Dolphin Pod Optimization) προτάθηκε από τους Serani and Diez (2018). Άλλοι αλγόριθμοι που προτάθηκαν την ίδια χρονιά είναι ο Αλγόριθμος Βελτιστοποίησης Βασισμένος στον Γατόπαρδο (Cheetah Based Optimization Algorithm) (Klein, Mariani and Coelho, 2018), οι Κυκλικές Δομές Τετραοδοντίδων (Circular Structures of Puffer Fish) (M. C. Catalbas and A. Gulten, 2018), ο Αλγόριθμος Ψαριού που αναπαράγεται από το Στόμα (Mouth Brooding Fish algorithm) (Jahani and Chizari, 2018) και ο Αλγόριθμος του Σκαθαριού Pity (Pity Beetle Algorithm) (Kallioras, Lagaros and Avtzis, 2018).

Το ίδιο έτος προτάθηκαν ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σκαθαριών (Beetle Swarm Optimization Algorithm) από τους Wang, Yang and Liu (2018) και ο Βελτιστοποιητής Διαπεραστικού Αετού (Eagle Piercing Optimizer) από τους Khan et al. (2018) ως τεχνικές επίλυσης προβλημάτων μηχανικού. Ο Αλγόριθμος Αναζήτησης Κουκουβάγιας (Owl Search Algorithm) (Jain et al., 2018) προτάθηκε για την επίλυση του προβλήματος Ανταλλαγής Ροής Θερμότητας (Heat Flow Exchange) και η Βελτιστοποίηση Σμήνους Pontogammarus Maeoticus (Pontogammarus Maeoticus Swarm Optimization) (Ghojogh and Sharifian, 2018) προτάθηκε για την βελτιστοποίηση διάταξης ηλιακών φωτοβολταϊκών (Solar PV Arrays). Επιπρόσθετα, ο Αλγόριθμος Αναζήτησης Ρινόκερου (Rhinoceros Search Algorithm) (Deb et al., 2018) προτάθηκε ως μέθοδος προβλημάτων χρονοπρογραμματισμού συστημάτων συνεχούς ροής (flow shop scheduling).

Ο Αλγόριθμος του Κόνδορα των Άνδεων (Andean Condor Algorithm) (Almonacid and Soto, 2019) προτάθηκε για την επίλυση προβλημάτων Κυτταρικής Δημιουργίας (Cell Formation problems). Με πηγή έμπνευσης από την ίδια οικογένεια πτηνών, οι Alsattar, Zaidan and Zaidan (2019) ανέπτυξαν την Αναζήτηση Καραφλού Αετού (Bald Eagle Search). Ο Αλγόριθμος Βελτιστοποίησης Γαιϊδάρου και Λαθρέμπορου (Donkey and Smuggler Optimization Algorithm) προτάθηκε από τους Shamsaldin et al. (2019) για την επίλυση Προβλημάτων Πλανόδιου Πωλητή. Ο Αλγόριθμος Βελτιστοποίησης Γερακιού (Falcon Optimization Algorithm) (Vasconcelos Segundo, Mariani and Santos Coelho, 2019), ο Βελτιστοποιητής Ιπτάμενου Σκίουρου (Flying Squirrel Optimizer) (Azizyan et al., 2019), η Βελτιστοποίηση Γερακιών Χάρτις (Harris Hawks Optimization) (Heidari et al., 2019) και ο Αλγόριθμος Βελτιστοποίησης Καπνογλάρονου (Sooty Tern Optimization Algorithm) (Dhiman and Kaur, 2019) εφαρμόστηκαν σε πρότυπα προβλήματα μηχανικού στην εργασία που παρουσιάστηκαν. Ο Lamy (2019) παρουσίασε το Τεχνητό Τάισμα Πτηνών (Artificial Feeding Birds), εφαρμόζοντας την μέθοδο του σε διάφορα προβλήματα, όπως το Πρόβλημα του Πλανόδιου Πωλητή, την εκπαίδευση Νευρωνικών Δικτύων κ.ά..

Πρόσφατα προτάθηκαν, επίσης, ο Αλγόριθμος Βελτιστοποίησης Ρακούν (Raccoon Optimization Algorithm) (S. Zangbari Koochi et al., 2019), ο Αλγόριθμος Βελτιστοποίησης Θαλάσσιου Λέοντα (Sea Lion Optimization Algorithm) (Masadeh, 2019) και ο Αλγόριθμος Αναζήτησης Σκίουρου (Squirrel Search Algorithm) (Jain, Singh and Rani, 2019).

2.4.2 Μέθοδοι βασισμένες στην χλωρίδα

Το 2006, οι Mehrabian and Lucas (2006) προτείνουν τη Βελτιστοποίηση Χωροκατακτητικού Ζιζανίου (Invasive Weed Optimization), που μαζί με τον Αλγόριθμο Δενδρυλλίων που Αναπτύσσονται (Saplings Growing up Algorithm) (Karci and Alatas, 2006) είναι οι πρώτες μέθοδοι που βασίστηκαν στα φυτά. Δύο χρόνια αργότερα, οι W. Cai, W. Yang and X. Chen (2008) προτείνουν τη Βελτιστοποίηση Ανάπτυξης Φυτών (Plant Growth Optimization) με σκοπό να λύνουν προβλήματα μείωσης ταχύτητας (speed reducer problem). Το 2009 έκανε την εμφάνισή του ο Αλγόριθμος Ορυζώνα (Paddy Field Algorithm), ο οποίος προτάθηκε από τους Premaratne, Samarabandu and Sidhu (2009).

Ο Αλγόριθμος Βελτιστοποίησης Τεχνητού Φυτού (Artificial Plant Optimization Algorithm) προτάθηκε από τους Z. Zhao et al. (2011) για την επίλυση προβλημάτων μηχανικού και ο Αλγόριθμος Πολλαπλασιασμού Φυτών (Plant Propagation Algorithm) χρησιμοποιήθηκε για την επίλυση του προβλήματος Καθαρισμού Χλωροβενζίνης Chlorobenzene Purification problem) στην εργασία που παρουσιάστηκε (Salhi and Fraga, 2011). Ο Αλγόριθμος Βελτιστοποίησης εμπνευσμένος από την Οικολογία (Ecology-inspired Optimization Algorithm) (R. S. Parpinelli and H. S. Lopes, 2011) έκανε επίσης την εμφάνισή του το 2011. Έχοντας εμπνευστεί από τον τρόπο διάδοσης ενός είδους μούχλας, που ονομάζεται φύσαρο (physarum), οι Song et al. (2012) πρότειναν την Βελτιστοποίηση Φύσαρου (Physarum Optimization). Ο Yang (2012a) πρότεινε το 2012 τον αρκετά γνωστό Αλγόριθμο Επικονίασης Λουλουδιών (Flower Pollination Algorithm) ως μέθοδο επίλυσης πρότυπων προβλημάτων μηχανικού.

Το 2013, οι Qi et al. (2013) προτείνουν τη Βελτιστοποίηση Μάζας Ρίζας (Root Mass Optimization) κι έναν χρόνο αργότερα οι Zhang, Zhu and Chen (2014) παρουσιάζουν τον Αλγόριθμο Ανάπτυξης Ριζών (Root Growth Algorithm). Την ίδια χρονιά, ο Αλγόριθμος Βελτιστοποίησης Δάσους (Forest Optimization Algorithm) προτείνεται ως μέθοδος συσταδοποίησης από τους Ghaemi and Feizi-Derakhshi (2014). Ο τρόπος ανάπτυξης των ριζών των δέντρων και τα δέντρα γενικότερα αποτέλεσαν πηγή έμπνευσης για τον Αλγόριθμο Παρακλαδιών Ρίζας (Runner-root Algorithm) (Merrikh-Bayat, 2015), τον Αλγόριθμο Αναπαραγωγής Φυτών που αναπτύσσονται από Σπόρους (Seed-based Plant Propagation Algorithm) (Sulaiman and Salhi, 2015), τον Αλγόριθμο Σπόρου Δέντρου (Tree Seed Algorithm) (Kiran, 2015), την Φυσική Ανάπλαση Δάσους (Natural Forest Regeneration) (Moez, Kaveh and Taghizadieh, 2016) και τον Αλγόριθμο Βελτιστοποίησης Ριζών Δέντρου (Root

Tree Optimization Algorithm) (Labbi et al., 2016). Το 2016, οι Cheng et al. (2016) έλυσαν προβλήματα χρονοδρομολόγησης (scheduling) με την προτεινόμενη μέθοδό τους, την οποία ονόμασαν Αλγόριθμο Υδροχαρούς Ζιζανίου (Waterweeds Algorithm).

Δύο ακόμη αλγόριθμοι εμπνευσμένοι από γνωρίσματα των δέντρων προτάθηκαν το 2018: η Βελτιστοποίηση Βασισμένη στα Δέντρα (Tree Based Optimization) από τους Ghojogh, Sharifian and Mohammadzade (2018) και ο Αλγόριθμος Ανάπτυξης Δέντρων (Tree Growth Algorithm) από τους Cheraghalipour, Hajiaghaei-Keshteli and Paydar (2018). Την ίδια χρονιά έκανε την εμφάνισή του ο Αλγόριθμος Αναρρόφησης Οτρικουλάριων (BladderWorts Suction Algorithm), ο οποίος προτάθηκε από τους R. Gowri and R. Rathipriya (2018). Οι M. Bidar et al. (2018) εφάρμοσαν την μέθοδο που πρότειναν την ίδια χρονιά, τη Βελτιστοποίηση Αναπαραγωγής Μανιταριών (Mushroom Reproduction Optimization), σε πρότυπα προβλήματα μηχανικού. Οι R. Gowri and R. Rathipriya (2018) πρότειναν μια μέθοδο ακόμη εμπνευσμένη από το φύσαρο (physarum), υπό το όνομα Αλγόριθμος Βελτιστοποίησης Ενέργειας Φύσαρου (Physarum-energy Optimization Algorithm), προτάθηκε από τους Feng et al. (2019) ως μέθοδος επίλυσης Προβλημάτων Πλανόδιου Πωλητή.

2.4.3 Άλλα παραδείγματα

Η μικρότερη υποκατηγορία των μεθόδων NEO αποτελείται από αλγορίθμους που είναι βασισμένοι κατά κύριο λόγο στον τρόπο λειτουργίας των οργάνων του σώματος. Ο πρώτος χρονικά αυτής της υποκατηγορίας είναι ο Αλγόριθμος Επιλογής Κλώνου (Clonal Selection Algorithm) (de Castro and von Zuben, 2000), ο οποίος προτάθηκε το 2000. Εάν θέλαμε να διευρύνουμε το εύρος των μεθόδων που εξετάζονται, τόσο ο Γενετικός Αλγόριθμος (Genetic Algorithm) (Holland, 1975, 1992a) όπως και η Διαφορική Εξέλιξη (Differential Evolution) (Storn and Price, 1997) θα μπορούσαν να συμπεριληφθούν σε αυτή την υποκατηγορία. Ωστόσο, και οι δύο προαναφερόμενες μέθοδοι θεωρούνται Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms), που στη βιβλιογραφία οι Εξελικτικοί Αλγόριθμοι αντιμετωπίζονται ως ο πρόδρομος των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση και πολλές φορές θεωρούνται ως κατηγορία-γονέας των δεύτερων.

Δύο αλγόριθμοι, που η ένταξή τους στους Εμπνευσμένους από τη Φύση εγείρει αρκετή συζήτηση, είναι εμπνευσμένοι από τη μετενσάρκωση (reincarnation). Αρχικά, ο Αλγόριθμος της Μετενσάρκωσης (Reincarnation Algorithm) προτάθηκε από τον A. Sharma (2010) για την επίλυση Προβλημάτων Πλανόδιου Πωλητή, ενώ λίγο αργότερα οι Nguyen and Bhanu (2012) παρουσίασαν τη Βελτιστοποίηση Επιβίωσης Ζόμπι (Zombie Survival Optimization), την οποία χρησιμοποίησαν για παρακολούθηση εικόνων (image tracking).

Ο Αλγόριθμος Βελτιστοποίησης Βλαστοκύτταρων (Stem Cells Optimization Algorithm) ήταν ο πρώτος αλγόριθμος που αντλούσε την έμπνευσή του από το

ανθρώπινο σώμα (Taherdangko, Yazdi and Bagheri, 2012). Ωστόσο, ο πρώτος αλγόριθμος που είναι εμπνευσμένος από κάποια ζωτική λειτουργία του σώματος είναι ο αλγόριθμος της Καρδιάς (Heart algorithm), ο οποίος προτάθηκε για ανάλυση συστάδων (cluster analysis) το 2014 από τον Hatamlou (2014). Ένα χρόνο αργότερα παρουσιάστηκε η Βελτιστοποίηση Επιθετικής Ανάπτυξης Όγκου (Invasive Tumor Growth Optimization) από τους Tang et al. (2015), ενώ ακόμη πιο μετά, οι Jaddi, Alvankarian and Abdullah (2017) παρουσίασαν τον Αλγόριθμο εμπνευσμένο από το Συκώτι (Kidney-inspired Algorithm) και τον εφάρμοσαν σε Προβλήματα Πλανόδιου Πωλητή. Μια μέθοδος που μοιάζει σε μεγάλο βαθμό με τα Τεχνητά Νευρωνικά Δίκτυα, η Επικοινωνία Νευρώνων (Neuronal Communication) έκανε πρόσφατα την εμφάνισή της ως μέθοδος επίλυσης πρότυπων προβλημάτων μηχανικού (Asil Gharebaghi and Ardalan Asl, 2017).

Βασισμένοι στο φαινόμενο της αλλοστάσεως (allostasis) όπου το ανθρώπινο σώμα προσπαθεί να σταθεροποιήσει την κατάσταση που δεν λειτουργεί σωστά ένα όργανο, οι Osuna-Enciso et al. (2016) παρουσίασαν την Αλλοστατική Βελτιστοποίηση (Allostatic Optimization). Μια τέτοια κατάσταση μπορεί να είναι και μια αρρώστια, μια έννοια στην οποία βασίστηκε η Βελτιστοποίηση Τεχνητής Μεταδοτικής Ασθένειας (Artificial Infectious Disease Optimization) (Huang, 2016). Πιο πρόσφατη μέθοδος αυτής της κατηγορίας πηγής έμπνευσης ονομάζεται Αλγόριθμος Επιστήμης Χημειοθεραπείας (Chemotherapy Science Algorithm) και προτάθηκε από τους Salmani and Eshghi (2017) για την επίλυση Προβλημάτων Πλανόδιου Πωλητή.

2.5 Νοημοσύνη Εμπνευσμένη από Φυσικά Φαινόμενα και Νόμους των Επιστημών (ΝΕΦΦΝΕ)

Η κατηγορία των Αλγορίθμων που είναι Εμπνευσμένοι από Φυσική και Χημεία (Physics and Chemistry) επαναπροσδιορίστηκε στα πλαίσια της διατριβής (Tzanetos and Dounias, 2017b) ως μέθοδοι Εμπνευσμένες από Φυσικά Φαινόμενα και Νόμους των Επιστημών (Physical Phenomena & Laws of Science). Η συγκεκριμένη κατηγορία περιέχει τους αλγόριθμους που αντλούν την έμπνευσή τους από φυσικά φαινόμενα που τα διέπουν νόμοι της Φυσικής, της Μηχανικής κλπ ή από καθαυτούς τους νόμους, όπως διακρίνεται και από τα ονόματά τους που περιέχονται στον Πίνακας I-3 στο Παράρτημα I.

Οι δημοσιεύσεις των Xing and Gao (2014), Biswas et al. (2013), Can and Alatas (2015) και Siddique and Adeli (2016) έχουν χαρτογραφήσει εν μέρει την συγκεκριμένη κατηγορία αλγορίθμων, χωρίς ωστόσο σε καμία εξ αυτών να περιέχεται το σύνολο των μεθόδων. Στις υποενοτήτες που ακολουθούν γίνεται μια διεξοδική ανάλυση των μεθόδων Νοημοσύνης Εμπνευσμένης από Φυσικά Φαινόμενα και Νόμους των Επιστημών (ΝΕΦΦΝΕ).

2.5.1 Αλγόριθμοι Εμπνευσμένοι από Φαινόμενα του Σύμπαντος

Οι Hsiao et al. (2005) εμπνεύστηκαν από την κίνηση των αστεροειδών στο διάστημα και τη βαρυτική έλξη μεταξύ αυτών και παρουσίασαν τον Αλγόριθμο Βαρύτητας του Διαστήματος (Space Gravitational Algorithm), ο οποίος και ήταν ο πρώτος χρονικά αυτής της υποκατηγορίας. Υπογράμμισαν, μάλιστα, ότι η μέθοδός τους βασίστηκε στον Γενικευμένο Νόμο της Σχετικότητας του Einstein. Στην ίδια εργασία, δοκίμασαν την μέθοδό τους πάνω στο πρόβλημα βέλτιστου σχεδιασμού ενός ελεγκτή αναλογικού-ολοκληρωτικού και διαφορικού ελέγχου (PID controller), ξεπερνώντας σε απόδοση κλασικές μεθόδους όπως η Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization), κάνοντας λιγότερες επαναλήψεις και χρησιμοποιώντας μικρότερο πληθυσμό.

Την επόμενη χρονιά, οι Erol and Eksin (2006) παρουσίασαν τον αλγόριθμο Μεγάλης Έκρηξης – Μεγάλης Σύνθλιψης (Big Bang – Big Crunch ή BB-BC), ο οποίος μοντελοποιούσε το γνωστό φαινόμενο της δημιουργίας του σύμπαντος. Στην ουσία έχει δύο φάσεις: (α) αυτή της Μεγάλης Έκρηξης (Big Bang), όπου οι λύσεις παίρνουν τυχαίες θέσεις στον χώρο και (β) αυτή της Μεγάλης Σύνθλιψης (Big Crunch), όπου σταδιακά έρχονται όλο και πιο κοντά, προσπαθώντας να συγκεντρωθούν γύρω από ένα βέλτιστο σημείο. Σημείωσαν, δε, ότι στόχος τους ήταν να προτείνουν μια τέτοια μέθοδο που να ξεπερνά τα προβλήματα πρόωρης σύγκλισης, ταχύτητας σύγκλισης και χρόνου εκτέλεσης του κλασικού Γενετικού Αλγόριθμου (ΓΑ). Κάτι που τελικά και κατάφεραν, όπως φάνηκε από τις συγκρίσεις στα αποτελέσματα της μεθόδου τους με τον ΓΑ πάνω σε εξισώσεις ελέγχου αποδοτικότητας.

Ύστερα από τρία χρόνια, οι Rashedi, Nezamabadi-Pour and Saryazdi (2009), δουλεύοντας πάνω στο ίδιο μοτίβο (βαρυτική έλξη μαζών στο διάστημα) και έχοντας επιρροή από την Βελτιστοποίηση Κεντρομόλου Δυνάμεως (Central Force Optimization - αναφέρεται παρακάτω), παρουσίασαν τον Αλγόριθμο Βαρυτικής Αναζήτησης (Gravitational Search Algorithm ή GSA). Τα αποτελέσματά τους τα συνέκριναν με αυτά της Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization), του Πραγματικού Γενετικού Αλγόριθμου (Real Genetic Algorithm) και της Βελτιστοποίησης Κεντρομόλου Δυνάμεως (Central Force Optimization), της οποίας ήθελαν να είναι μια καλύτερη εκδοχή η δική τους μέθοδος. Όντας ένας από τους καλύτερους αλγόριθμους, ο Αλγόριθμος Βαρυτικής Αναζήτησης (ABA) ενέπνευσε τους Flores, López and Barrera (2011) και με τη σειρά τους παρουσίασαν την Βελτιστοποίηση Βαρυτικών Διαδράσεων (Gravitational Interactions Optimization ή GIO) το 2011. Παρόλο που σαν γενική ιδέα μοιάζει πολύ με τον ABA και την Αναζήτηση Φορτισμένου Συστήματος (Charged System Search - που αναφέρεται επίσης παρακάτω), έχει τις εξής διαφορές:

1. η σταθερά G δεν είναι μειούμενη σε βάθος επαναλήψεων, όπως στον ABA

2. οι λύσεις αλληλεπιδρούν μεταξύ τους, ενώ στον GSA αλληλεπιδρούν μόνο οι k κοντινές
3. δεν υπάρχουν φορτία, όπως στον CSS και
4. ο GIO καθορίζει όλα τα τοπικά και ολικά βέλτιστα για συναρτήσεις με πολλά βέλτιστα (multimodal functions).

Τα χαρακτηριστικά αυτά φαίνονται να βοηθούν τον εν λόγω αλγόριθμο να ξεπερνάει την κλασική Βελτιστοποίηση Σμήνους Σωματιδίων (PSO) και μια παραλλαγή του (NPSO), ύστερα από σύγκριση αποτελεσμάτων σε σχετικές συναρτήσεις.

Την ίδια χρονιά, ο Shah-Hosseini (2011) πρότεινε τον Αλγόριθμο Αναζήτησης βασισμένο στις κινήσεις που παρατηρούνται στον Γαλαξία (Galaxy-based Search Algorithm), στον οποίο οι λύσεις θεωρούνται πλανήτες και προσεγγίζουν ένα βέλτιστο μέσω χαοτικής σπειροειδούς κίνησης. Ο σκοπός της δημιουργίας αυτής της μεθόδου, σύμφωνα με τον συγγραφέα, ήταν η επίλυση του προβλήματος ανάλυσης κύριων συνιστωσών (Principal Components Analysis).

Δύο χρόνια μετά, ο Hatamlou (2013) προτείνει τον αλγόριθμο «Μαύρη Τρύπα» (Black Hole), όπου εδώ η καλύτερη λύση θεωρείται μια μαύρη τρύπα που ελκύει τις υπόλοιπες λύσεις προς το μέρος της. Στα πλεονεκτήματα του αλγόριθμου αυτού έναντι άλλων είναι το γεγονός ότι είναι απλός και φτιάχνεται εύκολα, αλλά και ότι δεν περιέχει παραμέτρους που να χρειάζονται ρύθμιση πριν τρέξει. Μάλιστα, έχει φανεί στη βιβλιογραφία, αλλά και ο ίδιος ο συγγραφέας είχε επισημάνει ότι ο αλγόριθμος αυτός είναι καλό εργαλείο για προβλήματα κατηγοριοποίησης (Clustering). Αυτή τη θέση ενισχύουν και τα αποτελέσματα του αλγόριθμου Μαύρης Τρύπας πάνω σε Set δεδομένων (datasets) από το Machine Learning Repository του UCI, όπου αποδείχθηκε στατιστικά καλύτερος από άλλες μεθόδους, όπως k -means, Αλγόριθμος Βαρυτικής Αναζήτησης (GSA), Βελτιστοποίηση Σμήνους Σωματιδίων (PSO), αλγόριθμος Μεγάλης Έκρηξης – Μεγάλης Σύνθλιψης (BB-BC).

Ο τελευταίος από αυτή την υποκατηγορία βασίζεται στις έννοιες της κοσμολογίας: άσπρη τρύπα, μαύρη τρύπα και «σκουληκότρυπα» (καλύτερα white hole, black hole, wormhole) και παρουσιάστηκε το 2015 από τους Mirjalili, Mirjalili and Hatamlou (2016), παίρνοντας το όνομα Βελτιστοποιητής Πολύ-σύμπαντος (Multi-verse Optimizer ή MVO). Και γι' αυτόν τον αλγόριθμο, σημαντικό ρόλο έπαιξαν οι πολύ καλές επιδόσεις του Αλγόριθμου Βαρυτικής Αναζήτησης, οι οποίες προέτρεψαν τους συγγραφείς να στραφούν στην επιστήμη για έμπνευση. Τα αποτελέσματα από 19 προβλήματα (test problems για την ακρίβεια) έδειξαν ότι ο Βελτιστοποιητής Πολύ-σύμπαντος είναι καλή μέθοδος για επίλυση προβλημάτων διακριτού χώρου. Ο λόγος αποδίδεται στην καλή ισορροπία που έχουν πετύχει οι συγγραφείς στο exploration (αναζήτηση σε όλο τον χώρο λύσεων) και στο exploitation (τοπική αναζήτηση) της μεθόδου αυτής.

2.5.2 Αλγόριθμοι Εμπνευσμένοι από Νόμους των (Θετικών) Επιστημών

Αρχικά, το 2002, οι Zarand et al. (2002) πρότειναν την Βελτιστοποίηση βασισμένη στην Υστέρηση (Hysteretic Optimization ή HO) βασιζόμενοι στο φαινόμενο της Υστέρησης. Οι συγγραφείς σημειώνουν ότι βασικό ρόλο για την επιλογή τους ήταν η επιτυχημένη πορεία της Προσομοιωμένης Ανόπτωσης (Simulated Annealing - SA) ως τότε, η οποία ήταν ένας βασικός λόγος που πολλοί ερευνητές στράφηκαν στη Φυσική και τις άλλες Θετικές Επιστήμες για έμπνευση. Η Βελτιστοποίηση βασισμένη στην Υστέρηση συγκρινόμενη με την Προσομοιωμένη Ανόπτωση σε προβλήματα με μαγνητικά μοντέλα (frustrated magnetic models) και στο Πρόβλημα του Περιοδεύοντος Πωλητή (Traveling Salesman Problem – TSP), σε πολλές περιπτώσεις ήταν εξίσου αποδοτικός. Οι συγγραφείς υπογράμμισαν το γεγονός ότι ο αλγόριθμος που πρότειναν μπορεί να συνδυαστεί με κάποιον άλλο και να δημιουργήσει ένα καλό υβριδικό σχήμα, αλλά και το γεγονός ότι είναι αποδοτικός σε προβλήματα με περιορισμένο χώρο λύσεων (προβλήματα κόμβων και ακμών).

Ένα χρόνο αργότερα, οι Birbil and Fang (2003) προτείνουν την Βελτιστοποίηση Ηλεκτρομαγνητισμού (Electromagnetism-like Optimization ή E/M Opt.), έναν εκ των πιο αποδοτικών και πολυχρησιμοποιημένων αλγορίθμων, όπως μπορεί να δει κανείς στον Πίνακα II-3 του Παραρτήματος II. Μια καινοτομία του αλγόριθμου αυτού ήταν η ύπαρξη ενός διανύσματος σε κάθε λύση, που άλλαζε κατεύθυνση σύμφωνα με την καλύτερη λύση κάθε φορά και έτσι μετακινούσε τις λύσεις στον χώρο με έναν έξυπνο τρόπο. Οι συγγραφείς τόνισαν το γεγονός ότι η μέθοδός τους μπορεί να λειτουργήσει και ως απλός αλγόριθμος, αλλά και ως μέρος ενός υβριδικού σχήματος, κάτι που ενισχύεται από την ύπαρξη πολλών υβριδικών παραλλαγών της Βελτιστοποίησης Ηλεκτρομαγνητισμού στη βιβλιογραφία. Ωστόσο, ο συγκεκριμένος αλγόριθμος δείχνει να λειτουργεί καλύτερα όταν υπάρχει κάποιο μοτίβο στο πρόβλημα που επιλύει. Διαφορετικά, η απόδοσή του μειώνεται.

Δύο χρόνια αργότερα, το 2005, οι Sacco and Oliveira (2005) προτείνουν τον Αλγόριθμο Σύγκρουσης Σωματιδίων (Particle Collision Algorithm ή PCA) βασιζόμενοι στη σύγκρουση σωματιδίων. Παρόλο που παρατηρεί κανείς αρκετές ομοιότητες με την Προσομοιωμένη Ανόπτωση (Simulated Annealing), μια μεγάλη διαφορά είναι ότι ο Αλγόριθμος Σύγκρουσης Σωματιδίων δεν χρειάζεται παραμετροποίηση. Ο αλγόριθμος ήταν στατιστικά καλύτερος από τον Γενετικό Αλγόριθμο και κάποιες εκδοχές αυτού, αφού εφαρμόστηκε σε γνωστές συναρτήσεις ελέγχου αποδοτικότητας (test functions) και στο πρόβλημα σχεδιασμού πυρηνικού πυρήνα (nuclear core design problem).

Το 2007, ο Formato (2007) προτείνει τον επόμενο αλγόριθμο που αποδείχθηκε εξαιρετικά χρήσιμος, σύμφωνα και με τον Πίνακα II-3 του Παραρτήματος I. Ο αλγόριθμος αυτός ονομάστηκε Βελτιστοποίηση Κεντρομόλου Δυνάμεως (Central Force Optimization ή CFO). Η ιδέα πίσω από αυτή τη μέθοδο ήταν να ελκύονται οι λύσεις μεταξύ τους, ώστε να αλληλεπιδρούν πιο δυναμικά. Τα αποτελέσματα της

Βελτιστοποίησης Κεντρομόλου Δυνάμεως σε όλες τις γνωστές συναρτήσεις ελέγχου αποδοτικότητας (test functions) έδειξαν ότι είναι ένας πολύ χρήσιμος ντετερμινιστικός αλγόριθμος (deterministic search algorithm) για πολυδιάστατα προβλήματα βελτιστοποίησης (multidimensional optimization problems).

Το 2009, οι Xie, Zeng and Cui (2009) παρουσιάζουν την Βελτιστοποίηση Τεχνητής Φυσικής (Artificial Physics Optimization ή APO) βασιζόμενοι στην ιδέα της Μιμητικής της Φυσικής (Physicomimetics), η οποία είχε αρχίσει να ανθίζει μετά από τους ήδη υπάρχοντες αλγόριθμους. Ο συγκεκριμένος αλγόριθμος, ωστόσο, όταν προτάθηκε παρουσίαζε αρκετά μειονεκτήματα, τα οποία φάνηκαν στη σύγκριση με την Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization) και εκδοχές αυτής. Τα μειονεκτήματα αυτά διορθώθηκαν σε μελλοντικές εργασίες.

Το 2010, οι Kaveh and Talatahari (2010) προτείνουν την Αναζήτηση Φορτισμένου Συστήματος (Charged System Search ή CSS), συνεχίζοντας την παράδοση με τους αλγόριθμους στους οποίους η καλύτερη λύση ελκύει τις υπόλοιπες κοντά της (attraction-based). Οι συγγραφείς ανέφεραν το γεγονός ότι η Αναζήτηση Φορτισμένου Συστήματος κάνει για όλων των ειδών τα προβλήματα βελτιστοποίησης, ειδικά για αυτά που έχουν μη «λείο» χώρο λύσεων (non-smooth) και μη κυρτό (non-convex). Η αποδοτικότητά του ελέγχθηκε επιτυχώς τόσο σε πρότυπα προβλήματα μηχανικού, όσο και σε συναρτήσεις ελέγχου αποδοτικότητας (test functions). Την ίδια χρονιά, οι Lam and Li (2010) πρότειναν την Βελτιστοποίηση εμπνευσμένη από τις Χημικές Αντιδράσεις (Chemical-reaction-based Optimization ή CRO), στην οποία εισήγαγαν τη λογική τεσσάρων σταδίων των χημικών αντιδράσεων:

1. αναποτελεσματική σύγκρουση στοιχείων
2. αποσύνθεση
3. ενδομοριακή αναποτελεσματική σύγκρουση και
4. σύνθεση

Ενώ η μέθοδός τους είναι Εμπνευσμένη από τη Φύση (Nature Inspired), οι συγγραφείς δε συνέκριναν τα αποτελέσματά τους με καμία άλλη τέτοια μέθοδο.

Στηριζόμενοι στον νόμο του Snell για την ανάκλαση του φωτός, όταν αυτό ταξιδεύει από ένα φωτεινότερο σε ένα πιο σκοτεινό μέσο, οι Kaveh and Khayatad (2012) πρότειναν την Βελτιστοποίηση Ακτίνων (Ray Optimization ή RO) το 2012. Ο αλγόριθμος εφαρμόστηκε σε πρότυπα προβλήματα μηχανικού και σε μαθηματική βελτιστοποίηση, αλλά επίσης δεν συγκρίθηκε με κάποια Εμπνευσμένη από τη Φύση (Nature Inspired) μέθοδο. Την ίδια χρονιά, οι Sadollah et al. (2012) μοντελοποιούν την τυχαιότητα της κατεύθυνσης και της ταχύτητας που έχουν τα κομμάτια μιας νάρκης αφού εκραγεί και ονομάζουν τον αλγόριθμό τους Βελτιστοποίηση βασισμένη στο φαινόμενο Έκρηξης Νάρκης (Mine Blast Optimization ή MBO). Δοκιμάζοντας την απόδοσή του σε πρότυπα προβλήματα μηχανικού, αποδεικνύεται καλύτερος από μια εκδοχή του Γενετικού Αλγόριθμου, τη Βελτιστοποίηση Σμήνους Σωματιδίων

(Particle Swarm Optimization) και εκδοχές αυτού, αλλά και της Βελτιστοποίησης Αποικίας Μυρμηγκιών (Ant Colony Optimization). Ο τρίτος αλγόριθμος που παρουσιάστηκε την ίδια χρονιά, στηρίχθηκε πάνω στη σπειροειδή κίνηση που κάνουν τα σωματίδια σε φαινόμενα όπως ο τυφώνας. Γι' αυτό και οι Nasir et al. (2013), που παρουσίασαν αυτή τη μέθοδο, την ονόμασαν Αλγόριθμο Δυναμικής της Σπειροειδούς Κινήσεως (Spiral Dynamics Algorithm).

Την επόμενη χρονιά, οι Abdechiri, Meybodi and Bahrami (2013) εμπνέονται από την κίνηση των αερίων και παρουσιάζουν την Βελτιστοποίηση Κίνησης των Αερίων κατά τον Robert Brown (Van der Pas, 1971) (Gases Brownian Motion Optimization ή GBMO), όπου οι λύσεις θεωρούνται μόρια του αέρα και ακολουθούν τυχαίες κινήσεις. Ο αλγόριθμος συγκρίνεται στο πρόβλημα της ικανοποιήσιμότητας (satisfiability problem ή αλλιώς SAT) με την Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization) και τον Αλγόριθμο Οργάνωσης Πρωταθλήματος (League Championship Algorithm), αλλά και σε συναρτήσεις ελέγχου αποδοτικότητας (test functions) με διάφορες εκδοχές του Γενετικού Αλγόριθμου. Τα αποτελέσματα έδειξαν ότι η προτεινόμενη μέθοδος είναι καλό εργαλείο για προβλήματα πολλών διαστάσεων (high dimensional problems), κυρίως λόγω του πολύ καλού μηχανισμού exploration. Ειδικότερα, όσο αυξάνεται ο πληθυσμός των λύσεων, τόσο καλύτερη απόδοση έχει ο αλγόριθμος, αν και αυξάνεται ο χρόνος υπολογισμού, παραμένοντας όμως μικρότερος από άλλες τεχνικές.

Το 2014, οι Kaveh and Mahdavi (2014) αντλούν την ιδέα της Βελτιστοποίησης Σύγκρουσης Σωμάτων (Collision Bodies Optimization) από τους νόμους της Φυσικής που ισχύουν όταν δύο σώματα συγκρούονται. Στόχος τους ήταν να προτείνουν έναν απλό σχετικά αλγόριθμο, όπως οι ίδιοι αναφέρουν στην εργασία τους. Συγκρίνοντας την απόδοση του αλγόριθμου αυτού με τους δύο άλλους που είχε προτείνει ο ένας εκ των συγγραφέων, ο Kaveh (Βελτιστοποίηση Ακτίνων - RO και Αναζήτηση Φορτισμένου Συστήματος - CSS), φαίνεται ότι παρόλο που έχει μεγαλύτερο χρόνο εκτέλεσης, είναι αποδοτικότερος από τους άλλους.

Τέλος, το 2015 ο Husseinzadeh Kashan (2015) μελέτησε τους νόμους της Οπτικής και πρότεινε τον σχετικό αλγόριθμο (Optics Inspired Optimization). Στα πλεονεκτήματά του έναντι άλλων είναι το γεγονός ότι έχει λίγες παραμέτρους μόνο ενώ παραμένει ανταγωνιστικός. Η απόδοσή του μετρήθηκε σε σύγκριση με τη Βελτιστοποίηση Σμήνους Σωματιδίων σε πρότυπα προβλήματα μηχανικού και συναρτήσεις ελέγχου αποδοτικότητας (benchmark functions), όντας καλύτερη.

2.5.3 Αλγόριθμοι Εμπνευσμένοι από Φυσικά Φαινόμενα

Σε αυτή την κατηγορία ανήκει και ο Simulated Annealing ή Προσομοιωμένη Ανόπτηση (Kirkpatrick, Gelatt and Vecchi, 1983), ο οποίος όπως φάνηκε παραπάνω αποτέλεσε πηγή έμπνευσης για τους ερευνητές, ώστε να στραφούν στην Φυσική και τις άλλες Θετικές Επιστήμες για να βρουν μοντέλα που δουλεύουν καλά στη φύση και να προτείνουν τους αντίστοιχους αλγόριθμους. Η Προσομοιωμένη Ανόπτηση

μοντελοποιεί την ιδέα της Ανόπτησης από τη Μεταλλουργία, όπου τα μέταλλα αποκτούν πιο λείες επιφάνειες μέσω ελεγχόμενης ψύξης τη στιγμή που έχουν θερμομανθεί σε πολύ υψηλές θερμοκρασίες.

Παρόλο που η Προσομοιωμένη Ανόπτηση (SA) είναι τόσο επιτυχημένος αλγόριθμος με χιλιάδες εφαρμογές σε διάφορα πεδία προβλημάτων, χρειάστηκε να περάσουν χρόνια για να εμφανιστεί άλλος αλγόριθμος από αυτή την υποκατηγορία.

Το 2007, οι Rabanal, Rodríguez and Rubio (2007) πρότειναν έναν αλγόριθμο που ονόμασαν Δυναμική Σχηματισμού Ποταμών (River Formation Dynamics ή RFD). Είχαν ως κίνητρο τη δημιουργία μιας μεθόδου που δε θα είχε τα προβλήματα της Βελτιστοποίησης Αποικίας Μυρμηγκιών:

1. τα μονοπάτια φερομόνης πολλές φορές συγχέονταν
2. δεν δημιουργούνταν εύκολα νέα μονοπάτια και
3. η φερομόνη θα έπρεπε να αυξάνεται όταν ένα μονοπάτι το ακολουθούν πολλά μυρμηγκια-λύσεις.

Το 2009, ο Shah-Hosseini (2009) προτείνει την Νοήμονα Σταγόνα Νερού (Intelligent Water Drop ή IWD) που μοντελοποιεί τον τρόπο που οι σταγόνες βρίσκουν το δρόμο από τις κορυφές των βουνών προς λίμνες λόγω της κινητικής τους ενέργειας. Ο στόχος του συγγραφέα ήταν να φτιάξει μια μέθοδο που να επιλύει προβλήματα Επιχειρησιακής Έρευνας (όπως το Πρόβλημα του Περιοδευόντος Πωλητή) και η επιτυχία του φαίνεται στα κατά πολύ καλύτερα αποτελέσματα της Νοήμονος Σταγόνας Νερού σε σχέση με μεθόδους όπως η Βελτιστοποίηση Αποικίας Μυρμηγκιών (ACO) και η Βελτιστοποίηση Αποικίας Μελισσών (Bee Colony Optimization).

Το 2012 παρουσιάζεται ο Αλγόριθμος Κύκλου του Νερού (Water Cycle Algorithm) από τους Eskandar et al. (2012). Στην ουσία ο αλγόριθμος μοντελοποιεί τον κύκλο που ακολουθεί το νερό από τα σύννεφα πέφτοντας με τη βροχή σε λόφους, κατεβαίνοντας τους ως τους πρόποδες και ύστερα εξατμιζόμενο ξανά για να δημιουργήσει νέα σύννεφα. Ο σκοπός τους ήταν να προτείνουν μια μέθοδο που να είναι ικανή να επιλύει προβλήματα διακριτού χώρου λύσεων, αλλά και πρότυπα προβλήματα μηχανικού. Συγκρινόμενη με τον Γενετικό Αλγόριθμο και τη Βελτιστοποίηση Σμήνους Σωματιδίων, σε πολλές περιπτώσεις η προτεινόμενη μέθοδος ήταν εξίσου αποτελεσματική. Ένα ακόμη μοντέλο με βάση τα σύννεφα πρότειναν την επόμενη χρονιά οι Yan and Hao (2013), ονομάζοντάς το Βελτιστοποίηση εμπνευσμένη από τα Σύννεφα της Ατμόσφαιρας (Atmosphere Clouds Model Optimization). Η καινοτομία της μεθόδου αυτής είναι η διασπορά των λύσεων στον χώρο σε αντίθεση με τις περισσότερες μεθόδους που συγκεντρώνουν τις λύσεις τους σε ένα σημείο (Reverse Search Method). Τα αποτελέσματα του αλγόριθμου συγκρίθηκαν με αυτά της Βελτιστοποίησης Σμήνους Σωματιδίων και του Γενετικού σε συναρτήσεις πολλών τοπικών μέγιστων (multimodal functions),

αποδεικνύοντας ότι ο συγκεκριμένος αλγόριθμος είναι καλό εργαλείο για τέτοια προβλήματα.

Το 2014, οι Rbough and Imrani (2014) βασιζόμενοι στη δουλειά που έγινε για την Βελτιστοποίηση Σπειροειδούς Δυναμικής (Spiral Dynamics Optimization) και προσθέτοντας στοιχεία από το φαινόμενο του τυφώνα, παρουσιάζουν την Βελτιστοποίηση βασισμένη στο φαινόμενο του Τυφώνα (Hurricane-based Optimization). Μάλιστα μέσα από έλεγχο απόδοσης πάνω σε προβλήματα με διαφορετικές διαστάσεις, έδειξαν ότι είναι καλός στην επίλυση μη γραμμικών προβλημάτων.

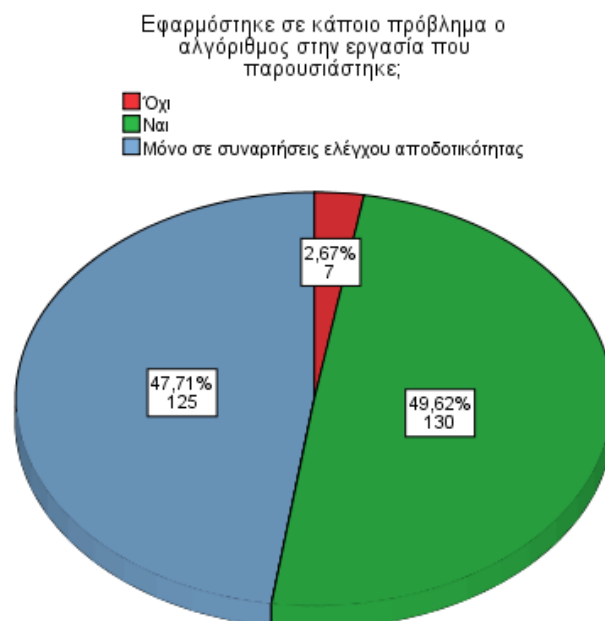
Την επόμενη χρονιά η συγκεκριμένη υποκατηγορία γνωρίζει μια ραγδαία άνοδο, όπου προτείνονται τέσσερις νέες μέθοδοι. Οι Shareef, Ibrahim and Mutlag (2015) προτείνουν μια προσέγγιση εμπνευσμένη από τη διαδικασία αναζήτησης εκτόνωσης κεραυνών (Lightning Search Algorithm) που χρησιμοποιεί τον μηχανισμό που χωρίζεται σε διάφορα μονοπάτια ένας κεραυνός και με αυτό τον τρόπο οι λύσεις ψάχνουν βέλτιστα σε διαφορετικές περιοχές. Η μέθοδος αυτή αποδείχθηκε αποδοτικότερη σε αρκετές περιπτώσεις, συγκρινόμενη με τον Αλγόριθμο Διαφορικής Αναζήτησης (Differential Search Algorithm), τον Αλγόριθμο Οπισθοδρομικής Αναζήτησης (Backtracking Search Algorithm), τον Αλγόριθμο Λαμπυρίδων (Firefly algorithm), τη Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization) και την Αναζήτηση Αρμονίας (Harmony Search) σε benchmark συναρτήσεις. Ο Zheng (2015) πρότεινε την Βελτιστοποίηση Κυμάτων Νερού (Water Wave Optimization), έχοντας στόχο να βρει έναν μηχανισμό αναζήτησης λύσεων για πολυδιάστατους χώρους λύσεων. Χρησιμοποιώντας τα φαινόμενα της διάδοσης (propagation), διάθλασης (refraction) και διάλυσης των κυμάτων (breaking), ο συγγραφέας πρότεινε μια μέθοδο που βρέθηκε αποδοτικότερη από τη Βελτιστοποίηση Χωροκατακτητικού Αγριόχορτου (Invasive Weed Optimization), τη Βελτιστοποίηση βασισμένη στη Βιογεωγραφία (Biogeography-based Optimization), του Αλγόριθμου της Νυχτερίδας (Bat algorithm) και άλλες μεθόδους, πάνω σε προβλήματα του πραγματικού κόσμου.

Μια ακόμη ιδέα προερχόμενη από τη Δυναμική του Νερού (Water Dynamics) ήταν μια προσέγγιση βασισμένη στον καταωνιστήρα ύδατος (Artificial Showering Algorithm) που πρότειναν οι Ali et al. (2015), όπου εδώ οι λύσεις διαχέονται τυχαία στον χώρο αντιγράφοντας την πτώση του νερού στο μπάνιο. Όντας απλός στην υλοποίησή του, αποδείχθηκε ιδιαίτερα ανταγωνιστικός σε σύγκριση με τον αλγόριθμο της Προσομοιωμένης Ανόπτησης και τον Γενετικό Αλγόριθμο πάνω σε προβλήματα ελέγχου αποδοτικότητας (benchmark). Τέλος, στο ίδιο μοτίβο (Water Dynamics) οι Colak and Varol (2015) πρότειναν μια προσέγγιση βασισμένη στην περιστροφική κίνηση των κυμάτων του νερού (Circular Water Waves), που τα αποτελέσματά του σε συναρτήσεις έλεγχου αποδοτικότητας (test functions) δείχνουν ότι είναι πολλά υποσχόμενη μέθοδος.

2.6 Εφαρμογές Μεθόδων Εμπνευσμένων από τη Φύση

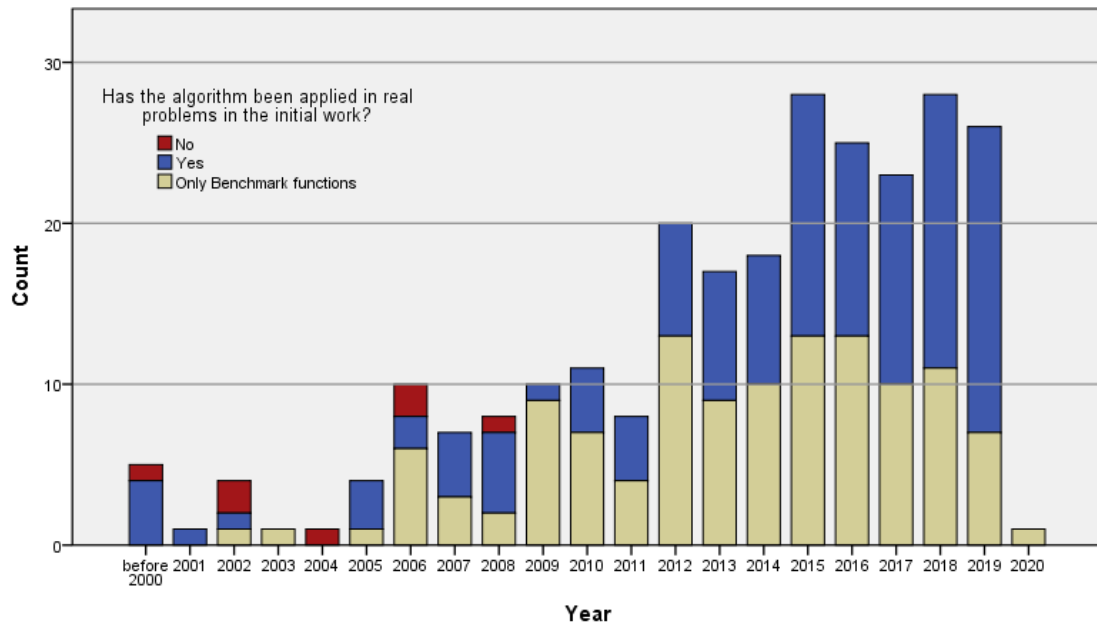
Σημαντικά συμπεράσματα για τη χρησιμότητα των μεθόδων που είναι Εμπνευσμένες από τη Φύση μπορούν να εξαχθούν μελετώντας το πλήθος και το είδος των εφαρμογών που λύνει κάθε μέθοδος.

Όπως φαίνεται στο Γράφημα 2-4, ένα πολύ μικρό ποσοστό από τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση δεν είχε κάποια εφαρμογή στην εργασία που παρουσιάστηκε ο αλγόριθμος. Η σημαντικότητα αυτής της παρατήρησης έγκειται στο γεγονός ότι έναυσμα για να αναπτύξει κανείς μια νέα μέθοδο πρέπει να αποτελεί είτε η διαφορετική προσέγγιση στην επίλυση προβλημάτων είτε ένα δυσεπίλυτο πρόβλημα που δεν μπορούν να λύσουν άλλες στρατηγικές.



Γράφημα 2-4: Κυκλικό διάγραμμα αναπαράστασης της ύπαρξης ή μη πραγματικής εφαρμογής στην αρχική εργασία παρουσίασης των μεθόδων

Το μεγαλύτερο ποσοστό των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση εφαρμόστηκε σε ένα πραγματικό πρόβλημα στην εργασία που παρουσιάστηκε η μέθοδος πρώτη φορά (Tzanetos and Dounias, 2019). Στο Γράφημα 2-5 παρατηρεί κανείς ότι η απουσία εφαρμογής στην αρχική εργασία παρουσίασης ενός Αλγορίθμου Εμπνευσμένου από τη Φύση εξαλείφθηκε το 2009. Έκτοτε έχουν αυξηθεί σταδιακά οι εργασίες που περιέχουν πραγματικά προβλήματα και έχουν ελαττωθεί εκείνες που οι αλγόριθμοι δοκιμάζονταν σε συναρτήσεις ελέγχου (benchmark functions).



Γράφημα 2-5: Εφαρμογή στην αρχική εργασία που παρουσιάζεται μια μέθοδος ανά έτος

Στις υποενότητες, παρακάτω, αναλύεται το πλήθος των εργασιών στο οποίο έχει χρησιμοποιηθεί κάθε κατηγορία αλγορίθμων ανά χώρο εφαρμογής. Σε παρένθεση είναι το πλήθος των εργασιών σε επιστημονικά περιοδικά. Η ανάλυση γίνεται ξεχωριστά ανά κατηγορία αλγορίθμων με σκοπό να εξαχθούν κάποια συγκριτικά συμπεράσματα στο τέλος. Στο Παράρτημα II της διατριβής παρατίθενται συγκεντρωτικοί πίνακες που σημειώνεται το πλήθος των δημοσιευμένων εργασιών ανά πεδίο προβλημάτων βελτιστοποίησης.

Πίνακας 2.2: Αλγόριθμοι που θεωρούνται ότι τείνουν να καθιερωθούν

Αλγόριθμος	Αναφορές	Αριθμός εργασιών με εφαρμογές	Κατηγορία
Αλγόριθμος Πυγολαμπίδας	≈1368	>100	ΝΣ
Βελτιστοποιητής Γκρίζου Λύκου	≈2194	>100	ΝΣ
Βελτιστοποίηση βασισμένη στη Βιογεωγραφία	≈2280	>150	ΝΕΟ
Αλγόριθμος Νυχτερίδας	≈2697	>150	ΝΣ
Αναζήτηση Κούκου	≈3759	>150	ΝΣ
Αλγόριθμος Βαρυτικής Αναζήτησης	≈3670	>150	ΝΕΦΦΝΕ
Αλγόριθμος Ανακατεμένου Άλματος Βατράχων	≈768	>150	ΝΕΟ
Βελτιστοποίηση Ηλεκτρομαγνητισμού	≈774	>150	ΝΕΦΦΝΕ
Αλγόριθμος Επικοινωνίας Λουλουδιών	≈930	>100	ΝΕΟ
Βελτιστοποίηση Χωροκατακτητικού Ζιζανίου	≈959	>100	ΝΕΟ

ΝΣ: Νοημοσύνη Σμήνους, ΝΕΟ: Νοημοσύνη Εμπνευσμένη από Οργανισμούς, ΝΕΦΦΝΕ: Νοημοσύνη Εμπνευσμένη από Φυσικά Φαινόμενα και Νόμους των Επιστημών

Σε κάθε κατηγορία γίνεται μια αναφορά στους καθιερωμένους (established) αλγόριθμους, δηλαδή εκείνους που έχουν προταθεί πάνω περίπου 20 χρόνια πριν και αριθμούν εκατοντάδες ή χιλιάδες εργασίες με εφαρμογές. Στον Πίνακα 2.2 παρουσιάζονται συνοπτικά οι αλγόριθμοι που τείνουν να γίνουν καθιερωμένοι, καθώς οι εργασίες που περιέχουν εφαρμογές ξεπερνάνε τις 100 σε κάθε έναν. Ο αριθμός των εργασιών αυτών αυξάνεται καθημερινά, οπότε οι συγκεκριμένοι αλγόριθμοι δεν έχουν συμπεριληφθεί στους πίνακες του Παράρτημα II.

2.6.1 Εφαρμογές Αλγορίθμων Εμπνευσμένων από τη Νοημοσύνη Σμήνους

Ο πρώτος αλγόριθμος Νοημοσύνης Σμήνους ήταν η Αναζήτηση Στοχαστικής Διάχυσης (Stochastic Diffusion Search), η οποία παρουσιάστηκε από τον J. M. Bishop (1989) και είναι εμπνευσμένη από την επικοινωνία ενός-προς-ένα (one-by-one) ενός είδους μυρμηγκιών που ονομάζονται *Leptothorax acervorum*. Στη βιβλιογραφία μπορεί να βρει κανείς πολλές εφαρμογές αυτής της μεθόδου, όπως η Επιλογή Χαρακτηριστικών (feature selection) (V. Bhasin, P. Bedi and A. Singhal, 2014), η αναγνώριση αντικειμένων (Bishop and Torr, 1992), η αυτο-τοπικοποίηση (self-localization) κινητών ρομπότ (Beattie and Bishop, 1998), η αναγνώριση εικόνων (image recognition) σε εφαρμογές ιατρικής (M. M. al-Rifaie and A. Aber, 2012) και η επιλογή θέσης για ασύρματα δίκτυα (Hurley and Whitaker, 2002).

Η Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization) προτάθηκε ως μέθοδος επίλυσης προβλημάτων διακριτού χώρου, όπως το γνωστό Πρόβλημα του Περιοδούντος Πωλητή (Dorigo and Stützle, 1999), προβλήματα δρομολόγησης οχημάτων (Fountas and Vlachos, 2005), προβλήματα χρονοδρομολόγησης (scheduling problems) (Fidanova and Durchova, 2006; Gutjahr and Rauner, 2007; W. Wen et al., 2015), προβλήματα ανάθεσης (assignment problem) (Stützle and Dorigo, 1999), το πρόβλημα επικάλυψης συνόλου (Set Covering) (Lessing, Dumitrescu and Stützle, 2004), αλλά εφαρμόστηκε και σε άλλα πεδία, όπως στη ρομποτική (Garcia et al., 2009), σε προβλήματα κατηγοριοποίησης (D. Martens et al., 2007), στην πρωτεϊνική αναδίπλωση (protein folding) (Shmygelska and Hoos, 2005) κ.ά.

Η Βελτιστοποίηση Σμήνους Σωματιδίων (BSS) έχει εφαρμοστεί σε ένα ευρύ φάσμα προβλημάτων βελτιστοποίησης, όπως σχεδιασμό κεραιών (antennas design) (H. Wu et al., 2009), στη βιο-ιατρική (R. C. Eberhart and Xiaohui Hu, 1999), στα δίκτυα επικοινωνιών (M. Shen et al., 2014), στα χρηματο-οικονομικά (J. Nenortaitė and R. Simutis, 2005), στα συστήματα ενέργειας (A. A. A. Esmin and G. Lambert-Torres, 2006), στη ρομποτική (Zhang, Gong and Zhang, 2013), σε προβλήματα χρονοδρομολόγησης (Liao, Chao-Tang Tseng and Luarn, 2007), αλλά και σε προβλήματα κατηγοριοποίησης/συσταδοποίησης (classification/clustering) (B. Xue, M. Zhang and W. N. Browne, 2013) και πρόβλεψης (Kuo et al., 2009; Wang et al., 2010).

Οι τρεις προαναφερθείσες τεχνικές δεν έχουν συμπεριληφθεί στην επιμέρους ανάλυση των εφαρμογών των αλγορίθμων ΝΣ, καθώς θεωρούνται καθιερωμένοι (established) αλγόριθμοι που έχουν χρησιμοποιηθεί σε αμέτρητες εργασίες και έχουν επιλύσει πολλά προβλήματα. Επιπλέον, καθιερωμένες μέθοδοι μπορούν να θεωρηθούν η Τεχνητή Αποικία Μελισσών (Artificial Bee Colony) και η Βελτιστοποίηση Τροφοληψίας Βακτηρίων (Bacterial Foraging Optimization) λόγω των αναρίθμητων εργασιών στις οποίες έχουν χρησιμοποιηθεί.

Ο Πίνακας 2.3 σε συνδυασμό με τις μεθόδους ΝΣ που συμπεριλαμβάνονται στον Πίνακα 2.2 αποτελεί μια εικόνα για το ποιες μέθοδοι αποτελούν χρήσιμα εργαλεία. Πέρα από την Αναζήτηση Κούκου (Cuckoo Search), τον Αλγόριθμο Νυχτερίδας (Bat Algorithm), τον Αλγόριθμο Πυγολαμπίδας (Firefly Algorithm) και τον Βελτιστοποιητή Γκριζού Λύκου (Grey Wolf Optimizer), που περιλαμβάνονται στον Πίνακα 2.3, εξίσου πολλές εφαρμογές εμφανίζει και ο Αλγόριθμος των Μελισσών (The Bees Algorithm). Σημαντική λεπτομέρεια το γεγονός ότι οι τρεις (3) εκ των αλγορίθμων που αναφέρονται σε αυτή την παράγραφο έχουν προταθεί από τον Καθηγητή Xin-She Yang, ο οποίος έχει εργαστεί σε μεγάλο βαθμό πάνω στις μαθηματικές βάσεις (mathematical foundations) που διέπουν τις μεθόδους που είναι εμπνευσμένες από τη Φύση (Yang and He, 2019b) και διεξάγει έρευνα πάνω στις εφαρμογές τους (Yang, 2012b, 2013; Yang and He, 2019a).

Πίνακας 2.3: Δημοσιευμένες εργασίες Αλγορίθμων Νοημοσύνης Σμήνους ανά κατηγορία προβλημάτων

	Βελτιστοποίηση	Κατηγοριοποίηση	Άλλη
Τροφοληψία	317 (217)	48 (32)	66 (34)
Κοινωνική Συμπεριφορά	229 (144)	57 (38)	78 (53)
Άλλη συμπεριφορά σμήνους	287 (194)	35 (28)	78 (49)
Σύνολο	833 (555)	140 (98)	222 (136)

Οι αλγόριθμοι ΝΣ ακολουθούν την τάση που επικρατεί στους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση, όπου η πλειοψηφία τους εφαρμόζεται σε προβλήματα βελτιστοποίησης. Το 41,7% των αλγορίθμων ΝΣ έχει χρησιμοποιηθεί στην επίλυση προβλημάτων κατηγοριοποίησης (classification) ή συσταδοποίησης (clustering), όπου η πλειοψηφία αυτών έχει χρησιμοποιηθεί σε 1-5 εργασίες με σχετικά προβλήματα. Το είδος της στρατηγικής που αποτελεί πηγή έμπνευσης σε κάθε αλγόριθμο δείχνει να μην σχετίζεται με το είδος των προβλημάτων που λύνουν οι αλγόριθμοι κάθε κατηγορίας.

Μια λεπτομέρεια που ενδιαφέρει τους ερευνητές είναι το ποια μέθοδος είναι καλύτερη για κάθε πρόβλημα ή έστω για κατηγορίες προβλημάτων. Αυτή η πληροφορία μπορεί να εξαχθεί από τον Πίνακα II-1. Εκτός του Αλγορίθμου Νυχτερίδας (Bat Algorithm), της Αναζήτησης Κούκου (Cuckoo Search), του Αλγορίθμου Πυγολαμπίδας (Firefly Algorithm), του Αλγορίθμου Σμήνους Ψαριών

(Fish-swarm Algorithm), τη Βελτιστοποίηση Γκριζού Λύκου (Grey Wolf Optimizer), του Αλγόριθμου των Μελισσών (The Bees Algorithm) και του Αλγόριθμου Βελτιστοποίησης Φάλαινας (Whale Optimization Algorithm), που έχουν μεγάλο πλήθος εφαρμογών, οι υπόλοιποι αλγόριθμοι δεν έχουν χρησιμοποιηθεί στον ίδιο βαθμό με τις προαναφερόμενες.

Χαρακτηριστικές εξαιρέσεις αποτελούν:

- ✓ η Βελτιστοποίηση Αποικίας Μελισσών (Bee Colony Optimization) η οποία συγκεντρώνει πολλές εργασίες σε προβλήματα δρομολόγησης
- ✓ η Βελτιστοποίηση Σμήνους Κότας (Chicken Swarm Optimization), ο Αλγόριθμος Λιβελούλας (Dragonfly Algorithm), η Βελτιστοποίηση Κοπαδιού Ελεφάντων (Elephant Herding Optimization), η Βελτιστοποίηση Νυχτοπεταλούδας και Φλόγας (Moth-flame Optimization), ο Αλγόριθμος Σμήνους Σάλπας (Salp Swarm Algorithm) και ο Αλγόριθμος Βελτιστοποίησης του Πιθήκου Αράχνης (Spider Monkey Optimization Algorithm), όπου όλοι έχουν πολλές εφαρμογές σε προβλήματα ενέργειας
- ✓ ο αλγόριθμος Κοπάδι Κριλ (Krill Herd) με εφαρμογές σε προβλήματα ενέργειας και μηχανικού
- ✓ η Βελτιστοποίηση εμπνευσμένη από τα Περιστερία (Pigeon-inspired Optimization) με πολλές εφαρμογές μη επανδρωμένων οχημάτων (unmanned vehicles), οι οποίες εκτείνονται τόσο σε προβλήματα συσταδοποίησης όσο και σε προβλήματα βελτιστοποίησης.

2.6.2 Εφαρμογές Αλγορίθμων Εμπνευσμένων από Οργανισμούς

Αυτή η κατηγορία αλγορίθμων περιλαμβάνει τέσσερις μεθόδους που έχουν εφαρμοστεί σε ένα μεγάλο εύρος προβλημάτων: τον Βελτιστοποιητή Μηρμυλεοντίδας (Ant Lion Optimizer), τη Βελτιστοποίηση Φρουτόμυγας (Fruit Fly Optimization), τον Βελτιστοποιητή Αναζήτησης Ομάδας (Group Search Optimizer) και την Αναζήτηση Συμβιωτικών Οργανισμών (Symbiotic Organisms Search). Από τους Πίνακες του Παραρτήματος II απουσιάζει ο Αλγόριθμος Επιλογής Κλώνου, καθώς θεωρείται καθιερωμένος (Haktanirlar Ulutas and Kulturel-Konak, 2011).

Πίνακας 2.4: Δημοσιευμένες εργασίες Αλγορίθμων Εμπνευσμένων από Οργανισμούς ανά κατηγορία προβλημάτων

	Βελτιστοποίηση	Κατηγοριοποίηση	Άλλη
Πανίδα	1074 (834)	75 (58)	106 (84)
Χλωρίδα	298 (253)	28 (27)	14 (10)
Άλλα παραδείγματα	13 (12)	3 (2)	4 (3)
Σύνολο	1385 (1099)	106 (87)	124 (97)

Επίσης, οι μέθοδοι που αναφέρονται στον Πίνακα 2.2 (η Βελτιστοποίηση βασισμένη στη Βιογεωγραφία, ο Αλγόριθμος Επικονίασης Λουλουδιών, η Βελτιστοποίηση Χωροκατακτητικού Ζιζανίου και ο Αλγόριθμος Ανακατεμένου

Άλματος Βατράχων) αριθμούν αρκετές εφαρμογές. Εξίσου πολλές εφαρμογές εμφανίζονται στη Βελτιστοποίηση Σμήνους Γάτας, στον Αλγόριθμο Αναζήτησης Κόρακα, στη Βελτιστοποίηση Ζευγαρώματος Μελισσών και στον Γάμο Μελισσών.

Το ποσοστό των αλγορίθμων ΝΕΟ που έχουν εφαρμοστεί σε προβλήματα κατηγοριοποίησης και συσταδοποίησης είναι 26,8%, μικρότερο από το αντίστοιχο των μεθόδων ΝΣ. Οι διαφορές στα σύνολα των εργασιών των μεθόδων κάθε υποκατηγορίας, που φαίνονται στον Πίνακα 2.4, εξηγούνται από την μεγάλη απόκλιση του αριθμού των μεθόδων που ανήκουν σε κάθε μια.

2.6.3 Εφαρμογές Αλγορίθμων Εμπνευσμένων από Φυσικά Φαινόμενα και Νόμους των Επιστημών

Από τους γνωστότερους και ο πρώτος χρονικά αλγόριθμος που αντλούσε έμπνευση από τη φύση είναι η Προσομοιωμένη Ανόπτηση (Simulated Annealing), η οποία παρουσιάστηκε από τους Kirkpatrick, Gelatt and Vecchi (1983). Ανάμεσα στις εκατοντάδες εφαρμογές της μεθόδου αυτής είναι η χρονοδρομολόγηση εργασιών (van Laarhoven, Aarts and Lenstra, 1992), η κατανομή πόρων (Boctor, 1996; Anagnostopoulos and Koulinas, 2010), τα προβλήματα δέσμευσης μονάδων ενέργειας (unit commitment) (F. Zhuang and F. D. Galiana, 1990), προβλήματα συσταδοποίησης (Selim and Alsultan, 1991), η επεξεργασία εικόνων (Carnevali, Coletti and Patarnello, 1987), κατασκευαστικές εφαρμογές μηχανικού (Cunha Maria da Conceição and Sousa Joaquim, 1999) κ.ά.

Η Προσομοιωμένη Ανόπτηση δεν συμπεριλαμβάνεται στην επιμέρους ανάλυση των εφαρμογών των αλγορίθμων ΝΕΦΦΝΕ, καθώς είναι σαφές ότι είναι καθιερωμένη μέθοδος και συχνά χρησιμοποιείται στη βιβλιογραφία ως σημείο αναφοράς για την μέτρηση της απόδοσης άλλων μεθόδων.

Πίνακας 2.5: Δημοσιευμένες εργασίες Αλγορίθμων Εμπνευσμένων από Φυσικά Φαινόμενα και Νόμους των Επιστημών ανά κατηγορία προβλημάτων

	Βελτιστοποίηση	Κατηγοριοποίηση	Άλλη
Φαινόμενα του Σύμπαντος	232 (192)	29 (22)	5 (5)
Νόμοι Θετικών Επιστημών	275 (208)	9 (4)	10 (10)
Φυσικά Φαινόμενα	58 (29)	1 (1)	0
Σύνολο	575 (435)	40 (29)	15 (14)

Όπως φαίνεται παραπάνω, στον Πίνακα 2.5, η έρευνα σχετικά με τους αλγόριθμους που είναι εμπνευσμένοι από Φυσικά Φαινόμενα και Νόμους των Θετικών Επιστημών απέδωσε μια αντιπροσωπευτική εικόνα για τη χρησιμότητα των Nature Inspired αλγορίθμων σε προβλήματα βελτιστοποίησης. Παρόλο που ολοκληρωμένη εικόνα για αυτή τη χρησιμότητα θα υπάρξει στο πέρας της διατριβής, ήδη είναι προφανές από το πλήθος των εργασιών ότι οι μέθοδοι που πραγματεύεται η

παρούσα διατριβή είναι ένα πολύ καλό εργαλείο για την επίλυση προβλημάτων βελτιστοποίησης.

Στον παραπάνω πίνακα (αλλά και στον Πίνακα Π-3) δεν περιέχονται οι σχετικές δημοσιεύσεις της Προσομοιωμένης Ανόπτωσης (Simulated Annealing), καθώς είναι συντριπτικά πολύ περισσότερες από όλες τις υπόλοιπες μεθόδους μαζί. Από τους εν λόγω Πίνακες θα μπορούσαν να απαλειφθούν επίσης ο Αλγόριθμος Βαρυτικής Αναζήτησης (Gravitational Search Algorithm) και ο Αλγόριθμος Αναζήτησης Αρμονίας (Harmony Search), οι οποίοι αισίως ξεπέρασαν τις 200 εφαρμογές έκαστος.

2.7 Συμπεράσματα

Σκοπός αυτής της ενότητας είναι να διερευνήσει πόσοι και ποιοι είναι οι αλγόριθμοι που είναι εμπνευσμένοι από τη φύση, καθώς και να απαντήσει τι είδους προβλήματα μπορούν να λύσουν;

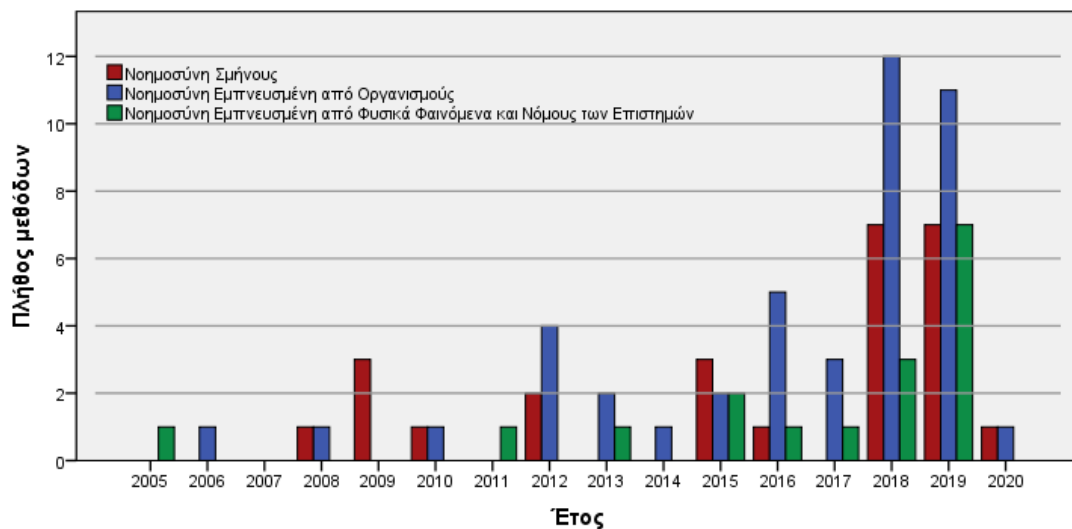
Στις ενότητες του παρόντος Κεφαλαίου παρουσιάζονται εν συντομία όλοι οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση, οι οποίοι επιμερίστηκαν με βάση την πηγή έμπνευσής τους. Στην ενότητα 2.6 μάλιστα, συνοψίζονται οι χώροι εφαρμογής των εν λόγω αλγορίθμων, ενώ στους Πίνακες του Παραρτήματος II φαίνονται αναλυτικότερα οι δημοσιευμένες εργασίες ανά χώρο εφαρμογής για κάθε αλγόριθμο ξεχωριστά.

Δύο ερωτήματα που γεννώνται είναι (α) πόσες εκ των μεθόδων (και ποιες) δεν έχουν εφαρμοστεί σε κάποιο πρόβλημα και (β) τι διαφορές υπάρχουν μεταξύ των μεθόδων με ίδια πηγή έμπνευσης (και συχνά παρόμοιο όνομα). Αυτά τα ερωτήματα απαντάει η ανάλυση που γίνεται στην παρούσα ενότητα.

2.7.1 Συμπεράσματα επί των εφαρμογών των Αλγορίθμων Εμπνευσμένων από τη Φύση

Ένα πρόβλημα που παρατηρείται είναι η απουσία εφαρμογών σε αρκετές μεθόδους, το οποίο είναι λογικό για τις πιο πρόσφατα δημοσιευμένες, αλλά παρατηρείται και σε αρκετές μεθόδους που έχουν δημοσιευθεί προ δύο και περισσότερων ετών. Συνολικά, 31,3% των μεθόδων ΝΣ, 42,6% των μεθόδων ΝΕΟ και 38,6% των μεθόδων ΝΕΦΦΝΕ δεν έχουν λύσει κάποιο πρόβλημα βελτιστοποίησης. Τα αντίστοιχα ποσοστά για προβλήματα κατηγοριοποίησης /συσταδοποίησης είναι 57,8%, 73,1% και 62,8% αντίστοιχα, ενώ για τα υπόλοιπα είδη προβλημάτων είναι 51,8%, 69,4% και 71,9% αντίστοιχα. Ασχέτως κατηγορίας προβλήματος, το 27,7% των μεθόδων ΝΣ, το 37% των μεθόδων ΝΕΟ και το 26,9% των μεθόδων ΝΕΦΦΝΕ αριθμούν μηδενικές εφαρμογές. Ένας στους τρεις αλγόριθμους δεν εμφανίζει κάποια εφαρμογή από την στιγμή που παρουσιάζεται.

Ωστόσο, η πλειοψηφία των αλγορίθμων που δεν παρουσιάζουν εργασίες στις οποίες να εφαρμόζονται σε κάποιο πρόβλημα αποτελείται από μεθόδους που έχουν προταθεί πρόσφατα, όπως φαίνεται στο Γράφημα 2-6, και κατ' επέκταση είναι σχετικά αναμενόμενο να μην έχουν εμφανιστεί σε περαιτέρω δημοσιεύσεις. Οι αλγόριθμοι που έχουν προταθεί τουλάχιστον μια δεκαετία πριν και δεν έχουν εφαρμογές είναι ο Αλγόριθμος Βαρύτητας Διαστήματος (Space Gravitational Algorithm) του 2005, ο Αλγόριθμος Δενδρυλλίων που Αναπτύσσονται (Saplings Growing up Algorithm) του 2006, ο Αλγόριθμος Συλλογής Γύρης από Μέλισσες (Bee Collecting Pollen Algorithm) και η Βελτιστοποίηση Αλμάτων Βατράχων (Jumping Frogs Optimization) του 2008, ο αλγόριθμος Βομβίνου (Bumblebees algorithm) και η Βελτιστοποίηση Συντρόφου Δελφινιού (Dolphin Partner Optimization) του 2009, το Μοντέλο Ιεραρχικού Σμήνους (Hierarchical Swarm Model) και ο Αλγόριθμος Μετενσάρκωσης (Reincarnation Algorithm) του 2010. Ενδιαφέρον θα είχε να διερευνηθεί κανείς την απόδοση των προαναφερόμενων μεθόδων σε διάφορα είδη προβλημάτων.

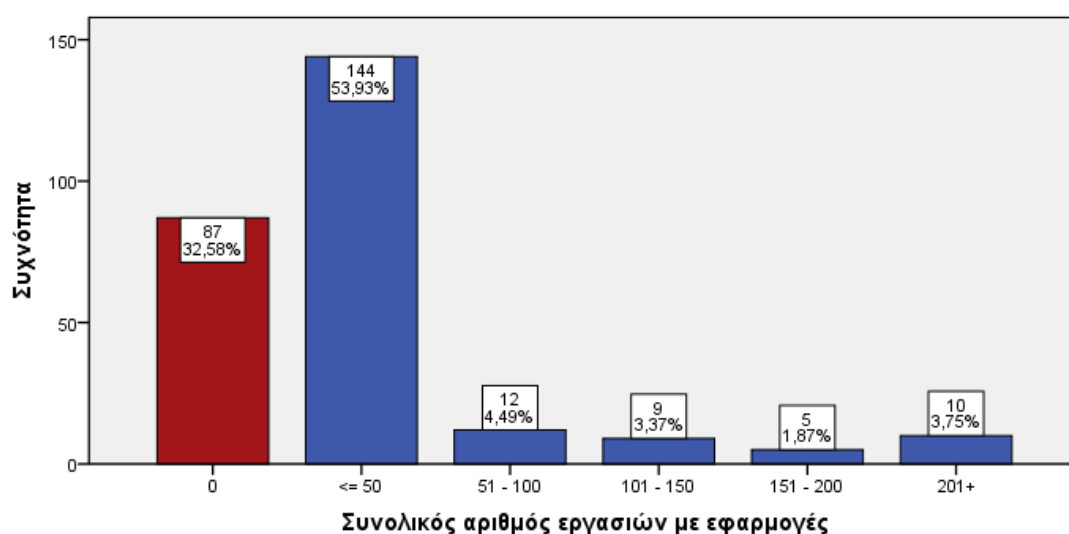


Γράφημα 2-6: Έτος εμφάνισης των μεθόδων που δεν έχουν εφαρμογές στη βιβλιογραφία

Μια μεγάλη μερίδα ερευνητών αντιτάσσεται στην παρουσία των μεθόδων που είναι Εμπνευσμένες από τη Φύση, καθώς θεωρούν ότι δεν προσφέρουν κάποια αξία στην επίλυση προβλημάτων. Ωστόσο, οι εν λόγω ερευνητές συγκρίνουν τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση με άλλες μεθόδους Υπολογιστικής Νοημοσύνης, όπως τα Τεχνητά Νευρωνικά Δίκτυα. Όπως φαίνεται στους Πίνακες Π-1 έως Π-3, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση χρησιμοποιούνται κατά κύριο λόγο σε προβλήματα βελτιστοποίησης. Μια μικρή μερίδα αλγορίθμων έχει χρησιμοποιηθεί σε άλλες κατηγορίες προβλημάτων, π.χ. κατηγοριοποίησης, συσταδοποίησης και πρόβλεψης, στα οποία τα Τεχνητά Νευρωνικά Δίκτυα και άλλες μέθοδοι Μηχανικής Μάθησης προσφέρουν καλύτερη απόδοση. Ειδικότερα, το 33,3% των αλγορίθμων που μελετήθηκαν στα πλαίσια της διατριβής έχουν τουλάχιστον μια εφαρμογή σε προβλήματα κατηγοριοποίησης ή

συσταδοποίησης. Επιπλέον, το 38,5% των αλγορίθμων παρουσιάζουν τουλάχιστον μια εφαρμογή σε άλλα πεδία προβλημάτων.

Μια παρατήρηση που μπορεί διατυπωθεί αναλύοντας τους Πίνακες του Παραρτήματος II είναι το γεγονός ότι δεν εμφανίζονται πολλές εφαρμογές σε προβλήματα χρηματοοικονομικών, όπως η βελτιστοποίηση χαρτοφυλακίου (Portfolio optimization), η παρακολούθηση ενός δείκτη (index tracking), η πρόβλεψη των αποδόσεων των μετοχών (stock prediction) και η πρόβλεψη μιας αγοράς (market forecasting). Από τα παραπάνω προβλήματα, η πρόβλεψη των αποδόσεων των μετοχών και η πρόβλεψη μιας αγοράς συμπεριλαμβάνονται στην κατηγορία “Άλλα” προβλήματα. Ωστόσο, το ποσοστό ύπαρξης τέτοιων εφαρμογών σε αυτή την κατηγορία είναι αναλόγως χαμηλό με εκείνο των προβλημάτων “Χρηματοοικονομικής”.



Γράφημα 2-7: Ομαδοποίηση των αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση με βάση το πλήθος των δημοσιεύσεων που επιλύουν πραγματικά προβλήματα

Όπως φαίνεται στο Γράφημα 2-7, η πλειοψηφία των Νοημόνων Μεθόδων που είναι Εμπνευσμένες από τη Φύση έχει λιγότερες από 50 εφαρμογές. Το εν λόγω Γράφημα αναφέρεται στο σύνολο των εργασιών που μια μέθοδος έχει επιλύσει κάποιο πρόβλημα, ασχέτως του είδους του προβλήματος ή της κατηγορίας προβλημάτων που ανήκει. Με βάση το συγκεκριμένο Γράφημα είναι πιο ευδιάκριτη η διαφοροποίηση των καθιερωμένων (established) μεθόδων από τις υπόλοιπες.

Πιο συγκεκριμένα, δέκα αλγόριθμοι φαίνεται να ξεπερνούν τις 200 εφαρμογές, οι οποίοι και αναφέρονται παραπάνω στην παρούσα ενότητα. Επιπλέον, 14 ακόμη μπορούν να θεωρηθούν οριακά καθιερωμένοι (near-established). Κατ' επέκταση, περίπου το 10% των μεθόδων Νοημοσύνης Εμπνευσμένης από τη Φύση έχουν αποδειχθεί καλά εργαλεία επίλυσης προβλημάτων.

2.7.2 Αλγόριθμοι με κοινή πηγή έμπνευσης

Μελετώντας τις ενότητες του παρόντος Κεφαλαίου διαπιστώνει κανείς πως αρκετές μέθοδοι φαίνεται να έχουν ίδια πηγή έμπνευσης ή ακόμη και να μοντελοποιούν το ίδιο φαινόμενο. Η ύπαρξη διαφορών ή μη στα βήματα κάθε μεθόδου από όσες μοντελοποιούν το ίδιο φαινόμενο ή έχουν κοινή πηγή έμπνευσης μπορεί να διερευνηθεί αν αναλυθεί η διαδικασία κάθε μίας από τις προαναφερόμενες μεθόδους ξεχωριστά. Αυτό θα απαιτούσε λεπτομερή ανάλυση κάθε αλγορίθμου ξεχωριστά, που για λόγους έκτασης είναι αδύνατο να γίνει στο παρόν κείμενο. Παρακάτω αναλύονται συγκεκριμένες περιπτώσεις αλγορίθμων που έχουν κοινή πηγή έμπνευσης και διερευνάται αν επιλύουν διαφορετικά προβλήματα ή αν είναι όλες εξίσου αποδοτικές.

Ο Αλγόριθμος Πιθήκου (Monkey Algorithm) είναι βασισμένος σε έναν πιθανολογικό (probabilistic) τρόπο σκαρφαλώματος των πιθήκων, ενώ η Αναζήτηση Πιθήκων (Monkey Search) πλησιάζει περισσότερο στην πραγματικότητα, όπου οι πιθήκοι σκαρφαλώνουν σε ένα κλαδί και τυχαία επιλέγουν το επόμενο κλαδί που βρίσκεται σε υψηλότερο επίπεδο από αυτό που βρίσκονται. Παρόλο που και οι δύο αυτές στρατηγικές βασίζονται στο ίδιο φυσικό ανάλογο, ο Αλγόριθμος Πιθήκου χρησιμοποιήθηκε περισσότερες φορές στη βιβλιογραφία, με 18 δημοσιευμένες εργασίες έναντι μόλις τριών (3) της Αναζήτησης Πιθήκων. Μάλιστα, η πλειονότητα των εφαρμογών του Αλγορίθμου Πιθήκου είναι στην βέλτιστη τοποθέτηση αισθητήρων (optimal sensor placement) (Yi et al., 2014, 2015), ενώ εμφανίζει επίσης εφαρμογές προβλημάτων συσταδοποίησης (Zhang, Zhang and Sun, 2011; Hodashinsky and Samsonov, 2017).

Παρόλο που και η Βελτιστοποίηση Σμήνους Κατσαρίδων (Cockroach Swarm Optimization) και ο Αλγόριθμος Επιδρομής Κατσαρίδων (Roach Infestation Algorithm) είναι βασισμένοι στον τρόπο εύρεσης τροφής των κατσαρίδων, καμία εκ των δύο μεθόδων δεν δείχνει να αποτελεί αξιόπιστη τεχνική. Και οι δύο αυτές μέθοδοι εμφανίζουν ελάχιστες εφαρμογές, με τη Βελτιστοποίηση Σμήνους Κατσαρίδων να εφαρμόζεται κυρίως σε προβλήματα δρομολόγησης και Πλανόδιου Πωλητή. Ο αλγόριθμος Τεχνητών Κοινωνικών Κατσαρίδων (Artificial Social Cockroaches) έχει μόνο μια εργασία, στην οποία εφαρμόζεται στην επαναφορά δεδομένων (information retrieval) (Bouarara, Hamou and Abdelmalek, 2018). Τέλος, η Εξέλιξη Σμήνους κατσαρίδων (Cockroach Swarm Evolution) και η Βελτιστοποίηση Αποικίας Κατσαρίδων (Cockroach Colony Optimization) δεν έχουν επιλύσει κάποιο πραγματικό πρόβλημα.

Η Βελτιστοποίηση Τροφοληψίας Βακτηρίων (Bacteria Foraging Optimization), όπως αναφέρεται και παραπάνω, θεωρείται καθιερωμένη μέθοδος με πολλές εφαρμογές. Επιπλέον, η Χημειοταξία Βακτηρίων (Bacterial Chemotaxis) και ο Αλγόριθμος Σμήνους Βακτηρίων (Bacterial Swarming Algorithm) έχουν εφαρμοστεί κατά κύριο λόγο σε προβλήματα ενέργειας, όπως η οικονομική κατανομή φορτίου

(economic load dispatch). Η Χημειοταξία Βακτηρίων έχει εφαρμογές σε προβλήματα πρόβλεψης, επίσης. Ωστόσο, κανένας εκ των δύο αλγορίθμων δεν πλησιάζει σε σύνολο τις δημοσιευμένες εργασίες της Βελτιστοποίησης Τροφοληψίας Βακτηρίων.

Μεταξύ των μεθόδων που βασίζονται σε συμπεριφορές των πιγκουίνων, ο Αλγόριθμος Βελτιστοποίησης Αναζήτησης Πιγκουίνων (Penguins Search Optimization Algorithm) αποδεικνύεται πιο χρήσιμος (σύμφωνα και με τους πίνακες του Παραρτήματος II). Ωστόσο, κανείς εκ των τριών μεθόδων αυτής της πηγής έμπνευσης δεν ξεπερνάει τις 10 δημοσιεύσεις που επιλύουν πραγματικά προβλήματα. Ο Αλγόριθμος Βελτιστοποίησης Αναζήτησης Πιγκουίνων έχει έξι (6) εργασίες, δύο (2) έχει ο Βελτιστοποιητής Αυτοκρατορικού Πιγκουίνου (Emperor Penguin Optimizer), ενώ η Αποικία Αυτοκρατορικών Πιγκουίνων (Emperor Penguins Colony) δεν έχει χρησιμοποιηθεί σε κάποια εφαρμογή, όντας πιο πρόσφατη μέθοδος.

Οι αλγόριθμοι που βασίζονται στις μέλισσες συνθέτουν μια μεγάλη (ίσως την μεγαλύτερη) ομάδα μεθόδων με κοινή πηγή έμπνευσης. Ο αλγόριθμος της Κυψέλης (BeeHive algorithm) προτάθηκε για την επίλυση προβλημάτων δρομολόγησης, λόγω του μοντέλου επικοινωνίας μεταξύ μελισσών στο οποίο στηρίζεται. Ο συγκεκριμένος αλγόριθμος αποτελεί μια από τις ελάχιστες εξαιρέσεις, όπου ενώ έχει παρόμοια ή ίδια πηγή έμπνευσης με άλλες μεθόδους, προτάθηκε και επιλύει ένα συγκεκριμένο είδος προβλημάτων, όπου η πλειοψηφία των εφαρμογών του είναι σε προβλήματα δικτύων (network problems) (Wang et al., 2008; K. Kiran et al., 2014). Παρόμοια είναι η περίπτωση της Βελτιστοποίησης Αποικίας Μελισσών (Bee Colony Optimization), η οποία έχει εφαρμοστεί κατά κύριο λόγο σε Προβλήματα Πλανόδιου Πωλητή, προβλήματα ανάθεσης (assignment) και χρονοδρομολόγησης (scheduling).

Ο Αλγόριθμος Των Μελισσών (The Bees Algorithm) είναι μια εκ των πιο διαδεδομένων τεχνικών που βασίζονται στις μέλισσες, απαριθμώντας 71 εργασίες με εφαρμογές, από τις οποίες οι 32 επιλύουν κατασκευαστικά προβλήματα μηχανικού. Ένας εξίσου διαδεδομένος αλγόριθμος είναι ο Γάμος Μελισσών (Marriage in Honey Bees) με 70 εργασίες, εκ των οποίων οι 51 είναι επίσης σε κατασκευαστικά προβλήματα μηχανικού. Η Βελτιστοποίηση Μελισσών (Honey Bees Optimization) είναι επίσης καλή μέθοδος βελτιστοποίησης (απαριθμώντας 69 εργασίες), η οποία έχει εφαρμοστεί κυρίως σε προβλήματα ενέργειας (στις 36 εξ αυτών των εργασιών). Η Τεχνητή Αποικία Μελισσών (Artificial Bee Colony) δεν μπαίνει στην ίδια βάση σύγκρισης όντας καθιερωμένη μέθοδος στη βιβλιογραφία.

Σε αντίθεση με τις προαναφερόμενες μεθόδους, ο αλγόριθμος Βομβίνου (Bumblebees algorithm) και ο Αλγόριθμος Συλλογής Γύρης Μελισσών (Bee Pollen Collecting Algorithm) δεν έχουν χρησιμοποιηθεί σε κάποια πραγματική εφαρμογή, παρόλο που έχουν παρουσιαστεί το 2009 και το 2008 αντίστοιχα. Η Βελτιστοποίηση Σμήνους Μελισσών (Bee Swarm Optimization), ο αλγόριθμος Βελτιστο-Μελισσών (OptBees) και η Βελτιστοποίηση Ζευγαρώματος Βομβίνων (Bumblebees Mating Optimization) εμφανίζονται σε πολύ λίγες εργασίες.

Μεταξύ των αλγορίθμων που είναι εμπνευσμένοι από τα δελφίνια, ο Ηχοεντοπισμός Δελφινιών (Dolphin Echolocation) έχει μακράν τις περισσότερες εφαρμογές. Τόσο ο Αλγόριθμος Σμήνους Δελφινιών (Dolphin Swarm Algorithm) όσο και η Βελτιστοποίηση Δεξαμενής Δελφινιών (Dolphin Pod Optimization) δεν μπορούν να συναγωνιστούν τον Ηχοεντοπισμό Δελφινιών, καθώς έχουν μια ή δύο εφαρμογές. Η Βελτιστοποίηση Συντρόφου Δελφινιού (Dolphin Partner Optimization) και ο Αλγόριθμος Βελτιστοποίησης Σμήνους Δελφινιών (Dolphin Swarm Optimization Algorithm) δεν έχουν εφαρμοστεί σε κάποιο πρόβλημα πέρα από την αρχική τους εργασία.

Το ίδιο φαινόμενο, αλλά σε μικρότερη κλίμακα, παρατηρείται και στις μεθόδους που είναι εμπνευσμένες από σκαθάρια. Ο Αλγόριθμος Σκαθαριού Πίτι (Pity Beetle Algorithm) και ο Αλγόριθμος Βελτιστοποίησης Σμήνους Σκαθαριών (Beetle Swarm Optimization Algorithm) δεν έχουν εφαρμογές, ενώ ο Αλγόριθμος Αναζήτησης Κεραιών Σκαθαριού (Beetle Antennae Search Algorithm) έχει χρησιμοποιηθεί σε επτά (7) εργασίες, εκ των οποίων οι πέντε (5) λύνουν προβλήματα βελτιστοποίησης.

Ο Αλγόριθμος Βελτιστοποίησης Λέοντα (Lion Optimization Algorithm) και ο Αλγόριθμος του Λιονταριού (Lion's Algorithm) έχουν λιγότερες από πέντε (5) εφαρμογές έκαστος. Ο Βελτιστοποιητής Περηφάνιας Λέοντα (Lion Pride Optimizer) έχει χρησιμοποιηθεί για συσταδοποίηση (clustering) και για καταμερισμό εικόνας (image segmentation) (W. Li and M. Jiang, 2018), ενώ η Βελτιστοποίηση Περηφάνιας Λέοντα (Lion Pride Optimization) όντας πιο πρόσφατα δημοσιευμένη, δεν έχει εφαρμογές.

Ο Αλγόριθμος Ανακατεμένου Άλματος Βατράχων (Shuffled Frog Leaping Algorithm), ο αλγόριθμος Ιαπωνικού Δενδροβάτραχου (Japanese Tree Frogs) και η Βελτιστοποίηση Αλμάτων Βατράχων (Jumping Frogs Optimization), παρόλο που μοιράζονται την ίδια πηγή έμπνευσης (αλλά από διαφορετικά είδη βατράχων), διαφοροποιούνται αρκετά στις εφαρμογές τους. Ο Αλγόριθμος Ανακατεμένου Άλματος Βατράχων τείνει να γίνει καθιερωμένος αλγόριθμος έχοντας αρκετές εφαρμογές σε πρότυπα προβλήματα μηχανικού, ενώ ο αλγόριθμος Ιαπωνικού Δενδροβάτραχου έχει χρησιμοποιηθεί στα δίκτυα αισθητήρων (sensor networks) (Hernández and Blum, 2011) και στον χρωματισμό γράφων (graph coloring) (Hernández and Blum, 2012). Από την άλλη, η Βελτιστοποίηση Αλμάτων Βατράχων δεν έχει χρησιμοποιηθεί σε κάποια εφαρμογή, παρόλο που δημοσιεύθηκε το 2008.

Παρόμοια είναι και η περίπτωση του Βελτιστοποιητή Γκρίζου Λύκου (Grey Wolf Optimizer), του Αλγορίθμου Αγέλης Λύκων (Wolf Pack Algorithm) και της Αναζήτησης Λύκων (Wolf Search), όπου ο πρώτος έχει χρησιμοποιηθεί σε ένα εύρος εφαρμογών, ενώ οι άλλοι δύο εμφανίζουν πολύ λίγες συγκριτικά εφαρμογές. Επιπλέον, η ίδια κατάσταση παρατηρείται και μεταξύ του Αλγορίθμου Νυχτερίδας (Bat Algorithm) και του Αλγορίθμου Ηχοεντοπισμού Νυχτερίδων (Bat Sonar

Algorithm), όπου ο πρώτος είναι καθιερωμένος αλγόριθμος και ο δεύτερος δεν έχει εφαρμογές. Η σύγκριση μεταξύ Βελτιστοποίησης Αποικίας Μυρμηγκιών (Ant Colony Optimization) και Βελτιστοποίησης Μυρμηλεοντίδας (Ant Lion Optimization) δεν θα ήταν δίκαιη δεδομένης της δημοτικότητας της Βελτιστοποίησης Αποικίας Μυρμηγκιών. Ωστόσο, η Βελτιστοποίηση Μυρμηλεοντίδας εμφανίζεται σε περίπου 100 εργασίες, όπου εφαρμόζεται στην οικονομική κατανομή φορτίου (economic load dispatch), σε προβλήματα δρομολόγησης, σε προβλήματα χρονοπρογραμματισμού (scheduling) και σε πρότυπα προβλήματα μηχανικού.

Παρόλο που η Βελτιστοποίηση του Πιθήκου Αράχνης (Spider Monkey Optimization) είναι μεταγενέστερη των άλλων δύο μεθόδων που βασίζονται στους πίθηκους (δημοσιεύθηκε το 2014, ενώ οι άλλες δύο το 2007 και 2008), απαριθμεί 38 εργασίες. Ο Αλγόριθμος Πιθήκου (Monkey Algorithm) έχει χρησιμοποιηθεί σε 21 εργασίες, ενώ η Αναζήτηση Πιθήκου (Monkey Search) μόλις τρεις (3).

2.7.3 Συνοπτικά συμπεράσματα

Συνοψίζοντας, στη βιβλιογραφία εμφανίζονται πολλές μέθοδοι, ωστόσο οι διαφορετικές μεθοδολογίες είναι ελάχιστες. Στη συντριπτική τους πλειοψηφία, οι μέθοδοι που συναντώνται στη βιβλιογραφία δεν έχουν κάποια καινοτομία ως προς την προσέγγιση ή την επίλυση των προβλημάτων που καλούνται να λύσουν. Αντιθέτως, η μεθοδολογία τους μοιάζει σε μεγάλο βαθμό (ή ακόμη και βασίζεται εξ ολοκλήρου) σε καθιερωμένες μεθόδους, όπως η Βελτιστοποίηση Σμήνους Σωματιδίων και η Βελτιστοποίηση Αποικίας Μυρμηγκιών. Περαιτέρω ανάλυση γι' αυτή την παρατήρηση γίνεται στο επόμενο κεφάλαιο.

Στο παρόν κεφάλαιο χωρίζονται οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση σε κατηγορίες με βάση την κύρια πηγή έμπνευσής τους σε: (α) Νοημοσύνη Σμήνους (Swarm Intelligence), (β) Αλγόριθμους Εμπνευσμένους από Οργανισμούς (Organisms-based) και (γ) Αλγόριθμους Εμπνευσμένους από Φυσικά Φαινόμενα και Νόμους των Επιστημών (Physical Phenomena and Laws of Science). Επιπρόσθετα, γίνεται μια πρωταρχική ανάλυση των προβλημάτων στα οποία εφαρμόζονται, η οποία ωστόσο δεν δείχνει να σχετίζεται η πηγή έμπνευσης με τα είδη προβλημάτων τα οποία μπορεί να λύσει ένας Αλγόριθμος Εμπνευσμένος από τη Φύση.

Ο βασικός στόχος των νοημόνων μεθόδων που παρουσιάζονται στην παρούσα διατριβή είναι να ψάξουν τον χώρο λύσεων για μια βέλτιστη λύση, η οποία είναι δύσκολο να εντοπιστεί όταν αυτός ο χώρος είναι πολυεπίπεδος και έχει πολλά τοπικά βέλτιστα. Η διαφορά της μεθοδολογίας που εφαρμόζει κάθε αλγόριθμος βασίζεται στον τρόπο που θα γίνει αυτή η αναζήτηση στον χώρο λύσεων.

3. Νοήμονες Αλγόριθμοι Εμπνευσμένοι από τη Φύση στην Επίλυση Προβλημάτων Βελτιστοποίησης

3.1 Προβλήματα Βελτιστοποίησης

Η βελτιστοποίηση, γενικότερα, ενέχει την έννοια της εύρεσης της βέλτιστης λύσης μεταξύ ενός συνόλου λύσεων, με σκοπό την επίλυση ενός προβλήματος. Τέτοια προβλήματα συναντώνται στον χώρο της Λήψης Αποφάσεων, της Μηχανικής, της Διοίκησης κλπ. Ωστόσο, αν θέλαμε να αποδώσουμε μαθηματικά την έννοια της βελτιστοποίησης, σύμφωνα με τους Boyd and Vandenberghe, 2004; Yang and He (2019b), αυτή θα ήταν:

$$\underset{\vec{x}}{\text{βελτιστοποίηση της } f(\vec{x})}, \quad \vec{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d \quad (3.1.1)$$

όπου $f(\vec{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ είναι η αντικειμενική συνάρτηση που πρέπει να βελτιστοποιηθεί ως προς \vec{x} . Με x_1, x_2, \dots, x_d συμβολίζονται οι μεταβλητές απόφασης του προβλήματος, όπου d είναι το πλήθος των μεταβλητών απόφασης.

Συνήθως η παραπάνω σχέση παρουσιάζεται ως ελαχιστοποίηση (minimization) της $f(\vec{x})$ και επισημαίνεται ότι εφόσον χρησιμοποιηθεί η $-f(\vec{x})$, μπορεί να λυθεί ένα πρόβλημα μεγιστοποίησης (maximization) ως πρόβλημα ελαχιστοποίησης. Ωστόσο, η σχέση (3.1.1) εδώ αποτυπώνει έναν γενικό φορμαλισμό των προβλημάτων βελτιστοποίησης.

Γενικότερα, ως πρόβλημα βελτιστοποίησης μπορεί να ορισθεί ένα πρόβλημα, στο οποίο καλούμαστε να βρούμε τις τιμές εκείνες των x_i (όπου $i \in 1, 2, \dots, d$) που βελτιστοποιούν (είτε στην μέγιστη είτε στην ελάχιστη τιμή, ανάλογα το πρόβλημα) την $f(x_i)$. Κατ' επέκταση, αν μιλάμε για προβλήματα Λήψης Απόφασης (Decision Making), ο ορισμός αυτός μπορεί να μετατραπεί ως εξής: σε ένα πρόβλημα βέλτιστης λήψης απόφασης καλούμαστε να βρούμε τις τιμές εκείνες των μεταβλητών απόφασης x_i που βελτιστοποιούν την αντικειμενική συνάρτηση $f(x_i)$.

Όταν οι μεταβλητές απόφασης παίρνουν πραγματικές τιμές:

$$\vec{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d \quad (3.1.2)$$

το πρόβλημα ονομάζεται συνεχούς βελτιστοποίησης (continuous optimization), ενώ αντίστοιχα, αν οι μεταβλητές απόφασης λαμβάνουν διακριτές (ακέραιες) τιμές:

$$\vec{x} = (x_1, x_2, \dots, x_d) \in \mathbb{Z}^d \quad (3.1.3)$$

αναφερόμαστε στο πρόβλημα ως διακριτής βελτιστοποίησης (discrete optimization).

Ανάλογα με το αν μεταβάλλεται η λύση στην πάροδο του χρόνου, μπορούμε να διακρίνουμε τα προβλήματα βελτιστοποίησης σε (α) στατικά και (β) δυναμικά. Στα στατικά προβλήματα υπολογίζεται η βέλτιστη λύση μια φορά, ενώ αντίθετα στα δυναμικά, η βέλτιστη λύση αλλάζει συνεχώς στην πάροδο του χρόνου. Αυτό τα καθιστά ιδιαίτερη πρόκληση για τους ερευνητές που ασχολούνται με αλγόριθμους βελτιστοποίησης.

3.1.1 Προβλήματα Βελτιστοποίησης υπό Περιορισμούς (Constrained Optimization Problems)

Συνήθως, τα προβλήματα που καλείται να επιλύσει κανείς στον πραγματικό κόσμο διέπονται από περιορισμούς. Για να εκφραστούν οι περιορισμοί αυτοί, πρέπει στην σχέση (3.1.1) να προστεθούν οι παρακάτω σχέσεις:

$$g_i(\vec{x}) \leq 0, \quad i = 1, 2, \dots, m \quad (3.1.4)$$

$$h_i(\vec{x}) = 0, \quad i = 1, 2, \dots, p \quad (3.1.5)$$

όπου, οι σχέσεις που συμβολίζονται με $g_i(\vec{x}) \leq 0$ λέγονται περιορισμοί ανισότητας (inequality constraints) και οι σχέσεις που συμβολίζονται ως $h_i(\vec{x}) = 0$ λέγονται περιορισμοί ισότητας (equality constraints).

3.1.2 Προβλήματα Βελτιστοποίησης Πολλαπλών Στόχων (Multi-objective Optimization Problems)

Σε πολλές περιπτώσεις, οι αποφάσεις που καλείται κανείς να λάβει σε πραγματικά προβλήματα μπορεί να έχουν δύο ή περισσότερους στόχους, οι οποίοι εκφράζονται μέσω διαφορετικών αντικειμενικών συναρτήσεων. Οπότε η σχέση (3.1.1) τροποποιείται κατάλληλα:

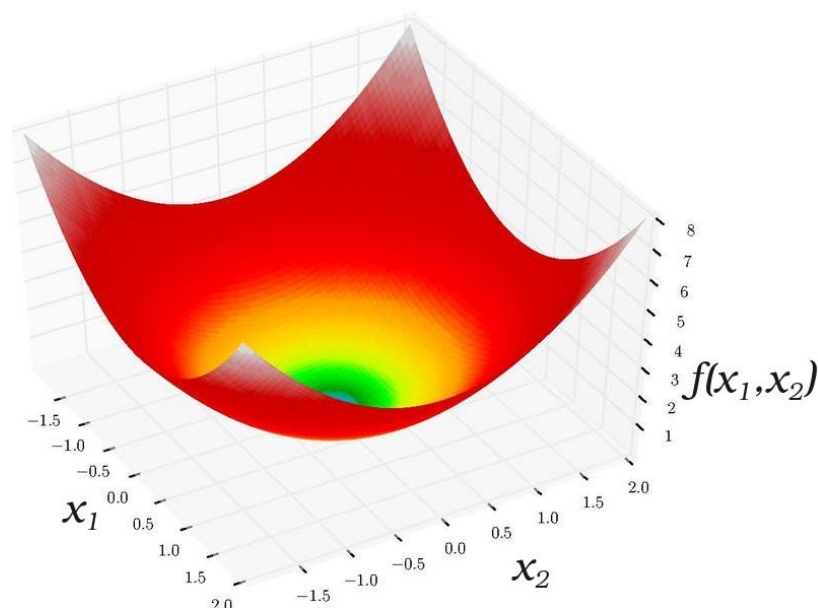
$$\text{βελτιστοποίηση της } f_i(\vec{x}), \quad i = \{1, 2, 3, \dots, n\} \quad (3.1.6)$$

όπου $f_i(\vec{x})$ συμβολίζονται οι αντικειμενικές συναρτήσεις που πρέπει να βελτιστοποιηθούν ως προς \vec{x} και n είναι ο αριθμός των αντικειμενικών συναρτήσεων που περιγράφουν το πρόβλημα. Το διάνυσμα των μεταβλητών απόφασης μεταβάλλεται σε $\vec{x} = (x_{i1}, x_{i2}, \dots, x_{id}) \in \mathbb{R}^d$, όπου x_{id} συμβολίζεται η d -οστή μεταβλητή απόφασης της i -οστής αντικειμενικής συνάρτησης και d είναι το πλήθος των μεταβλητών απόφασης.

3.1.3 Γιατί η βελτιστοποίηση είναι μια δύσκολη διαδικασία;

Η δυσκολία των προβλημάτων βελτιστοποίησης είναι ανάλογη του πλήθους των μεταβλητών απόφασης, αλλά κυρίως ανάλογη του χώρου στον οποίο αυτές ορίζονται. Για παράδειγμα, σε ένα πρόβλημα ελαχιστοποίησης με δύο μεταβλητές απόφασης $\vec{x} = (x_1, x_2)$, που ορίζονται για $x_1, x_2 \in [-2, 2]$, ο χώρος λύσης θα μπορούσε να μοιάζει με αυτόν της Εικόνας 3.1.

Ο εν λόγω χώρος λύσεων είναι ομαλός (smooth) και η καλύτερη λύση διακρίνεται εύκολα. Ωστόσο, τα πραγματικά προβλήματα δεν έχουν τόσο ευδιάκριτες λύσεις. Επιπλέον, ένα πρόβλημα βελτιστοποίησης σπάνια έχει μόνο δύο μεταβλητές απόφασης, ενώ συνήθως οι μεταβλητές απόφασης ορίζονται όλες για διαφορετικό εύρος τιμών (χώρος λύσεων). Κατ' επέκταση, ο χώρος λύσεων γίνεται πολυεπίπεδος και το εύρος κάθε επιπέδου είναι δυσανάλογο των τιμών εύρους των άλλων επιπέδων.



Εικόνα 3.1: Χώρος λύσεων της συνάρτησης σφαίρας (Sphere function) για δύο μεταβλητές απόφασης

Όσο μεγαλύτερο είναι το διάστημα στο οποίο ορίζεται κάθε μεταβλητή, τόσες περισσότεροι πιθανοί συνδυασμοί προκύπτουν. Ο έλεγχος όλων των πιθανών συνδυασμών (λύσεων) θα απαιτούσε τεράστια υπολογιστική ισχύ και θα διαρκούσε πολύ χρονικά. Αυτή η ανάγκη για υπολογιστικούς πόρους εκφράζεται μέσω της πολυπλοκότητας χρόνου και χώρου των μεθόδων που καλούνται να επιλύσουν προβλήματα βελτιστοποίησης.

3.2 Πολυπλοκότητα Χώρου και Χρόνου

Η σημαντικότητα της χρήσης ενός νοήμονος αλγορίθμου έγκειται στο γεγονός ότι μπορεί να εκτελέσει περισσότερες διεργασίες σε μια μονάδα χρόνου σε σύγκριση με τον ανθρώπινο εγκέφαλο. Για να αποτυπωθεί η πολυπλοκότητα των προβλημάτων, αλλά και ο χρόνος που απαιτείται για την εκτέλεση όλων των βημάτων ενός αλγορίθμου, στην Επιστήμη των Υπολογιστών έχουν καθιερωθεί κάποιες αλγεβρικές εκφράσεις για την πολυπλοκότητα του χρόνου (time complexity) και του χώρου (space complexity).

3.2.1 Πολυπλοκότητα Χρόνου

Οι Hartmanis and Stearns (1965) διερεύνησαν την υπολογιστική πολυπλοκότητα (computational complexity) των αλγορίθμων και μέσα από μια σειρά αποδείξεων κατέληξαν σε μια γενική έκφραση της πολυπλοκότητας συναρτήσει του χρόνου. Ωστόσο, ο Knuth (1976) έκανε ευρέως γνωστό τον μαθηματικό formalισμό που χρησιμοποιείται στην Επιστήμη των Υπολογιστών (Computer Science) για την πολυπλοκότητα μιας μεθόδου.

Μηχανή Turing ορίζουμε μια υποθετική συσκευή, η οποία χειρίζεται σύμβολα σύμφωνα με ένα σύνολο κανόνων και προσομοιώνει την λογική οποιουδήποτε αλγορίθμου, οπότε είναι χρήσιμη στο να εξηγεί τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας στο εσωτερικό ενός υπολογιστή.

Συμβολίζοντας M μια Μηχανή Turing (Turing Machine), ο χρόνος εκτέλεσης (running time) και η πολυπλοκότητα του χρόνου (time complexity) της M είναι μια συνάρτηση $f: \mathbb{N} \rightarrow \mathbb{N}$, όπου με $f(n)$ συμβολίζεται ο μέγιστος αριθμός βημάτων που εκτελεί η M για κάθε είσοδο (input) μήκους n (Sipser, 2006).

Ωστόσο, μια Μηχανή Turing είναι μια ντετερμινιστική μέθοδος, ενώ ένας Μετα-ευρετικός Αλγόριθμος Εμπνευσμένος από τη Φύση (Meta-heuristic Nature Inspired Algorithm) είναι μια στοχαστική διαδικασία, όπως έχει αναφερθεί και παραπάνω. Γι' αυτό ο ακριβής υπολογισμός του χρόνου εκτέλεσης ενός Μετα-ευρετικού Αλγόριθμου Εμπνευσμένου από τη Φύση δεν μπορεί παρά να εκτιμηθεί. Ένας τρόπος για την εκτίμηση του χρόνου εκτέλεσης ενός αλγορίθμου ονομάζεται Ασυμπτωτική Ανάλυση (asymptotic analysis), όπου η εκτίμηση του χρόνου εκτέλεσης γίνεται με βάση το χειρότερο σενάριο πολυπλοκότητας χρόνου (worst-case time complexity), το οποίο προέρχεται την μεγαλύτερη τάξη μεγέθους των εισόδων του αλγορίθμου.

Ας υποθέσουμε ότι ο παρακάτω ψευδοκώδικας αντιπροσωπεύει μια διεργασία που απαιτείται να εκτελεστεί εντός ενός αλγορίθμου:

Ψευδοκώδικας 3.1: Διεργασία με βρόγχο «Όσο»

1:	$i = 0$
2:	Όσο $i < n$
3:	κάποια εντολή
4:	$i = i + 1$
5:	Τέλος

Για να εκτιμήσουμε τον υπολογιστικό χρόνο της διεργασίας πρέπει να βρούμε τον υπολογιστικό χρόνο που απαιτεί κάθε βήμα. Ένα βήμα που δεν εξαρτάται από την είσοδο (input) κάποιων παραμέτρων ή δεδομένων, λέμε ότι εκτελείται σε σταθερό χρόνο (constant time) και συμβολίζεται $T(n) = 1$. Κατ' επέκταση, το βήμα 1 του παραπάνω ψευδοκώδικα θα έχει σταθερό χρόνο εκτέλεσης $T(n) = 1$. Η επαναληπτική διαδικασία «Όσο» θα εκτελεστεί όσες φορές ισχύει η συνθήκη $i < n$ και μια ακόμη φορά, όπου η συνθήκη θα παραβιάζεται και θα τερματιστεί η επαναληπτική διαδικασία. Είναι προφανές ότι αφού το i ξεκινάει από την τιμή 0 και αυξάνεται κατά ένα σε κάθε επανάληψη, η επαναληπτική διαδικασία (βρόγχος) θα εκτελεστεί n φορές και μια θα εκτελεστεί μόνο ο έλεγχος. Τα βήματα που εκτελούνται εντός του βρόγχου θα γίνουν όσες φορές εκτελείται ο βρόγχος, δηλαδή n φορές. Οπότε ο απαιτούμενος χρόνος ανά βήμα θα είναι (Πίνακας 3.1):

Πίνακας 3.1: Απαιτούμενος χρόνος εκτέλεσης κάθε βήματος του Ψευδοκώδικας 3.1

Βήμα	Απαιτούμενος χρόνος $T(n)$
1: $i = 0$	1
2: Όσο $i < n$	n
3: κάποια εντολή	n
4: $i = i + 1$	n
5: Τέλος	1

Οπότε, η εξίσωση που περιγράφει τα βήματα που απαιτούνται για την ολοκλήρωση των βημάτων του Ψευδοκώδικας 3.1 δίνεται από την σχέση:

$$T(n) = 3n + 2 \quad (3.2.1)$$

Η αλγεβρική σχέση της πολυπλοκότητας του χρόνου, βάσει της Ασυμπτωτικής Ανάλυσης, συμβολίζοντας με \mathbb{R}^+ το σύνολο τον μη αρνητικών πραγματικών αριθμών και με f, T συναρτήσεις που $f, T: \mathbb{N} \rightarrow \mathbb{R}^+$, είναι:

$$f(n) = O(T(n)) \quad (3.2.2)$$

όπου $T(n)$ συμβολίζουμε το ασυμπτωτικό άνω όριο του αριθμού των βημάτων $f(n)$. Η σχέση (3.2.2) εκφράζει το γεγονός ότι για κάθε πραγματικό αριθμό $c > 0$, υπάρχει ένας αριθμός n_0 , που ισχύει $f(n) \leq c \cdot T(n)$ για όλα τα $n \geq n_0$. Η σχέση (3.2.2) αποκαλείται ασυμπτωτικός συμβολισμός (asymptotic notation) ή συμβολισμός του Μεγάλου-Ο (Big-O notation) και σημαίνει ότι χρειάζονται βήματα λιγότερα ή ίσα με $T(n)$ για την περάτωση μιας διαδικασίας. Ο συμβολισμός του Μεγάλου-Ο πρωτοεμφανίστηκε από τον Bachmann (1894) και έγινε πιο γνωστός μέσα από το βιβλίο του Landau (2000).

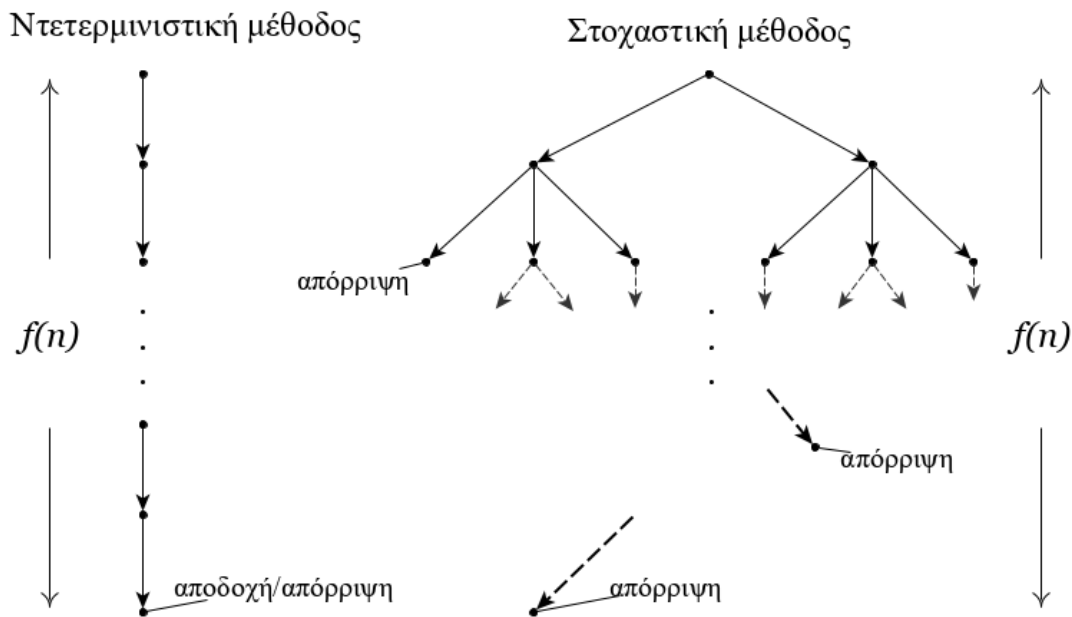
Πίνακας 3.2: Γνωστές πολυπλοκότητες χρόνου (time complexities)

Συμβολισμός του Μεγάλου-Ο	Όνομα
$O(1)$	Σταθερός χρόνος
$O(\log n)$	Λογαριθμικός χρόνος
$O(n)$	Γραμμικός χρόνος
$O(n^2)$	Τετραγωνικός χρόνος
$O(n^k)$	Πολυωνυμικός χρόνος
$O(2^n)$	Εκθετικός χρόνος

Βάσει του ορισμού της Ασυμπτωτικής Ανάλυσης (asymptotic analysis) που δόθηκε παραπάνω, το χειρότερο σενάριο πολυπλοκότητας χρόνου ή καλύτερα η μεγαλύτερη τάξη μεγέθους εισόδων είναι ο όρος $3n$ στην σχέση (3.2.1). Μη λαμβάνοντας υπόψη τον σταθερό παράγοντα 3, λέμε ότι η T είναι ασυμπτωτική κατά

n και συμβολίζεται $f(n) = O(n)$, σύμφωνα με την σχέση (3.2.2). Στον Πίνακα 3.2 παραπάνω παρουσιάζονται οι πιο γνωστές πολυπλοκότητες χρόνου.

Ο χρόνος ολοκλήρωσης μιας ντετερμινιστικής μεθόδου θα μπορούσε να προσφέρει μια καλύτερη βάση σύγκρισης για την πολυπλοκότητα χρόνου των μεθόδων που είναι Εμπνευσμένες από τη Φύση. Οι ντετερμινιστικές μέθοδοι, όπως αναφέρθηκε και παραπάνω, δίνουν το ίδιο αποτέλεσμα κάθε φορά που εκκινούν με τις ίδιες αρχικές συνθήκες. Οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση είναι στοχαστικές μέθοδοι, ένα χαρακτηριστικό στο οποίο βασίζεται η χρησιμότητά τους, καθώς με ίδιες αρχικές συνθήκες τα αποτελέσματα δεν είναι ίδια κάθε φορά που εκτελούνται οι αλγόριθμοι.



Σχήμα 3-1: Διαγραμματική απεικόνιση ντετερμινιστικών και στοχαστικών μεθόδων

Σύμφωνα με το θεώρημα 7.11 που αναπτύσσει ο Sipser (2006, p.256), αν θεωρήσουμε μια $T(n)$ συνάρτηση πολυπλοκότητας χρόνου, όπου ισχύει $T(n) \geq n$, τότε κάθε στοχαστική μέθοδος χρόνου $T(n)$ ισοδυναμεί με μια ντετερμινιστική μέθοδο χρόνου $2^{O(T(n))}$. Όπως φαίνεται και στο Σχήμα 3-1, μια στοχαστική μέθοδος θα εκτελέσει πολύ περισσότερες κατευθύνσεις αναζήτησης και κατ' επέκταση θα χρειαστεί πολύ περισσότερο χρόνο. Ωστόσο, η ποικιλία των αποτελεσμάτων που προσφέρει μια στοχαστική μέθοδος είναι αυτή που την κάνει αποτελεσματικότερη σε σύγκριση με μια ντετερμινιστική.

Το πλήθος όλων των πιθανών συνδυασμών ενός προβλήματος ορίζει και τη δυσκολία του εν λόγω προβλήματος. Όσο μεγαλύτερο είναι αυτό το πλήθος, τόσο πιο πολύ χρόνο χρειάζεται για να ερευνηθούν όλες οι πιθανές λύσεις και να βρεθεί η καλύτερη όλων. Για να ορισθεί η δυσκολία των προβλημάτων ως προς τον

απαιτούμενο χρόνο εύρεσης της καλύτερης λύσης έχουν χωριστεί τα προβλήματα σε μερικές κλάσεις.

3.2.2 Κλάση P

Θέλοντας να συγκρίνουμε δύο μεθόδους A και B, μια πολυωνυμικού και μια εκθετικού χρόνου αντίστοιχα, υποθέτουμε ότι και οι δύο εκτελούν μια διαδικασία $n = 100$ φορές, ενώ η μέθοδος πολυωνυμικού χρόνου A έχει $k = 3$ εμφολευμένες τέτοιες επαναληπτικές διαδικασίες. Σύμφωνα με την ασυμπτωτική ανάλυση, ο χρόνος εκτέλεσης της A μεθόδου είναι:

$$f_A(n) = O(n^k) = O(100^3) = O(10^6) \quad (3.2.3)$$

ενώ ο αντίστοιχος χρόνος της μεθόδου B υπολογίζεται:

$$f_B(n) = O(2^n) = O(2^{100}) = O(1.2676506 \times 10^{30}) \quad (3.2.4)$$

Είναι σαφές από τις σχέσεις (3.2.3) και (3.2.4) ότι η μέθοδος A είναι γρηγορότερη χρονικά από την B. Γενικότερα, υπολογιστικές μέθοδοι εκθετικού χρόνου δεν προσφέρουν κάποια χρησιμότητα στην επίλυση προβλημάτων, παρά μόνο στην περίπτωση της εξαντλητικής αναζήτησης (exhaustive search).

Γνωρίζοντας ότι οι ντετερμινιστικές μέθοδοι εκτελούνται σε πολυωνυμικό χρόνο (Sipser, 2006), στόχος είναι να αναπτυχθεί μια στοχαστική μέθοδος που να είναι καλύτερη σε απόδοση από την αντίστοιχη ντετερμινιστική, αλλά να εκτελείται σε πολυωνυμικό χρόνο. Για την ανάπτυξη αποδοτικών μεθόδων, ωστόσο, χρειάζεται να μελετηθεί πρώτα το πρόβλημα που καλούμαστε να επιλύσουμε. Η δυσκολία του προβλήματος (π.χ. εύρος χώρου λύσεων) καθορίζει και την πολυπλοκότητα των μεθόδων που απαιτούνται για να το επιλύσουν.

Ορίζουμε P την κλάση των προβλημάτων που μπορούν να επιλυθούν σε πολυωνυμικό χρόνο από μια ντετερμινιστική μέθοδο και συμβολίζουμε:

$$P = \bigcup_k \text{TIME}(n^k) \quad (3.2.5)$$

Η κλάση P περιέχει (πρακτικά) όλα τα προβλήματα που είναι επιλύσιμα από υπολογιστές. Εφόσον υπάρχει μια μέθοδος πολυωνυμικού χρόνου που επιλύει ένα πρόβλημα κλάσης P, οποιαδήποτε χρονικά πιο αργή μέθοδος θεωρείται λιγότερο αποδοτική, εφόσον δυνητικά μπορούν όλες οι μέθοδοι να βρουν την ίδια λύση. Κατ' επέκταση, για τα προβλήματα κλάσης P δεν είναι απαραίτητη η ανάπτυξη στοχαστικών μεθόδων, καθώς επιλύονται με ακρίβεια από ντετερμινιστικές μεθόδους.

Παραδείγματα προβλημάτων κλάσης P (S. Sahni and T. Gonzales, 1974) είναι τα προβλήματα γραμμικού προγραμματισμού, αλλά και αρκετά προβλήματα γράφων (graphs). Παραδείγματος χάριν, πρόβλημα κλάσης P θεωρείται η εύρεση ενός μονοπατιού (path) που θα ξεκινάει από τον αρχικό κόμβο a και θα καταλήγει στον

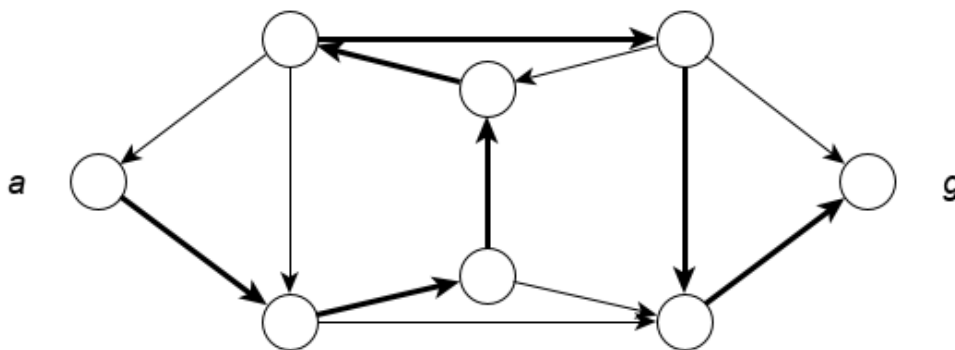
τελικό κόμβο g ενός κατευθυνόμενου (directed) γράφου $\langle G, a, g \rangle$. Ο παρακάτω Ψευδοκώδικας 3.2 περιγράφει τη βηματική διαδικασία επίλυσης του προβλήματος.

Το πρώτο βήμα εκτελείται μια φορά, άρα συμβολίζεται $O(1)$ η πολυπλοκότητά του. Ο βρόχος «Όσο» θα εκτελεστεί n φορές μέχρι να ολοκληρωθεί ένα μονοπάτι, ενώ ο τελικός έλεγχος του μονοπατιού (βήματα 8 έως 12) θα εκτελεστεί 1 φορά. Οπότε η διαδικασία που περιγράφεται στον Ψευδοκώδικας 3.2 είναι πολυπλοκότητας $O(n)$, σύμφωνα με την Ασυμπτωτική Ανάλυση. Δηλαδή είναι μια διαδικασία που επιλύει το πρόβλημα σε πολυωνυμικό χρόνο.

Ψευδοκώδικας 3.2: Αλγόριθμος διερεύνησης ύπαρξης μονοπατιού σύνδεσης αρχικού και τελικού κόμβου σε έναν γράφο $\langle G, a, g \rangle$

- 1: Σημείωσε τον κόμβο a
- 2: **Όσο** σημειώνονται νέοι κόμβοι
- 3: Δες όλες τις διαθέσιμες διαδρομές του γράφου
- 4: **Εάν** μια διαδρομή (b,c) ξεκινάει από έναν σημειωμένο κόμβο b και καταλήγει σε έναν κόμβο c που δεν έχει σημειωθεί
- 5: Σημείωσε τον κόμβο c
- 6: **Τέλος_εάν**
- 7: **Τέλος**
- 8: **Εάν** έχει σημειωθεί ο κόμβος g
- 9: αποδοχή
- 10: **Αλλιώς**
- 11: απόρριψη
- 12: **Τέλος_εάν**

3.2.3 Κλάση NP



Σχήμα 3-2: Γράφος Προβλήματος Χαμιλτόνιου Μονοπατιού

Απλές διεργασίες, όπως η εύρεση του μεγαλύτερου κοινού διαιρέτη για δύο αριθμούς και η διερεύνηση αν ένας αριθμός είναι πρώτος, είναι εύκολο να επιλυθούν γρήγορα ακολουθώντας μια απλή βηματική διαδικασία. Προβλήματα, όμως, τα οποία

έχουν πολλές πιθανές λύσεις και όσο αυξάνονται οι διαστάσεις του προβλήματος, οι συνδυασμοί αυτών συνθέτουν έναν τεράστιο χώρο λύσεων, ο οποίος δεν μπορεί να ερευνηθεί σε πολυωνυμικό χρόνο με σκοπό να βρεθεί ο καλύτερος μεταξύ των πιθανών συνδυασμών, είναι αδύνατο να προσεγγιστούν επιτυχώς από ντετερμινιστικές μεθόδους.

Για παράδειγμα, έστω ότι διερευνάται αν ο γράφος στο Σχήμα 3-2 έχει κάποιο Χαμιλτόνιο Μονοπάτι (Hamiltonian Path). Το Χαμιλτόνιο Μονοπάτι είναι η διαδρομή, η οποία περνάει από κάθε κόμβο του γράφου μόνο μια φορά. Για τον έλεγχο Χαμιλτόνιου Μονοπατιού, θα έπρεπε να προστεθεί ένας σχετικός έλεγχος στην αλγοριθμική διαδικασία που περιγράφεται στον Ψευδοκώδικας 3.2, όπως φαίνεται στον Ψευδοκώδικας 3.3 παρακάτω.

Ψευδοκώδικας 3.3: Αλγόριθμος διερεύνησης Χαμιλτόνιου Μονοπατιού

- 1: Σημείωσε τον κόμβο a
 - 2: **Όσο** σημειώνονται νέοι κόμβοι
 - 3: Δες όλες τις διαθέσιμες διαδρομές του γράφου
 - 4: **Εάν** μια διαδρομή (b,c) ξεκινάει από έναν σημειωμένο κόμβο b και καταλήγει σε έναν κόμβο c που δεν έχει σημειωθεί
 - 5: Σημείωσε τον κόμβο c
 - 6: **Τέλος_εάν**
 - 7: **Τέλος**
 - 8: **Εάν** έχει σημειωθεί ο κόμβος g
 - 9: **Εάν** σχηματίζεται Χαμιλτόνιο Μονοπάτι
 - 10: αποδοχή
 - 11: **Αλλιώς**
 - 12: απόρριψη
 - 13: **Τέλος_εάν**
 - 14: **Αλλιώς**
 - 15: απόρριψη
 - 16: **Τέλος_εάν**
-

Ο εν λόγω ψευδοκώδικας διερευνά αν το μονοπάτι που ελέγχεται είναι Χαμιλτόνιο ή όχι. Για να διερευνηθεί αν ο γράφος έχει Χαμιλτόνιο Μονοπάτι, θα έπρεπε να επαναληφθούν τα βήματα 1 έως 7 μέχρι να βρεθεί ένα Μονοπάτι που να ικανοποιεί την συνθήκη που περιγράφεται παραπάνω. Η αλγοριθμική διαδικασία που θα επέλυε αυτό το πρόβλημα είναι άγνωστο σε πόσο χρόνο θα μπορούσε να ολοκληρωθεί, καθώς ο χρόνος αυτός εξαρτάται από το πλήθος των πιθανών μονοπατιών του γράφου που πρέπει να διερευνηθούν.

Ωστόσο, είναι εφικτό να ελεγχθεί αν υπάρχει κάποιο Χαμιλτόνιο Μονοπάτι σε πολυωνυμικό χρόνο. Μια στοχαστική μέθοδος, για παράδειγμα, θα μπορούσε να

ελέγξει διάφορα μονοπάτια σε πολυωνυμικό χρόνο. Κατ' επέκταση, λέμε ότι το εν λόγω πρόβλημα έχει πολυωνυμική επαληθευσιμότητα (polynomial verifiability).

Οπότε, ορίζουμε NP την κλάση των προβλημάτων που υπάρχουν μη-ντετερμινιστικές μέθοδοι, οι οποίες μπορούν να επαληθεύσουν τα προβλήματα NP κλάσης σε πολυωνυμικό χρόνο. Με άλλα λόγια, αυτή η κλάση προβλημάτων παίρνει το όνομά της από τον όρο non-deterministic polynomial που αναφέρεται στις μη-ντετερμινιστικές πολυωνυμικού χρόνου μεθόδους που χρειάζονται και συμβολίζουμε:

$$NP = \bigcup_k \text{NTIME}(n^k) \quad (3.2.6)$$

Στη συγκεκριμένη κλάση ταξινομούνται τα περισσότερα γνωστά προβλήματα, όπως το Πρόβλημα του Πλανόδιου Πωλητή, τα Προβλήματα Δρομολόγησης Οχημάτων και αρκετά ακόμη κλασικά προβλήματα επιχειρησιακής έρευνας.

3.2.4 Κλάση NP-πληρότητας

Ένα ερώτημα που απασχόλησε την Επιστήμη των Υπολογιστών είναι αν οι κλάσεις P και NP συμπίπτουν, δηλαδή αν τα προβλήματα που μπορούν να επαληθευτούν σε πολυωνυμικό χρόνο, μπορούν να επιλυθούν εξίσου σε πολυωνυμικό χρόνο. Για τον σκοπό αυτό έχει προσδιοριστεί η έννοια της NP-πληρότητας (NP-completeness), όπου σε αυτή την κλάση ανήκουν προβλήματα τα οποία μπορούν να υποβιβαστούν σε ένα οποιοδήποτε άλλο πρόβλημα κλάσης NP σε πολυωνυμικό χρόνο, και των οποίων η λύση εξακολουθεί να μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο. Η παραπάνω παρατήρηση διατυπώθηκε από τους Stephen Cook (Cook, 1971) και Leonid Levin (Levin, 1973) και είναι γνωστή ως θεώρημα Cook-Levin.

Το πρόβλημα Χαμιλτόνιου Μονοπατιού που διατυπώνεται παραπάνω είναι ένα NP-πλήρες πρόβλημα, για παράδειγμα. Άλλα παραδείγματα προβλημάτων αυτής της κλάσης είναι το Πρόβλημα του Σακιδίου (Knapsack Problem), διάφορες εκδοχές προβλημάτων χρονοδρομολόγησης (scheduling), καθώς και οι απλές μορφές των προβλημάτων του Πλανόδιου Πωλητή και της Δρομολόγησης Οχημάτων.

Στη βιβλιογραφία έχουν διατυπωθεί αρκετές ακόμη κλάσεις πολυπλοκότητας προβλημάτων, όπως τα NP-δυσμενή (NP-hard) προβλήματα, τα οποία δεν επαληθεύονται σε πολυωνυμικό χρόνο. Γενικότερα, όλα τα προβλήματα NP κλάσεως ή όσων κλάσεων συνεπάγονται από αυτή, αν προσεγγιστούν από ντετερμινιστικές μεθόδους, μπορούν να επιλυθούν σε εκθετικό χρόνο:

$$NP \subseteq \text{EXPTIME} = \bigcup_k \text{NTIME}(2^{n^k}) \quad (3.2.7)$$

Η σύγκριση των σχέσεων (3.2.6) και (3.2.7) δικαιολογεί την ανάγκη ανάπτυξης στοχαστικών (μη-ντετερμινιστικών) μεθόδων για την επίλυση των προαναφερόμενων προβλημάτων.

3.2.5 Πολυπλοκότητα Χώρου

Με τον ίδιο τρόπο που ορίστηκε η πολυπλοκότητα του χρόνου, συμβολίζοντας M μια Μηχανή Turing (Turing Machine), η πολυπλοκότητα του χώρου (space complexity) της M είναι μια συνάρτηση $f: \mathbb{N} \rightarrow \mathbb{N}$, όπου με $f(n)$ συμβολίζεται ο μέγιστος χώρος (μνήμη) που χρειάζεται η M για κάθε είσοδο (input) μήκους n (Sipser, 2006).

Συμβολίζοντας με $\text{DSPACE}(f(n))$ την πολυπλοκότητα χώρου (την απαιτούμενη μνήμη) που χρειάζεται μια ντετερμινιστική μέθοδος, με $\text{NSPACE}(f(n))$ συμβολίζεται η αντίστοιχη πολυπλοκότητα χώρου μιας μη-ντετερμινιστικής μεθόδου, όπου βάσει του θεωρήματος του Savitch (1970) ισχύει:

$$\text{DSPACE}(f(n)) \subseteq \text{NSPACE}(f(n)) \subseteq \text{DSPACE}(f^2(n)) \quad (3.2.8)$$

που πρακτικά εκφράζει το γεγονός ότι κάθε μη-ντετερμινιστική μέθοδος που απαιτεί χώρο $f(n)$, μπορεί να αντικατασταθεί από μια ντετερμινιστική μέθοδο που απαιτεί χώρο $f^2(n)$. Κατ' επέκταση, η χρήση των (στοχαστικών) Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση συνεπάγεται αισθητά λιγότερες ανάγκες στην απαιτούμενη μνήμη για την επίλυση ενός προβλήματος, σε αντίθεση με μια ντετερμινιστική μέθοδο εξαντλητικής αναζήτησης.

Αντίστοιχα με τις κλάσεις που ορίστηκαν στην πολυπλοκότητα του χρόνου, ορίζονται κι εδώ:

$$\text{PSPACE} = \bigcup_k \text{DSPACE}(n^k) \quad (3.2.9)$$

όπου PSPACE είναι η κλάση προβλημάτων που επιλύονται με τη χρήση πολυωνυμικού χώρου από ντετερμινιστικές μεθόδους και:

$$\text{NPSPACE} = \bigcup_k \text{NSPACE}(n^k) \quad (3.2.10)$$

όπου NPSPACE συμβολίζει την κλάση προβλημάτων που επιλύονται με τη χρήση πολυωνυμικού χώρου από μη-ντετερμινιστικές μεθόδους.

Για να συσχετίσουμε τις πολυπλοκότητες χώρου και χρόνου, παρατηρούμε ότι $P \subseteq \text{PSPACE}$ λόγω του ότι μια γρήγορη μέθοδος προϋποθέτει να μην χρησιμοποιεί πολύ χώρο (πολλή μνήμη). Αντίστοιχα, εφόσον $f(n) \geq n$, ισχύει και $\text{NP} \subseteq \text{NPSPACE}$, αλλά και $\text{NP} \subseteq \text{PSPACE}$, καθώς γενικά μια μέθοδος (ντετερμινιστική ή μη) που ολοκληρώνεται σε $f(n)$ χρόνο, μπορεί να διεκπαιρώνει μια διεργασία σε κάθε στάδιο της βηματικής λειτουργίας της, οπότε μπορεί να χρησιμοποιήσει το πολύ $f(n)$ χώρο.

Εφόσον ισχύει $f(n) \geq n$, μια μέθοδος (ντετερμινιστική ή μη) εκτελεί το πολύ $f(n)2^{O(f(n))}$ διεργασίες (Sipser, 2006, p.194). Οπότε, αφού μια μέθοδος που

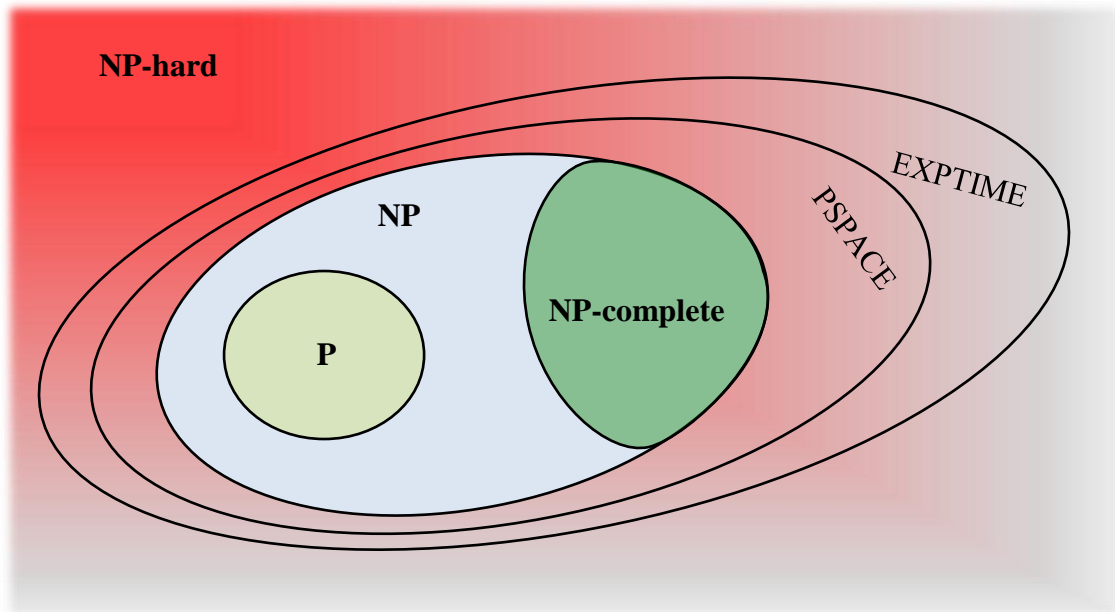
ολοκληρώνεται σε $f(n)$ χρόνο εκτελεί το πολύ $f(n)2^{O(f(n))}$ διεργασίες, συνεπάγεται ότι:

$$\text{PSPACE} \subseteq \text{EXPTIME} = \bigcup_k \text{NTIME}(2^{n^k}) \quad (3.2.11)$$

Συνοψίζοντας τα παραπάνω, οι μεταξύ των κλάσεων πολυπλοκότητας χρόνου και χώρου ισχύει:

$$\text{P} \subseteq \text{NP} \subseteq \text{PSPACE} = \text{NPSpace} \subseteq \text{EXPTIME} \quad (3.2.12)$$

Η σχέση (3.2.12) απεικονίζεται στο Σχήμα 3-3, παρακάτω.



Σχήμα 3-3: Σχέση μεταξύ των κλάσεων πολυπλοκότητας χρόνου και πολυπλοκότητας χώρου

3.2.6 Πολυπλοκότητα προβλημάτων βελτιστοποίησης

Γενικά, η πολυπλοκότητα των προβλημάτων, όπως διατυπώνεται στις προηγούμενες υποενότητες, αναφέρεται σε προβλήματα όπου οι εισοδοί (inputs) που θα δοθούν στη μέθοδο που θα κληθεί να τα λύσει είναι πεπερασμένες και συχνά διακριτές. Δηλαδή, είναι δυαδικής μορφής (0 ή 1). Για παράδειγμα, στο πρόβλημα του Σακιδίου (Knapsack Problem) η λύση μπορεί να αναπαριστάται με 0 αν ένα αντικείμενο δεν επιλέγεται για το σακίδιο και 1 στην αντίθετη περίπτωση.

Τα προβλήματα βελτιστοποίησης που μελετήθηκαν και επιλύθηκαν στην παρούσα διατριβή δεν έχουν εισόδους αυτής της μορφής. Για παράδειγμα, τα δεδομένα του προβλήματος της Βελτιστοποίησης Χαρτοφυλακίου αποτελούν τιμές αποδόσεων για συγκεκριμένες τιμές των μετοχών που συνιστούν έναν χρηματιστηριακό δείκτη. Αν αυτές οι αποδόσεις αλλάξουν ή κάποια μετοχή βρεθεί εκτός του δείκτη, αυτομάτως οι εισοδοί του προβλήματος αποκτούν διαφορετική

μορφή. Οπότε, το Σχήμα 3-3, καθώς και οι προαναφερόμενες κλάσεις, δεν είναι απόλυτα ακριβή για τα εν λόγω προβλήματα. Ωστόσο, εφόσον η Επιστήμη των Μαθηματικών θεωρεί τα προαναφερόμενα προβλήματα δύσκολα επιλύσιμα και υποθέτει ότι ανήκουν στην NP-δυσμενή κλάση, αυτή είναι και η υπόθεση εργασίας που γίνεται στην παρούσα διατριβή.

Επιπρόσθετα, πρέπει να αναφερθεί πως η πολυπλοκότητα του προβλήματος εξαρτάται από τον καλύτερο αλγόριθμο που μπορεί να το λύσει. Αν, για παράδειγμα, υπάρχει αλγόριθμος που μπορεί να λύσει ένα πρόβλημα σε πολυωνυμικό χρόνο, αυτό το πρόβλημα θεωρείται NP κλάσης. Ωστόσο, μπορεί άλλοι αλγόριθμοι να χρειάζονται εκθετικό χρόνο. Αυτό το γεγονός δεν εντάσσει το πρόβλημα σε δυσκολότερη κλάση. Ο καλύτερος αλγόριθμος, παρόλα αυτά, μπορεί να αποδεικνύεται από την Επιστήμη των Μαθηματικών ότι υπάρχει, αλλά να μην έχει αναπτυχθεί από την Επιστήμη Υπολογιστών ακόμη.

Εν κατακλείδι, τα προβλήματα που μελετήθηκαν και επιλύθηκαν στην παρούσα διατριβή δεν υπάρχει τρόπος να γνωρίζουμε αν μπορούν να υποβαθμιστούν σε απλούστερη κλάση πολυπλοκότητας, δηλαδή αν υπάρχει αλγόριθμος που τα επιλύει και απαιτεί πολυωνυμικό χρόνο για παράδειγμα. Αυτή η απόδειξη θα συνιστούσε από μόνη της σημαντική συνεισφορά στο πεδίο των μαθηματικών. Ωστόσο, μπορούμε να θεωρήσουμε ότι ανήκουν στην NP-δυσμενή κλάση και να κάνουμε υποθέσεις για το σε ποια κλάση θα μπορούσαν να υποβαθμιστούν σύμφωνα με τον χώρο που απαιτούν ή τον χρόνο επίλυσης που θα απαιτούσε ένας καλός αλγόριθμος.

3.3 Εξελικτικοί και Νοήμονες Μετα-Ευρετικοί Αλγόριθμοι

Αρκετές μέθοδοι έχουν προταθεί στη βιβλιογραφία για την επίλυση προβλημάτων βελτιστοποίησης. Αρχικά, υπάρχουν οι ακριβείς αλγόριθμοι (exact algorithms) που βρίσκουν την ακριβή λύση ενός προβλήματος. Ωστόσο, όσο αυξάνεται η πολυπλοκότητα του προβλήματος οι ακριβείς αλγόριθμοι δεν είναι ικανοί να εντοπίσουν την ακριβή λύση ενός προβλήματος σε πεπερασμένο χρόνο.

Για τον σκοπό αυτό προτάθηκαν ευρετικοί (heuristic) αλγόριθμοι, οι οποίοι μέσω εμπειρικών ή ευρετικών κανόνων είναι σε θέση να αναζητήσουν τη λύση του προβλήματος ανάμεσα σε λιγότερες πιθανές λύσεις από τον χώρο λύσεων. Η αδυναμία των ευρετικών τεχνικών να πετύχουν λύσεις υψηλής ποιότητας (high quality solutions) γέννησε τους μετα-ευρετικούς αλγόριθμους (metaheuristic algorithms), οι οποίοι μεταβάλλουν τις λύσεις στην πορεία της αλγοριθμικής διαδικασίας βασιζόμενοι σε κάποια πληροφορία ή σε ένα σύνολο κανόνων.

Με τον καιρό, αυτή η πληροφορία ή ακόμη και το σύνολο των κανόνων άρχισε να δημιουργείται μέσω νοημοσύνης (intelligence). Η νοημοσύνη στους Νοήμονες Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση προέρχεται από το φαινόμενο ή τη συμπεριφορά που συναντάται στη φύση και μοντελοποιεί ο αλγόριθμος.

Η αρχή έγινε με τους εξελικτικούς αλγόριθμους (evolutionary algorithms), οι οποίοι εξελίσσουν τις λύσεις στη διάρκεια της διαδικασίας τους με παρόμοιο τρόπο με αυτόν που εξελίσσονται τα έμβια όντα (διασταύρωση γονέων και τυχαίες μεταλλάξεις σε ένα μικρό ποσοστό του πληθυσμού). Με έναυσμα την Προσομοιωμένη Ανόπτηση (Simulated Annealing) άρχισαν να γεννώνται τεχνικές που αντλούν τη νοημοσύνη που αναφέρεται παραπάνω από άλλες παρατηρήσεις στη φύση.

Οι περισσότερες ευρετικές τεχνικές είναι ντετερμινιστικές μέθοδοι, δηλαδή με τις ίδιες αρχικές συνθήκες καταλήγουν στο ίδιο αποτέλεσμα. Ο στοχαστικός χαρακτήρας των Αλγόριθμων που είναι Εμπνευσμένοι από τη Φύση προσφέρει τη δυνατότητα εξερεύνησης μεγαλύτερου μέρους του χώρου λύσεων, καθότι με τις ίδιες αρχικές συνθήκες το αποτέλεσμα μπορεί να διαφέρει κατά πολύ κάθε φορά που τρέχει ένας Νοήμων Αλγόριθμος Εμπνευσμένος από τη Φύση.

3.3.1 Φυσικό Ανάλογο

Οι Εξελικτικοί (και ειδικότερα οι Εμπνευσμένοι από τη Φύση) Αλγόριθμοι αντλούν την έμπνευση τους από κάτι που παρατηρείται στη φύση, όπως έχει προαναφερθεί. Η ιδέα (concept) στην οποία στηρίζεται το αλγοριθμικό μοντέλο ενός Εξελικτικού Αλγορίθμου ή ενός Αλγορίθμου Εμπνευσμένου από τη Φύση αποτελεί το Φυσικό Ανάλογο του αλγορίθμου.

3.3.2 Πληθυσμός και Γενιές

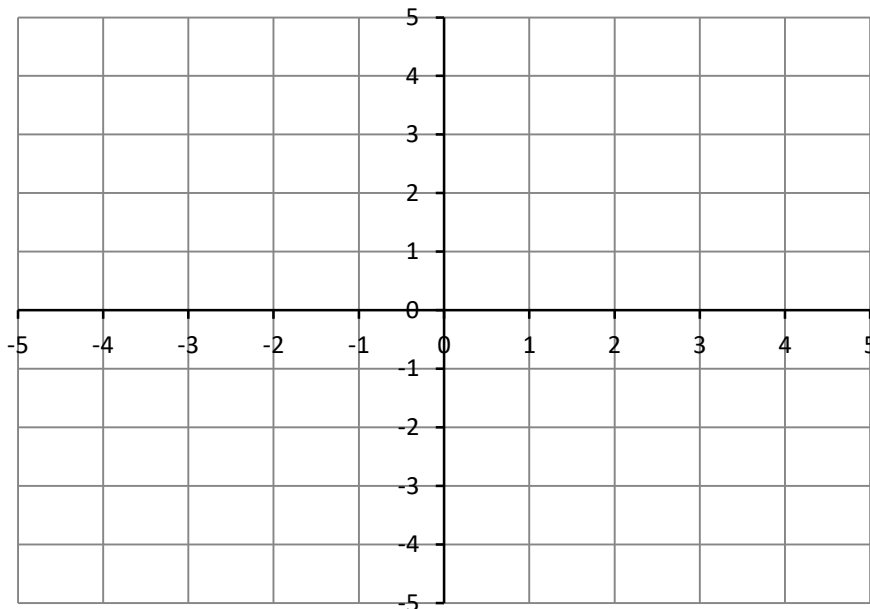
Στους Εξελικτικούς Αλγόριθμους (Evolutionary Algorithms) σημαντικό ρόλο παίζουν δύο παράμετροι: ο πληθυσμός (population) και οι γενιές (generations). Ο πληθυσμός είναι το σύνολο των υποψήφιων λύσεων (candidate solutions) σε μια δεδομένη στιγμή που εκτελείται ο αλγόριθμος. Οι γενιές συμβολίζουν τις επαναλήψεις που θα κάνει ο αλγόριθμος μέχρι τον τερματισμό του.

Στους Γενετικούς Αλγόριθμους το πλήθος των λύσεων μπορεί να μεταβάλλεται. Δηλαδή, παρατηρείται σε κάποιες εκδοχές του Γενετικού Αλγορίθμου να μειώνεται ο αριθμός των λύσεων ύστερα από κάποιες γενιές. Ωστόσο, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση διατηρούν σταθερό τον αριθμό του πληθυσμού των αναζητητών.

3.4 Εξερεύνηση και Εκμετάλλευση

Πολύ συχνά στη βιβλιογραφία συναντώνται οι όροι exploration και exploitation, οι οποίοι αναφέρονται στην εξερεύνηση του χώρου λύσεων και στην τοπική αναζήτηση γύρω από καλές λύσεις αντίστοιχα. Στην μετάφραση του βιβλίου των Christian and Griffiths (2016) χρησιμοποιούνται οι ακριβείς μεταφράσεις «εξερεύνηση» (exploration) και «εκμετάλλευση» (exploitation), μια ορολογία που υιοθετείται στην παρούσα διατριβή.

Η εξερεύνηση (exploration) περιγράφει την ανάγκη κάλυψης όσο το δυνατόν μεγαλύτερου χώρου από τον χώρο λύσεων κατά την διαδικασία αναζήτησης λύσης. Αν, για παράδειγμα, ο χώρος λύσεων ενός προβλήματος αποτυπώνεται από το παρακάτω καρτεσιανό επίπεδο (Εικόνα 3.2):



Εικόνα 3.2: Παράδειγμα χώρου λύσεων

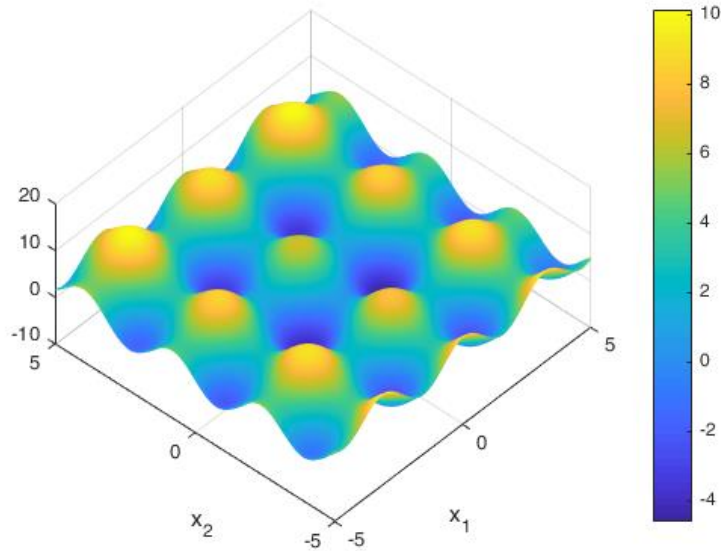
ενός προβλήματος με δύο διαστάσεις και άρα δύο μεταβλητές απόφασης), οι οποίες ορίζονται μόνο για ακέραιες τιμές ($x_i \in \mathbb{Z}$), τότε ιδανικά για να εξερευνήσουμε όλους τους πιθανούς συνδυασμούς των x_i θα έπρεπε να ελέγξουμε 121 ζεύγη λύσεων της μορφής (x_1, x_2) . Ο αριθμός αυτός προκύπτει από τον υπολογισμό του πλήθους των πιθανών λύσεων ps :

$$ps = k_1 \times k_2$$

όπου με k_1 συμβολίζεται το πλήθος των διαφορετικών τιμών που μπορεί να λάβει η μεταβλητή απόφασης x_1 και με k_2 συμβολίζεται το αντίστοιχο πλήθος για την μεταβλητή απόφασης x_2 .

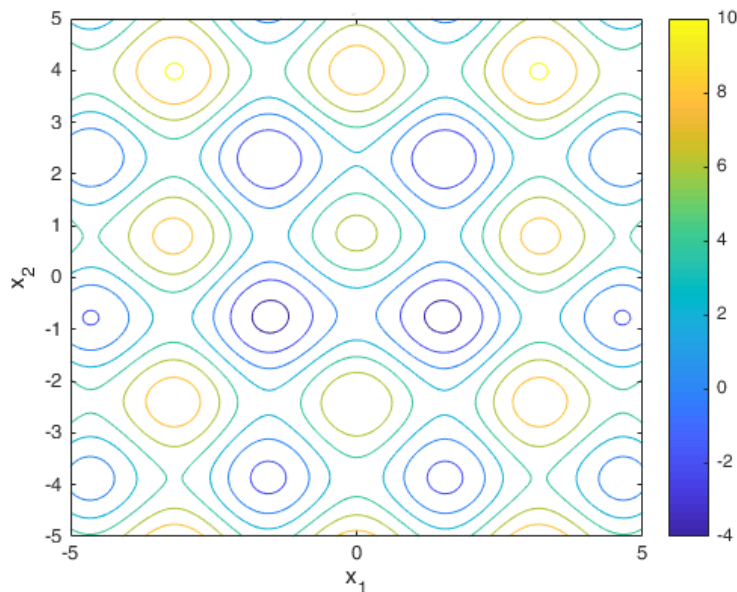
Είναι κατανοητό ότι όσο αυξάνεται το πλήθος των πιθανών λύσεων ανά μεταβλητή απόφασης, τόσο πιο δύσκολο είναι να ελεγχθούν όλες οι πιθανές λύσεις. Αντίστοιχα, αν στο παραπάνω παράδειγμα οι μεταβλητές ορίζονταν στον πραγματικό χώρο ($x_i \in \mathbb{R}$), δηλαδή στο διάστημα $[-5,5]$, αυτομάτως ο έλεγχος όλων των πιθανών συνδυασμών θα απαιτούσε άπειρο χρόνο.

Ας υποθέσουμε τώρα ότι θέλαμε να αποτυπώσουμε στον παραπάνω χώρο λύσεων (Εικόνα 3.2) την ποιότητα (πόσο καλές είναι δηλαδή) των λύσεων για κάθε συνδυασμό του παρακάτω γραφήματος:



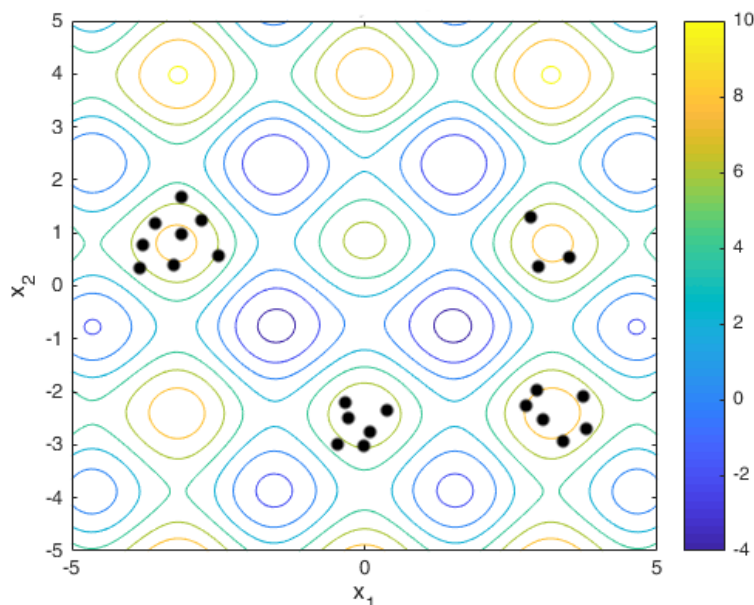
Εικόνα 3.3: Παράδειγμα γραφήματος ποιότητας λύσεων

Ο χώρος λύσεων θα έμοιαζε όπως φαίνεται στην Εικόνα 3.4, παρακάτω. Δεδομένου ότι το κίτρινο χρώμα απεικονίζει τις καλές λύσεις, δηλαδή εκείνες που μεγιστοποιούν την αντικειμενική συνάρτηση του παραδείγματος, παρατηρούμε ότι όσο πιο κοντά βρισκόμαστε σε κάποια «κορυφή» τόσο πιο έντονο είναι το κίτρινο χρώμα. Επειδή ο χώρος λύσεων είναι συνεχής σε αυτό το παράδειγμα, είναι κατανοητό ότι είναι αδύνατον να ελεγχθούν όλες οι πιθανές λύσεις, καθώς θα απαιτούνταν άπειρος χρόνος. Ωστόσο, ένας Εξελικτικός Αλγόριθμος θα ήταν δυνατό να ελέγξει ένα μέρος των πιθανών λύσεων, εξελίσσοντάς τις αρχικές υποψήφιες λύσεις.



Εικόνα 3.4: Χώρος λύσεων παραδείγματος

Αν υποθέσουμε ότι ο παραπάνω χώρος λύσεων περιγράφει ένα πρόβλημα μεγιστοποίησης, οι καλές λύσεις είναι στα σημεία που το χρώμα γίνεται πιο ανοιχτό κίτρινο, όπως φαίνεται και στην Εικόνα 3.3. Οι λύσεις στις κορυφές ονομάζονται τοπικά βέλτιστα του χώρου λύσεων. Για να προσεγγίσουμε τις βέλτιστες λύσεις (optimal solutions) θα έπρεπε να συγκεντρωθούν οι αναζητητές γύρω από τις περιοχές με το κίτρινο χρώμα, όπως φαίνεται στην Εικόνα 3.5.



Εικόνα 3.5: Συσσώρευση αναζητητών γύρω από τοπικά βέλτιστα

Όσο περισσότεροι αναζητητές συσσωρεύονται γύρω από μια βέλτιστη ή οριακά¹ βέλτιστη (near optimal) λύση, τόσο υψηλότερο είναι το επίπεδο της εκμετάλλευσης (exploitation).

Ο στόχος σε κάθε Νοήμονα Αλγόριθμο (Intelligent Algorithm) είναι να επιτευχθεί μια ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation). Δηλαδή να παράγονται τόσες πιθανές λύσεις, ώστε να εξερευνούν όσο το δυνατόν μεγαλύτερο μέρος του χώρου, αλλά και να «εκμεταλλεύονται» (να κάνουν μικρά βήματα) όσο το δυνατόν περισσότερο τον χώρο γύρω από τα τοπικά βέλτιστα.

3.5 Μηχανισμοί Νοημόνων Αλγορίθμων Εμπνευσμένων από τη Φύση

Μελετώντας τους Νοήμονες Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση, μπορούν να παρατηρηθούν κάποια βασικά χαρακτηριστικά που συντελούν έναν Νοήμονα Αλγόριθμο Εμπνευσμένο από τη Φύση. Η παρούσα ενότητα αναλύει

¹ Εννοώντας πλησίον του βέλτιστου.

αυτά τα χαρακτηριστικά και τα συνθέτει σε ένα μοτίβο (pattern) που σε μεγάλο βαθμό ακολουθούν οι μέθοδοι που είναι Εμπνευσμένες από τη Φύση.

3.5.1 Αναζητητές και Επαναληπτική Διαδικασία

Ο όρος αναζητητής (agent) αναφέρεται στο διάνυσμα:

$$\vec{X}_i = (x_i^1, x_i^2, x_i^3, \dots, x_i^d) \quad (3.5.1)$$

το οποίο έχει μήκος ίσο με το πλήθος d των μεταβλητών απόφασης του προβλήματος και με x_i^d συμβολίζεται η d -οστή μεταβλητή απόφασης του i αναζητητή, όπου το $i = \{1, 2, 3, \dots, N\} \in \mathbb{Z}$ και N είναι το πλήθος των αναζητητών του πληθυσμού. Κάθε αναζητητής περιέχει μια υποψήφια λύση (candidate solution) του προβλήματος.

Όπως αναφέρεται παραπάνω, η βασική φιλοσοφία πίσω από τους Εξελικτικούς (και κατ' επέκταση και τους Εμπνευσμένους από τη Φύση) Αλγόριθμους είναι η συνεχής εξέλιξη της λύσης. Δηλαδή η δημιουργία μιας νέας λύσης βασισμένη στην πληροφορία που περιέχει η παλαιότερη λύση. Για να επιτευχθεί αυτό ορίζεται μια επαναληπτική διαδικασία, κατά την οποία κάθε αναζητητής μετακινείται στον χώρο λύσεων. Η επαναληπτική διαδικασία ολοκληρώνεται όταν ικανοποιείται το κριτήριο τερματισμού της. Συνήθως, στους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση το κριτήριο τερματισμού είναι ο μέγιστος αριθμός επαναλήψεων, ο οποίος ορίζεται εξ' αρχής.

3.5.2 Αρχικοποίηση

Στην αρχή των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση γίνεται αρχικοποίηση (initialization) παραμέτρων και λύσεων. Οι παράμετροι του κάθε αλγόριθμου μπορεί να είναι διαφορετικοί. Ωστόσο, κοινές ανοικτές παράμετροι των μεθόδων που εξετάζει η παρούσα διατριβή είναι ο πληθυσμός (population) και οι γενιές (generations). Οι γενιές αναφέρονται και ως επαναλήψεις (iterations). Στο στάδιο της Αρχικοποίησης καθορίζονται οι επαναλήψεις που θα εκτελεστούν μέχρι να ολοκληρωθεί ο αλγόριθμος (βλ. επαναληπτική διαδικασία παραπάνω), καθώς και ο πληθυσμός, δηλαδή ο αριθμός των αναζητητών (agents) που θα διερευνήσουν τον χώρο λύσεων για μια βέλτιστη λύση.

Η αρχικοποίηση των λύσεων, συνήθως, γίνεται με τυχαίο τρόπο. Αν ο αλγόριθμος βασίζεται στη βελτίωση ενός πληθυσμού (population-based), τότε παράγονται N τυχαίες λύσεις, ενώ αν βασίζεται στο μονοπάτι (trajectory-based) που ακολουθεί μια λύση μέχρι να βρει μια οριακά βέλτιστη τιμή (optima), τότε παράγεται μια τυχαία λύση αρχικά. Παραδείγματα αλγορίθμων που βασίζονται στη βελτίωση ενός πληθυσμού (population-based) είναι ο αλγόριθμος Μεγάλης Έκρηξης – Μεγάλης Σύνθλιψης (Big Bang – Big Crunch), η Βελτιστοποίηση Αποικίας Τερμιτών (Termite Colony Optimization) και οι περισσότερες μέθοδοι Νοημοσύνης Σμήνους (Swarm Intelligence). Αντιπροσωπευτικά παραδείγματα Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση οι οποίοι βασίζονται στο μονοπάτι που ακολουθεί μια

λύση (trajectory-based) είναι ο αλγόριθμος Έξυπνης Σταγόνας Νερού (Intelligent Water Drop) και ο αλγόριθμος Δυνάμεων Δημιουργίας Ποταμιών (River Formation Dynamics), οι οποίοι ακολουθούν τη βέλτιστη λύση, καθώς αυτή βελτιώνεται μέσα από κάποια βήματα.

3.5.3 Αναζήτηση στον Χώρο Λύσεων και Εξέλιξη της Λύσης

Το επόμενο βήμα είναι η βελτίωση των λύσεων, είτε μέσω κάποιων κανόνων (rule-based), είτε μέσω μιας σειράς συναρτήσεων (equation-based) που μοντελοποιούν το φυσικό ανάλογο από το οποίο εμπνέεται ο αλγόριθμος. Το καλύτερο παράδειγμα μεθόδου που βελτιώνει την λύση μέσω κανόνων (rule-based) είναι ο αλγόριθμος Αναζήτησης Αρμονίας (Harmony Search), ο οποίος βελτιώνει την αρχική λύση-«μελωδία» μέσω των βημάτων που ένας συνθέτης παράγει νέα μελωδία (διατήρηση ήδη υπάρχουσας μελωδίας, τροποποίηση μιας υπάρχουσας μελωδίας, αυτοσχεδιασμός). Αντιθέτως, η Βελτιστοποίηση Κεντρομόλου Δυνάμεως (Central Force Optimization) βελτιώνει τις λύσεις μέσω εξισώσεων που προέρχονται από τη θεωρία της Φυσικής και ειδικότερα από το φαινόμενο της κεντρομόλου δυνάμεως. Εδώ πρέπει να σημειωθεί ότι υπάρχουν μέθοδοι που μπορούν να θεωρηθούν μικτές (mixed), καθώς βελτιώνουν τις λύσεις και με τους δύο τρόπους. Συνήθως, τέτοιες μέθοδοι είναι αυτές των οποίων οι λύσεις αλληλεπιδρούν μεταξύ τους. Ένα τέτοιο παράδειγμα είναι ο Αλγόριθμος Βαρυτικής Αναζήτησης (Gravitational Search Algorithm), ο οποίος βελτιώνει μεν την κάθε λύση μέσω των εξισώσεων που διέπουν το νόμο της βαρύτητας, αλλά παράλληλα τις μετακινεί μέσα στον χώρο λύσεων βάσει ενός κανόνα.

Η αλληλεπίδραση των λύσεων ή όχι, καθορίζει αν ένας αλγόριθμος είναι βασισμένος στην έλξη μεταξύ αναζητητών (attraction-based) ή όχι (non-attraction-based). Στην ουσία οι μέθοδοι που βασίζονται στην έλξη μεταξύ αναζητητών (attraction-based) έχουν έναν μηχανισμό που η καλύτερη λύση προσελκύει τις υπόλοιπες προς το μέρος της, κάτι που συνήθως γίνεται με διανύσματα (vectors). Η διαφορά αυτών των μεθόδων από τις εκείνες που δεν βασίζονται στην έλξη μεταξύ των αναζητητών (non-attraction) είναι ότι ο πληθυσμός βελτιώνεται συνεχώς, λόγω της αλληλεπίδρασης των λύσεων και έτσι συγκλίνει πιο γρήγορα σε ολικά βέλτιστα (global optima). Ένας ευρέως χρησιμοποιούμενος αλγόριθμος, στον οποίο η καλύτερη λύση προσελκύει τις υπόλοιπες προς το μέρος της, είναι η Βελτιστοποίηση βάσει του φαινομένου του Ηλεκτρομαγνητισμού (Electromagnetism-like Optimization), ενώ γνωστοί αλγόριθμοι που δεν έχουν τέτοια έλξη μεταξύ λύσεων είναι η Βελτιστοποίηση εμπνευσμένη από τις Χημικές Αντιδράσεις (Chemical-reaction-inspired Optimization), η Βελτιστοποίηση βασισμένη στη Βιο-γεωγραφία (Biogeography-based Optimization) και η Βελτιστοποίηση Σύγκρουσης Σωμάτων (Colliding Bodies Optimization). Οι αλγόριθμοι Νοημοσύνης Σμήνους (Swarm Intelligence) είναι σε μεγάλο ποσοστό βασισμένοι στην έλξη μεταξύ αναζητητών

(attraction-based), καθώς η θεωρία του σμήνους στηρίζεται στην αλληλεπίδραση των έμβιων οργανισμών μεταξύ τους ανήκοντας σε ομάδες.

3.5.4 Αξιολόγηση Λύσεων και Ενημέρωση Καλύτερης Λύσης

Ύστερα γίνεται αξιολόγηση των νέων λύσεων (fitness evaluation). Εδώ συνήθως χρησιμοποιείται κάποιος μηχανισμός αποδοχής-απόρριψης λύσεων, όπως το Metropolis Criterion (Metropolis et al., 1953) που περιέχεται στην Προσομοιωμένη Ανόπτηση (Simulated Annealing). Το κριτήριο Metropolis (Metropolis Criterion) αποδέχεται κάποιες λύσεις που είναι χειρότερες της καλύτερης να γίνουν αποδεκτές βάσει μιας πιθανότητας, με σκοπό την εξερεύνηση (exploration) του χώρου. Πολλές φορές ο μηχανισμός αυτός έχει σχέση με Πιθανοθεωρητικά Μοντέλα, με σκοπό να κάνει πιο αυστηρό τον αλγόριθμο στην αποδοχή νέων λύσεων, ώστε να μην «κολλάει» σε τοπικά βέλτιστα, αλλά να έχει υψηλότερο ποσοστό εξερεύνησης στην αρχή και υψηλότερο ποσοστό εκμετάλλευσης στο τέλος της αλγοριθμικής διαδικασίας.

Τέλος, γίνεται ενημέρωση (update best solution) της βέλτιστης λύσης που έχει βρεθεί μέχρι εκείνο το σημείο. Ύστερα ο αλγόριθμος επαναλαμβάνεται μέχρι να ικανοποιηθεί ένα κριτήριο τερματισμού (π.χ. μέγιστος αριθμός επαναλήψεων).

3.5.5 Ψευδοκώδικας και Διάγραμμα Ροής Νοημόνων Αλγορίθμων Εμπνευσμένων από τη Φύση

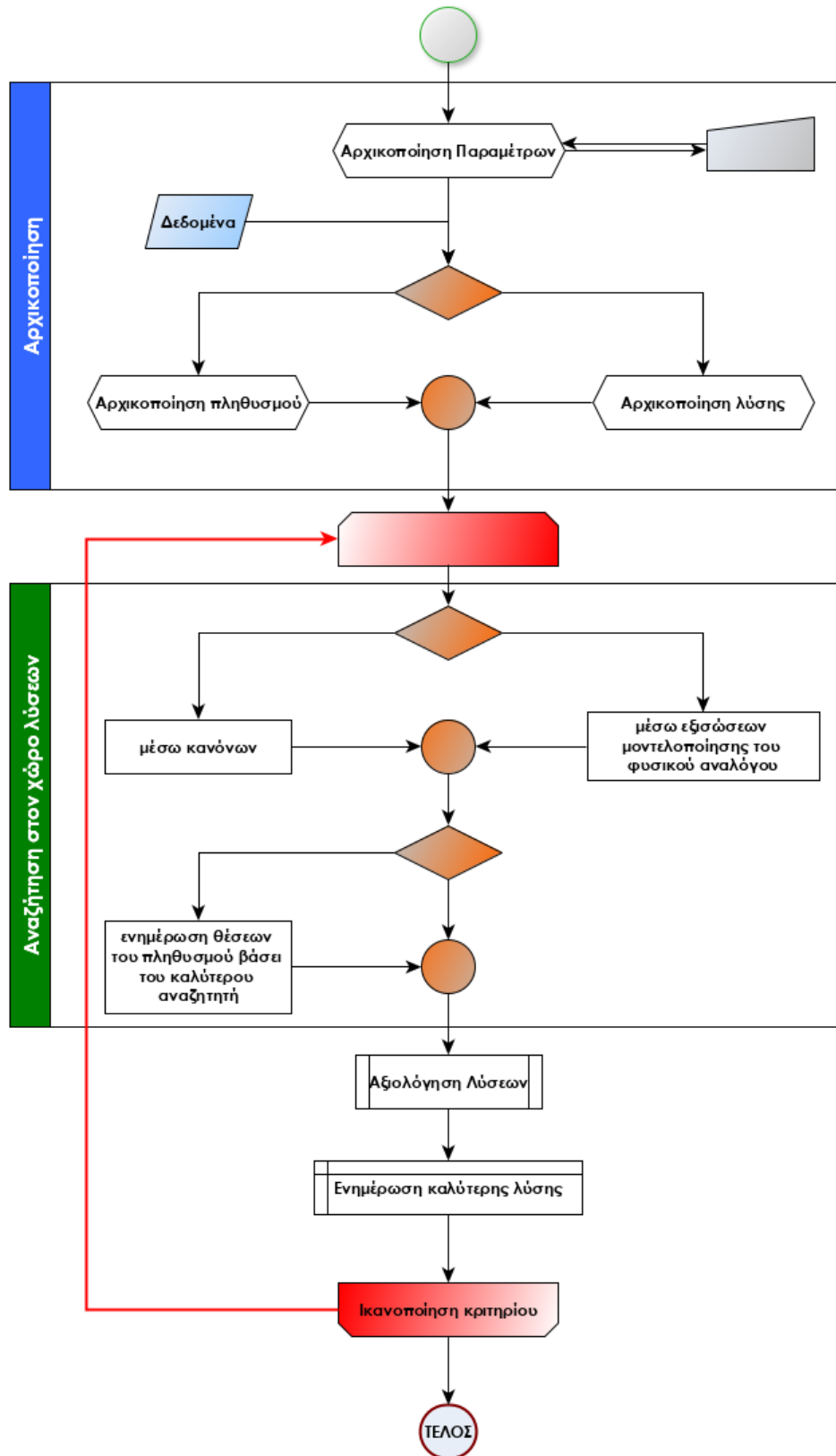
Ο παρακάτω ψευδοκώδικας συνθέτει όλα τα κομμάτια που αναφέρονται σε αυτή την ενότητα.

Ψευδοκώδικας 3.4: Γενική Μορφή Αλγορίθμων Εμπνευσμένων από τη Φύση

- 1: Αρχικοποίηση παραμέτρων
 - 2: Αρχικοποίηση αναζητητών/-ητή
 - 3: Όσο κριτήριο τερματισμού δεν ικανοποιείται
 - 4: Αναζήτηση στον χώρο λύσεων
 - 5: Εξέλιξη κάθε λύσης
 - 6: Ενημέρωση καλύτερης λύσης και Αντικειμενικής Συνάρτησης
 - 7: Τέλος
-

Στο Σχήμα 3-4 απεικονίζεται διαγραμματικά το μοτίβο που ακολουθούν οι μέθοδοι που είναι Εμπνευσμένες από τη Φύση.

Σχήμα 3-4: Μοτίβο μεθόδων εμπνευσμένων από Φαινόμενα της Φύσης



3.6 Προτεινόμενο Πλαίσιο Ανάπτυξης και Βελτίωσης Αλγορίθμων Εμπνευσμένων από τη Φύση

Μελετώντας τη βιβλιογραφία, διαπιστώνονται μερικά προβλήματα στην ανάπτυξη νέων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση. Σε συνδυασμό με τις παρατηρήσεις του 2^{ου} Κεφαλαίου, η παρούσα ενότητα παραθέτει αυτά τα προβλήματα και προτείνει λύσεις που συνθέτουν ένα πλαίσιο εργασίας (framework).

3.6.1 Αναζήτηση πηγής έμπνευσης

Πρόσφατα, μερικές εργασίες (Fister, Strnad and Yang, 2015; Sörensen, 2015; Fister Jr et al., 2016) αμφισβήτησαν την ύπαρξη ενός φυσικού αναλόγου πίσω από αρκετές μεθόδους που προτάθηκαν ως Εμπνευσμένες από τη Φύση (nature-inspired). Όπως αναλύθηκε και στην ενότητα 2.2.1, για να θεωρηθεί ένας αλγόριθμος ότι είναι εμπνευσμένος από τη φύση (nature-inspired) πρέπει να στηρίζεται σε ένα φυσικό ανάλογο, δηλαδή να μοντελοποιεί αλγοριθμικά μια διαδικασία που ακολουθεί κάποιο έμβιο ον στη φύση με σκοπό την εύρεση τροφής, την επιλογή κατάλληλου ζευγαριού για αναπαραγωγή καλύτερων απογόνων, την αποφυγή εχθρών, την προσαρμογή στο περιβάλλον ή να χρησιμοποιεί τους νόμους που διέπουν ένα φυσικό φαινόμενο. Διαφορετικά, η προτεινόμενη μέθοδος που δεν ικανοποιεί αυτή τη συνθήκη μπορεί να θεωρηθεί απλά μετα-ερευνητική (meta-heuristic).

Κάτι που επίσης εγείρει ερωτήματα είναι αν η στρατηγική στην οποία στηρίζεται ένας Αλγόριθμος Εμπνευσμένος από τη Φύση είναι η βέλτιστη στρατηγική που μπορεί να ακολουθήσει ο οργανισμός από τον οποίο έχει εμπνευστεί ο αλγόριθμος ή είναι μια απλώς ικανοποιητική στρατηγική για να επιλύσει ένα από τα προαναφερόμενα προβλήματα. Για παράδειγμα, ο τρόπος που συνεργάζονται οι μέλισσες στην συλλογή γύρης και την αποθήκευσή της στη φωλιά (Lu and Zhou, 2008) μπορεί να είναι λειτουργικός, αλλά να υπάρχει ένας καλύτερος τρόπος και να μην έχει υιοθετηθεί ακόμη (πράγμα που συμβαίνει μέσα από δοκιμές της φύσης σε βάθος εκατοντάδων ετών). Παρόλα αυτά, ως πηγή έμπνευσης μπορεί να χρησιμοποιηθεί οποιαδήποτε επιτυχής στρατηγική (δηλαδή μια στρατηγική που αποδεδειγμένα δουλεύει).

Ένα άλλο θέμα που πρέπει να θιχθεί σχετικά με την πηγή έμπνευσης είναι η αναπαραγωγή μεθόδων που βασίζονται στη Βελτιστοποίηση Σμήνους Σωματιδίων. Αρκετές μέθοδοι Νοημοσύνης Σμήνους, όπως η Βελτιστοποίηση Τροφοληψίας Βακτηρίων (Das et al., 2009), ο Αλγόριθμος Σμήνους Πτηνών (Meng et al., 2016), το Κοπάδι Κριλ (Gandomi and Alavi, 2012), η Βελτιστοποίηση Σμήνους Γατών (Chu, Tsai and Pan, 2006) και η Βελτιστοποίηση Σμήνους Κότας (Meng et al., 2014) είναι απλά τροποποιήσεις της Βελτιστοποίησης Σμήνους Σωματιδίων.

Οι Eberhart and Kennedy (1995) στηρίχτηκαν στην παρατήρηση ότι όλα τα μέλη ενός σμήνους ανακατευθύνουν το πέταγμά τους βάσει της μετακίνησης του

κοντινότερου μέλους γύρω τους. Η εξίσωση που χρησιμοποιήθηκε στη Βελτιστοποίηση Σμήνους Σωματιδίων παρουσιάζεται στην παρακάτω σχέση:

$$x_{k+1}^i = x_k^i + v_{k+1}^i \quad (3.6.1)$$

όπου το x_k^i εκφράζει τη θέση του i -οστού αναζητητή στην επανάληψη k , με x_{k+1}^i εκφράζεται η θέση του i -οστού αναζητητή στην επόμενη επανάληψη (δηλαδή στο νέο βήμα που θα γίνει) και με v_{k+1}^i εκφράζεται η ταχύτητα του i -οστού αναζητητή για την επόμενη επανάληψη. Αρχικά, ο υπολογισμός των ταχυτήτων γινόταν τυχαία, ενώ μετέπειτα οι Kennedy and Eberhart (1995) πρόσθεσαν την πληροφορία της θέσης του καλύτερου σωματιδίου και του τοπικού βέλτιστου, όπως φαίνεται στη σχέση:

$$v_{k+1}^i = c_1 \cdot rand \cdot (x_k^{best} - x_k^i) + c_2 \cdot rand \cdot (x^{best} - x_k^i) \quad (3.6.2)$$

όπου $rand$ είναι ένας τυχαίος αριθμός στο διάστημα (0,1), c_1 και c_2 είναι παράγοντες μάθησης (learning factors), με x_k^{best} συμβολίζεται η θέση του καλύτερου αναζητητή στην τρέχουσα επανάληψη k και x^{best} είναι η βέλτιστη λύση που έχει βρεθεί ως εκείνη τη στιγμή.

Τροποποιώντας την παραπάνω σχέση, εμφανίστηκαν αρκετοί αλγόριθμοι που οι συγγραφείς αναφέρουν ότι “το X [ζώο] δεν κινείται με αυτό τον τρόπο στο περιβάλλον του, αλλά...” ή “τα Y [ζώα] δεν δημιουργούν ομάδες/σμήνη, ωστόσο...”. Αυτό το φαινόμενο συναντάται για παράδειγμα στον Αλγόριθμο Βελτιστοποίησης Νυχτοπεταλούδας και Φλόγας (Mirjalili, 2015a), όπου ο συγγραφέας αναφέρει ότι “...αυτή η υπόθεση έγινε για την ανάπτυξη του αλγόριθμου MFO, ενώ πιθανώς δεν κινούνται με αυτό τον τρόπο οι νυχτοπεταλούδες.”

Χαρακτηριστική είναι η γενίκευση της σχέσης (3.6.2) που παρατηρείται στη Βελτιστοποίηση Μετανάστευσης Ζώων (Li, Zhang and Yin, 2014):

$$x_{k+1}^i = x_k^i + \delta \cdot (x_k^{best} - x_k^i) \quad (3.6.3)$$

όπου το δ μπορεί να μεταβληθεί ανάλογα το πρόβλημα, αλλά οι συγγραφείς χρησιμοποιούν μια γεννήτρια τυχίων αριθμών στην υλοποίηση της μεθόδου τους. Η ομοιότητα με τη σχέση (3.6.1) είναι προφανής, ενώ επίσης η σχέση (3.6.3) μπορεί να θεωρηθεί μια γενίκευση της (3.6.2) χωρίς την ύπαρξη παραγόντων μάθησης c_1 και c_2 .

Ένα ακόμη πρόβλημα που παρατηρείται είναι να συγχέεται η έννοια του πληθυσμού (population) με αυτή του σμήνους (swarm). Είναι σαφές ότι όλες οι μέθοδοι Νοημοσύνης Σμήνους είναι βασισμένες σε πληθυσμό λύσεων (population-based). Ωστόσο, δεν ισχύει το αντίθετο: μια μέθοδος που εξελίσσει έναν πληθυσμό λύσεων δεν είναι απαραίτητα και μέθοδος Νοημοσύνης Σμήνους. Ένας περιεκτικός ορισμός για το τι θεωρείται Νοημοσύνη Σμήνους δίνεται από τους Blum and Li (2008): “[Νοημοσύνης Σμήνους είναι ο] σχεδιασμός νοημόνων συστημάτων πολλών αναζητητών (multi-agent) που αντλούν εμπνευση από τη συλλογική συμπεριφορά

(collective behavior) κοινωνικών εντόμων όπως τα μυρμήγκια, οι τερμίτες, οι μέλισσες και οι σφήκες, καθώς επίσης και από άλλες κοινωνίες ζώων όπως τα σμήνη πουλιών ή τα κοπάδια ψαριών”.

Ο Αλγόριθμος Πυροτεχνημάτων (Firework Algorithm) (Tan and Zhu, 2010) είναι ένα παράδειγμα μεθόδου που συγγέει τη Νοημοσύνη Σμήνους με την ύπαρξη πληθυσμού λύσεων, καθώς αποκαλείται μέθοδος Νοημοσύνης Σμήνους, ενώ πρακτικά δεν είναι ούτε Αλγόριθμος Εμπνευσμένος από τη Φύση, σύμφωνα με τον ορισμό που δίνεται στην ενότητα 2.2.1.

3.6.2 Μέτρηση αποδοτικότητας

Στο Κεφάλαιο 2 θίγεται το ζήτημα της χρησιμότητας των Νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση και της εφαρμογής τους σε πραγματικά προβλήματα. Ωστόσο, για την μέτρηση της αποδοτικότητάς τους είναι αναγκαία η σύγκριση με τα αποτελέσματα άλλων μεθόδων. Η σύγκριση αυτή πρέπει να γίνει στην ίδια βάση, δηλαδή να χρησιμοποιηθούν οι ίδιες τιμές παραμέτρων (πληθυσμού και επαναλήψεων) και αν είναι δυνατό οι ίδιες πειραματικές προδιαγραφές (experimental specifications). Πειραματικές προδιαγραφές θεωρούνται οι τεχνικές προδιαγραφές του μηχανήματος που έγιναν τα πειράματα και του λογισμικού που χρησιμοποιήθηκε για την εκπόνησή τους.

Εφόσον είναι εφικτό, πρέπει να γίνονται τα ίδια πειράματα και για τις μεθόδους που χρησιμοποιούνται ως σύγκριση. Αυτό προαπαιτεί να υπάρχουν διαθέσιμοι οι κώδικες των μεθόδων αυτών ή να είναι εύκολο να αναπαραχθούν (να είναι γνωστές οι τιμές κάθε παραμέτρου και οι προσαρμογές των αλγορίθμων για το εκάστοτε πρόβλημα). Επειδή συχνά δεν ικανοποιούνται αυτές οι συνθήκες, προτείνεται να επιλέγονται οι μέθοδοι σύγκρισης (benchmarks) με βάση:

- α) το ποια μέθοδος έχει την καλύτερη απόδοση ως τώρα,
- β) τις κλασικές μεθόδους που μπορούν να χρησιμοποιηθούν για την επίλυση του προβλήματος και
- γ) τις μεθόδους που είναι πιο σταθερές (robust) στην απόδοσή τους.

Η καλύτερη απόδοση που έχει επιτευχθεί στη βιβλιογραφία θέτει έναν στόχο που δείχνει πόσο κοντά είναι η απόδοση της προτεινόμενης μεθόδου σε αυτόν. Επειδή οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση είναι στοχαστικές μέθοδοι, κατά πάσα πιθανότητα η καλύτερη απόδοση θα έχει επιτευχθεί μία ή κάποιες λίγες φορές. Οπότε είναι απαραίτητη η στατιστική ανάλυση της απόδοσης της προτεινόμενης μεθόδου, μέσω της κατανομής που δημιουργείται από πολλαπλές – ανεξάρτητες μεταξύ τους – εκτελέσεις της μεθόδου. Όσο πιο κοντά στην βέλτιστη τιμή βρίσκεται η μέση απόδοση και όσο πιο μικρή τυπική απόκλιση εμφανίζεται στα αποτελέσματα ανεξάρτητων εκτελέσεων της προτεινόμενης μεθόδου, τόσο πιο σταθερή (robust) θεωρείται. Επιπλέον, χρήσιμο σημείο αναφοράς αποτελεί η απόδοση των κλασικών μεθόδων της βιβλιογραφίας, όπως ο γραμμικός και ο

δυναμικός προγραμματισμός ή καθιερωμένες (established) μέθοδοι Νοημοσύνης Εμπνευσμένης από τη Φύση.

Άλλα χρήσιμα μέτρα αποδοτικότητας είναι η ανάλυση σύγκλισης (convergence analysis), η ισορροπία εξερεύνησης και εκμετάλλευσης (exploration and exploitation balance), ο λόγος της απόδοσης ως προς τον αριθμό των πιθανών λύσεων που αξιολογούνται:

$$\frac{\text{quality}}{\text{amount of solutions that are investigated}} \quad (3.6.4)$$

Κάθε μέτρο αποδοτικότητας εκφράζει διαφορετικά χαρακτηριστικά της μεθόδου που ελέγχεται. Η ανάλυση σύγκλισης (convergence analysis) απεικονίζει πόσο γρήγορα συγκλίνει ο αλγόριθμος στη βέλτιστη λύση και άρα για πόσο υπολογιστικό χρόνο (επαναλήψεις) γίνεται άσκοπη χρήση υπολογιστικών πόρων. Δηλαδή, πόσο γρήγορα συγκλίνει σε μια λύση ο αλγόριθμος, την οποία δε βελτιώνει περαιτέρω μέχρι να ολοκληρωθούν όλες οι επαναλήψεις του. Ίδανικά ο αλγόριθμος κάνει μεγαλύτερες βελτιώσεις στην αρχή της διαδικασίας του και μικρότερες βελτιώσεις προς το τέλος της. Εάν ο αλγόριθμος συγκλίνει γρήγορα σε ένα βέλτιστο και δεν εξελίσσεται, τότε συμβαίνει η επονομαζόμενη πρόωρη σύγκλιση (premature convergence). Για την αντιμετώπιση της πρόωρης σύγκλισης έχουν επιστρατευτεί μηχανισμοί μετάλλαξης στις περισσότερες μεθόδους Νοημοσύνης Εμπνευσμένης από τη Φύση, οι οποίοι στηρίζονται στον αντίστοιχο μηχανισμό του Γενετικού Αλγορίθμου.

Πολύ χρήσιμος είναι ο υπολογισμός της ισορροπίας μεταξύ εξερεύνησης και εκμετάλλευσης (exploration and exploitation balance), όπως επισημαίνεται και από τους Ollion and Doncieux (2011). Στη βιβλιογραφία χρησιμοποιείται η μέση απόσταση από το κέντρο του πληθυσμού των αναζητητών (O. Olorunda and A. P. Engelbrecht, 2008) και η διαφοροποίηση (diversity) των λύσεων εντός πληθυσμού (Hussain et al., 2019). Διάφοροι τρόποι μέτρησης της εξερεύνησης και της εκμετάλλευσης συγκεντρώνονται από τους Črepinšek, Liu and Mernik (2013). Ένας εξ αυτών των τρόπων είναι και η εντροπία (Liu, Mernik and Bryant, 2009), που ωστόσο δεν είναι ευρέως διαδεδομένη ως μέτρο ισορροπίας εξερεύνησης και εκμετάλλευσης.

Ο λόγος απόδοσης ως προς τον αριθμό των πιθανών λύσεων που αξιολογούνται (3.6.4), όπου ο δεύτερος συχνά υπολογίζεται από το γινόμενο του αριθμού των αναζητητών επί το πλήθος των επαναλήψεων, εκφράζει την απόδοση του αλγορίθμου με βάση το πλήθος των αξιολογήσεων. Όσο περισσότερες είναι οι αξιολογήσεις, τόσο χαμηλότερος θα είναι ο δείκτης εφόσον η απόδοση είναι παρόμοια (περίπου ίδια καλύτερη λύση). Στον παρονομαστή μπορεί να χρησιμοποιηθεί επίσης ο μέσος χρόνος εκτέλεσης του αλγορίθμου ή άλλα μεγέθη που εκφράζουν κάποιο χαρακτηριστικό της μεθόδου.

3.6.3 Αυτορρύθμιση και προσαρμογή

Σημαντικό σημείο στην ανάπτυξη μιας νέας μεθόδου είναι και η παραμετροποίησή της, δηλαδή η εύρεση των τιμών των παραμέτρων για τις οποίες επιτυγχάνεται η καλύτερη στατιστικά απόδοσή της. Οι παράμετροι που χρήζουν ρύθμισης (parameter tuning) ονομάζονται ανοικτές παράμετροι. Όλοι οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση έχουν ανοικτές παραμέτρους τον αριθμό των αναζητητών (πληθυσμός) και τον αριθμό των επαναλήψεων (γενιές).

Αν προστεθούν κι άλλες εκτός αυτών στις ανοικτές παραμέτρους του αλγορίθμου, τότε δυσκολεύει η υλοποίησή του σε διάφορα είδη προβλημάτων. Γι' αυτό είναι αναγκαίος ο καθορισμός των τιμών των παραμέτρων που μεγιστοποιούν την απόδοσή του. Τέτοιου είδους παράμετροι είναι οι παράγοντες μάθησης (learning factors) που χρησιμοποιούνται για τον υπολογισμό της νέας θέσης στη Βελτιστοποίηση Σμήνους Σωματιδίων, οι οποίοι φαίνονται στη σχέση (3.6.2) παραπάνω.

Προτείνεται να γίνεται αυτορρύθμιση των παραμέτρων αυτού του είδους εντός της αλγοριθμικής διαδικασίας, με σκοπό την ευκολότερη χρήση της μεθόδου από άλλους ερευνητές. Η αυτορρύθμιση των παραμέτρων μπορεί να επιτευχθεί με διάφορους τρόπους. Ένα παράδειγμα θα ήταν να δημιουργηθεί ένα διάνυσμα των τιμών της παραμέτρου για τις οποίες έχει διαπιστωθεί ότι μεγιστοποιείται η απόδοση του αλγορίθμου και να επιλέγει την κατάλληλη τιμή ο αλγόριθμος με βάσει κάποιο κριτήριο (π.χ. εξερεύνηση ή εκμετάλλευση).

Επιπλέον, προτείνεται η ύπαρξη ενός μηχανισμού διόρθωσης λύσεων, καθώς συχνά παρατηρείται το φαινόμενο της μετακίνησης ενός αναζητητή εκτός του πεδίου ορισμού του προβλήματος. Στη βιβλιογραφία αυτές οι λύσεις είτε σκοτώνονται (και άρα μειώνεται ο πληθυσμός) είτε αντικαθίστανται από άλλες με τυχαίο τρόπο. Ωστόσο, η πληροφορία που περιέχεται στο διάνυσμα της πιθανής μη εφικτής λύσης μπορεί να είναι σημαντική και άρα πρέπει να την εκμεταλλευτεί η νέα λύση που θα δημιουργηθεί.

3.6.4 Βελτίωση απόδοσης

Σύμφωνα με τους Fister, Strnad and Yang (2015), η συνεχής ανάπτυξη νέων μεθόδων δεν αποφέρει απαραίτητα και αποδοτικές τεχνικές για την επίλυση πραγματικών προβλημάτων. Γι' αυτό, στην παραπάνω εργασία οι συγγραφείς προτείνουν την προσαρμογή (adaptation) και τη δημιουργία υβριδικών σχημάτων (hybridization) που θα περιέχουν την εκάστοτε προτεινόμενη μέθοδο.

Ο όρος προσαρμογή (adaptation) αναφέρεται στην ανάπτυξη εκδόσεων (versions) της μεθόδου για διαφορετικά προβλήματα, δηλαδή στην εκπόνηση των απαιτούμενων προσαρμογών που πρέπει να γίνουν, ώστε να μπορεί ο εκάστοτε αλγόριθμος να αντιμετωπίσει τους περιορισμούς του προβλήματος, τη διαφορετική

μορφή των λύσεων ή να εστιάσει σε συγκεκριμένες περιοχές του χώρου λύσεων μέσω κάποια πληροφορίας.

Η ανάπτυξη υβριδικών σχημάτων αποτελεί μια πολύ καλή στρατηγική ενίσχυσης (boosting) της απόδοσης μιας μεθόδου. Πολλές φορές οι αδυναμίες που παρατηρούνται σε κάποια μέθοδο μπορούν να ξεπεραστούν όταν συνδυαστεί με κάποια άλλη μέθοδο. Για παράδειγμα, αν ένας Αλγόριθμος Εμπνευσμένος από τη Φύση αδυνατεί να εξελίξει λύσεις γύρω από τοπικά βέλτιστα, μπορεί να επωφεληθεί από την συλλογική συμπεριφορά (collective behavior) μιας μεθόδου Νοημοσύνης Σμήνους όπου οι λύσεις γύρω από τα τοπικά βέλτιστα θα ελκύουν κι άλλες γύρω τους. Επιπρόσθετα, αν κάποια μέθοδος ειδικεύεται στην τοπική αναζήτηση (local search), θα μπορούσε να συμπεριληφθεί σε ένα υβριδικό σχήμα με μια μέθοδο καθολικής αναζήτησης (global search) για να αντισταθμίσει η μια μέθοδος το κενό της άλλης.

3.7 Συμπεράσματα

Στόχος του παρόντος κεφαλαίου είναι να διερευνήσει ποια είναι εκείνα τα χαρακτηριστικά που καθιστούν έναν Αλγόριθμο Εμπνευσμένο από τη Φύση αποδοτικό, καθώς και να καθορίσει αν υπάρχει κάποιος αλγόριθμος καλύτερος όλων από πλευράς απόδοσης.

Επιπρόσθετα, στην υποενότητα 3.6 αναγνωρίζονται κάποια προβλήματα που παρουσιάζονται στην ανάπτυξη νέων μεθόδων στη βιβλιογραφία και προτείνεται ένα πλαίσιο εργασίας, το οποίο πρέπει να ακολουθήσει ο ενδιαφερόμενος ερευνητής για να αναπτύξει έναν Αλγόριθμο που είναι Εμπνευσμένος από τη Φύση. Σημαντική παρατήρηση είναι ότι τόσο το πρόβλημα της πρόωρης σύγκλισης όσο και το πρόβλημα της αμφιλεγόμενης ύπαρξης φυσικού αναλόγου παρατηρείται στις μεθόδους Νοημοσύνης Σμήνους και κυριότερα σε πολλές εκ των μεθόδων Νοημοσύνης Εμπνευσμένης από Οργανισμούς. Οι μέθοδοι Νοημοσύνης Εμπνευσμένης από Φυσικά Φαινόμενα και Νόμους των Επιστημών είναι πιο αυστηρά ορισμένες λόγω των φαινομένων που μοντελοποιούν, ενώ η έλξη μεταξύ των αναζητητών δεν είναι κύριο γνώρισμά τους (όπως στις μεθόδους Νοημοσύνης Σμήνους) με αποτέλεσμα να μην αντιμετωπίζουν προβλήματα σύγκλισης σε τοπικά βέλτιστα.

3.7.1 Σε ποιες περιπτώσεις είναι χρήσιμοι οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση;

Ένα επιχείρημα που προβάλλουν όσοι διατηρούν σκεπτικιστική στάση απέναντι στους Αλγορίθμους που είναι Εμπνευσμένοι από τη Φύση είναι το γεγονός ότι σε αρκετές περιπτώσεις μια ευρετική μέθοδος αποδεικνύεται επαρκής για την επίλυση ενός προβλήματος και παράλληλα είναι εύκολα υλοποιήσιμη. Ωστόσο, όσο αυξάνεται το μέγεθος του προβλήματος τόσο αυξάνεται και η δυσκολία του. Τα προβλήματα

που συναντώνται στην βιομηχανία και στην αγορά σπανίως κατηγοριοποιούνται στην κλάση P. Όσο αυξάνονται οι πιθανοί συνδυασμοί των μεταβλητών απόφασης, τόσο περισσότερος χρόνος απαιτείται για τον έλεγχο όλων των λύσεων. Κατ' επέκταση, η βελτιστοποίηση είναι δύσκολη διαδικασία, διότι στόχος είναι να βρεθεί η καλύτερη λύση μεταξύ των καλύτερων χωρίς να γνωρίζουμε εάν υπάρχει κάποια ακόμη καλύτερη λύση που δεν έχει εξερευνηθεί ως τώρα (Weise et al., 2009).

Σύμφωνα με τον Yang (2018), οι παραδοσιακοί αλγόριθμοι έχουν συνήθως αναπτυχθεί για την επίλυση συγκεκριμένων προβλημάτων (problem-specific), αδυνατούν να λύσουν μη γραμμικά (nonlinear), πολλών τοπικών μέγιστων (multimodal) προβλήματα, εκπονούν κατά κύριο λόγο τοπική αναζήτηση (local search), ενώ στην πλειοψηφία τους είναι ντετερμινιστικές μέθοδοι. Το γεγονός ότι οι παραδοσιακοί αλγόριθμοι έχουν συνήθως αναπτυχθεί για την επίλυση συγκεκριμένων προβλημάτων είναι γενικότερο πρόβλημα της Μηχανικής Μάθησης (Machine Learning) και ήταν μια από τις αφορμές που έστρεψαν την έρευνα σε μεθόδους που μπορούν να γενικεύουν τον τρόπο επίλυσης των προβλημάτων.

Αντιθέτως, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση δεν περιορίζονται από τον τύπο του προβλήματος, μπορούν να επιλύσουν προβλήματα μη γραμμικά (nonlinear) και πολλών διαστάσεων (high dimensional) και καλύπτουν μεγαλύτερο μέρος του χώρου λύσεων στην αναζήτησή τους, καθώς διεξάγουν καθολική αναζήτηση (global search). Το πιο σημαντικό χαρακτηριστικό τους, όμως, είναι το γεγονός ότι είναι στοχαστικές μέθοδοι, που σημαίνει ότι απαιτούν λιγότερη μνήμη (χώρο) και μπορούν στον ίδιο υπολογιστικό χρόνο να αναζητήσουν μεγαλύτερο μέρος του χώρου λύσεων από τις ντετερμινιστικές μεθόδους. Συνεπώς, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση αποδεικνύονται χρήσιμα εργαλεία βελτιστοποίησης στην περίπτωση που οι κλασικές μέθοδοι δεν αποδίδουν. Τέτοιες περιπτώσεις είναι τα προβλήματα κλάσεως NP, NP-πληρότητας και NP-δυσμενή.

Επιπρόσθετα, ο στοχαστικός χαρακτήρας τους και η δυνατότητα διεξαγωγής καθολικής αναζήτησης (global search) καθιστά τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση χρήσιμους στη διερεύνηση ύπαρξης άλλων τοπικών βέλτιστων (local optima), τα οποία θα μπορούσαν να οδηγήσουν σε άγνωστες ως τότε γειτονιές λύσεων. Σε αυτή την περίπτωση, οι λύσεις που προσεγγίζονται από μια ευρετική (ντετερμινιστική) μέθοδο μπορούν να δοθούν ως σημείο αρχικοποίησης ενός Αλγορίθμου Εμπνευσμένου από τη Φύση, ώστε να τις εξελίξει και να αναζητήσει καλύτερους πιθανούς συνδυασμούς.

Επιπλέον, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση είναι χρήσιμοι ως μέρος υβριδικών σχημάτων (hybrid schemes). Μπορούν να χρησιμοποιηθούν ως καθολικοί βελτιστοποιητές (global optimizers), όπου θα επωμίζονται την αναζήτηση του χώρου (εξερεύνησης), ενώ μια ευρετική μέθοδος θα αναλαμβάνει την τοπική αναζήτηση γύρω από τις λύσεις που παρέχει ο εκάστοτε Αλγόριθμος Εμπνευσμένος από τη Φύση με σκοπό την εκμετάλλευση. Στη βιβλιογραφία, οι Αλγόριθμοι που

είναι Εμπνευσμένοι από τη Φύση χρησιμοποιούνται επίσης ως μέθοδοι βελτίωσης της απόδοσης άλλων γνωστών νοημόνων μεθόδων, όπως τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) (Aljarah, Faris and Mirjalili, 2018) και οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines) (Zhang, Chen and He, 2010). Άλλα υβριδικά σχήματα που συναντώνται στη βιβλιογραφία περιέχουν Αλγόριθμους Εμπνευσμένους από τη Φύση μαζί με μεθόδους Μηχανικής Μάθησης (Machine Learning) και Βαθιάς Μάθησης (Deep Learning) (Kazem et al., 2013; Valdez, Melin and Castillo, 2014).

3.7.2 Υπάρχει ένας αλγόριθμος που υπερνικά τους υπόλοιπους σε απόδοση;

Το συγκεκριμένο ερώτημα απαντάται επαρκώς από το Θεώρημα της Μη-Υπαρξης-Άκοπου-Γεύματος (No-Free-Lunch Theorem). Βασιζόμενοι στο εν λόγω θεώρημα, πολλοί ερευνητές (Wolpert and Macready, 1995; D. H. Wolpert and W. G. Macready, 1997; Igel and Toussaint, 2005) έχουν αποδείξει ότι δεν υπάρχει ένας αλγόριθμος που να επιλύει όλα τα προβλήματα καλύτερα από οποιονδήποτε άλλο αλγόριθμο. Το Θεώρημα της Μη-Υπαρξης-Άκοπου-Γεύματος (No-Free-Lunch Theorem) μπορεί να συνοψιστεί στην παρακάτω έκφραση (Yang, 2014):

Εάν ένας αλγόριθμος A ξεπερνά σε απόδοση έναν αλγόριθμο B , μετρώντας την απόδοση βάσει μιας αντικειμενικής συνάρτησης, ο B θα είναι καλύτερος του A σε κάποια άλλη αντικειμενική συνάρτηση.

Το παραπάνω θεώρημα καθιστά σαφές ότι ανάλογα με το κριτήριο μέτρησης απόδοσης (π.χ. χρόνος, μέση ποιότητα λύσης, τυπική απόκλιση, κατανομή ποιότητας λύσεων), κάθε αλγοριθμική μέθοδος θα είναι χειρότερη από τις υπόλοιπες τουλάχιστον σε ένα κριτήριο εξ αυτών. Οπότε, εν μέρει η ανάπτυξη νέων μεθόδων είναι αναγκαία, καθώς κάθε μέθοδος συνεισφέρει σε διαφορετικές προσεγγίσεις του προβλήματος. Για παράδειγμα, η ανάγκη μιας απόφασης σε σύντομο χρόνο εις βάρος της εύρεσης μιας ενδεχόμενης καλύτερης απόδοσης, καθιστά χρήσιμη μια γρήγορη μέθοδο. Αντιθέτως, η ανάγκη εύρεσης των καλύτερων στρατηγικών σε ένα πρόβλημα απόφασης, αναδεικνύει τις πολυπλοκότερες μεθόδους οι οποίες όμως παρουσιάζουν υψηλή απόδοση.

Εν κατακλείδι, δεν υπάρχει ένας αλγόριθμος που να υπερνικά τους υπόλοιπους σε απόδοση. Η χρησιμότητα κάθε αλγόριθμου εξαρτάται από τον αντικειμενικό στόχο της επίλυσης του προβλήματος.

3.7.3 Υπάρχει ανάγκη για νέες μεθόδους που να είναι εμπνευσμένες από τη φύση;

Σύμφωνα με το Θεώρημα της Μη-Υπαρξης-Άκοπου-Γεύματος (No-Free-Lunch Theorem) η απάντηση στο παραπάνω ερώτημα θα ήταν καταφατική. Ωστόσο, συνυπολογίζοντας το συνεχώς αυξανόμενο πλήθος των Αλγορίθμων που είναι

Εμπνευσμένοι από τη Φύση και την ύπαρξη πολλών παρόμοιων μεθόδων (όπως αναλύθηκε στο Κεφάλαιο 2), δεν υπάρχει πραγματική ανάγκη για ανάπτυξη νέων μεθόδων που να αντλούν την έμπνευσή τους από τη φύση. Για να εργαστεί κανείς πάνω στην ανάπτυξη νέων μεθόδων, πρέπει να έχει έναν εκ των κάτωθι στόχων:

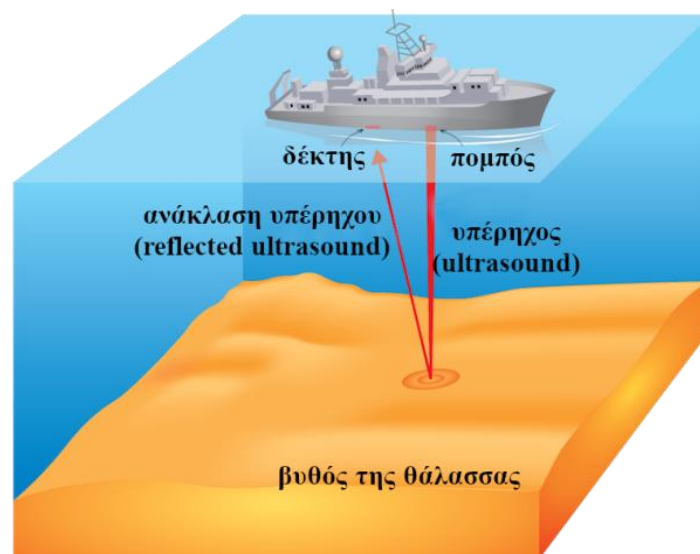
- ✓ την επίλυση ενός προβλήματος που καμία άλλη υπάρχουσα μέθοδος δεν μπόρεσε να λύσει
- ✓ την ενσωμάτωση μηχανισμών στην προτεινόμενη μέθοδο, που θα την καταστήσουν αποδοτικότερη έναντι άλλων

Ο Α. Hatamlou πρότεινε τον αλγόριθμο Μαύρης Τρύπας (Black Hole) (Hatamlou, 2013) και τον αλγόριθμο Καρδιάς (Heart algorithm) (Hatamlou, 2014) για συσταδοποίηση δεδομένων (data clustering). Τα αποτελέσματα του αλγόριθμου Μαύρης Τρύπας έπεισαν κι άλλους ερευνητές να τον χρησιμοποιήσουν (Li et al., 2015; E. Pashaei, M. Ozen and N. Aydin, 2015). Ωστόσο, η ύπαρξη του αλγόριθμου Μαύρης Τρύπας κατέστησε μη αναγκαία την ανάπτυξη του αλγόριθμου Καρδιάς, ένα γεγονός που φάνηκε και στην απόδοση του δεύτερου σε σύγκριση με τον πρώτο σε προβλήματα ελέγχου απόδοσης (benchmark) από τον χώρο της συσταδοποίησης (clustering). Από τη μια, ο αλγόριθμος Μαύρης Τρύπας προτάθηκε για μια κατηγορία προβλημάτων που δεν είχαν εφαρμοστεί οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση ως τότε. Από την άλλη, οποιαδήποτε νέα μέθοδος δεν προσδίδει κάποια αξία στην επίλυση του προβλήματος, εφόσον δεν το προσεγγίζει διαφορετικά ή δεν ενσωματώνει κάποιο μηχανισμό που να ελαχιστοποιεί τις αδυναμίες (drawbacks) των στοχαστικών μεθόδων.

4. Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό

4.1 Φυσικό Ανάλογο

Ο μηχανισμός που μοντελοποιείται στην προτεινόμενη μέθοδο είναι το σύστημα του SONAR, το οποίο χρησιμοποιεί το Ναυτικό στα Πολεμικά πλοία για την αναγνώριση στόχων, όπως π.χ. υποβρυχίων. Η βασική ιδέα πίσω από τον μηχανισμό αυτό είναι η εκπομπή ενός υπέρηχου (ultrasound) με σκοπό την εκτίμηση του όγκου των εμποδίων που συναντά ο υπέρηχος. Βάσει της έντασης του ήχου που αντανακλάται από το εμπόδιο, υπολογίζεται ο όγκος του και κατ' επέκταση μπορεί να αναγνωριστεί πιθανή απειλή (εχθρικά υποβρύχια). Με αυτό τον τρόπο μπορεί το πλήρωμα του πλοίου να υπολογίσει την απόσταση και τον όγκο του αντικειμένου που παρατηρεί (Εικόνα 4.1).



Εικόνα 4.1: Μηχανική του SONAR

Ένα χαρακτηριστικό του SONAR είναι η σάρωση (scan) μιας νοητής σφαίρας γύρω από το πλοίο. Για την μοντελοποίηση αυτού του φαινομένου, χρησιμοποιείται η έννοια της έντασης του ήχου από της αρχές της Ακουστικής (Lurton, 2002). Αρχικά, το Εκπέμπον Σήμα (Acoustic Power Output ή Sound Power) (P) πρέπει να υπολογιστεί:

$$P = \eta \cdot Pe \quad (4.1.1)$$

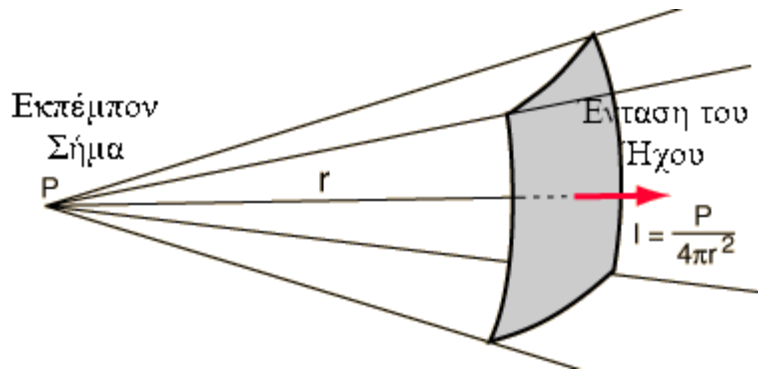
όπου Pe είναι το Εισερχόμενο Σήμα (Power Input) και με η συμβολίζεται η αποδοτικότητα του Πομπού (Transducer efficiency), η οποία ορίζεται ως το ποσοστό επί τις εκατό του εξερχόμενου σήματος προς το εισερχόμενο σήμα. Ακολούθως, η Ένταση του ήχου (Intensity) υπολογίζεται ως ο λόγος μεταξύ Εκπέμποντος Σήματος (Sound Power ή P) και της περιοχής που σαρώνεται, όπως φαίνεται και στην Εικόνα 4.2:

$$I = \frac{P}{area} \quad (4.1.2)$$

όπου το εμβαδόν υπολογίζεται ως:

$$area = 4 \cdot \pi \cdot r^2 \quad (4.1.3)$$

και r είναι η ακτίνα της νοητής σφαίρας που εκτείνεται γύρω από το πλοίο, το οποίο εκπέμπει το σήμα.



Εικόνα 4.2: Η Ένταση του Ήχου συναρτήσει του Εκπέμποντος Σήματος και του Εμβαδού

Είναι εμφανές ότι μια μείωση της έντασης I προκαλεί αύξηση της ενεργού ακτίνας r και κατ' επέκταση της περιοχής που ερευνείται (σαρώνεται). Αυτή η σχέση χρησιμοποιείται εξίσου στην μεθοδολογία του αλγόριθμου.

4.2 Αλγοριθμικό Μοντέλο

Ας υποθέσουμε ότι κάθε αναζητητής $X_i = \{x_i^1, x_i^2, x_i^3, \dots, x_i^n\}$ είναι ένας πομπός (transmitter), όπου $i \in 1, 2, \dots, N$ και N είναι ο μέγιστος αριθμός αναζητητών (agents), ενώ n είναι ο αριθμός των διαστάσεων του προβλήματος. Ο αριθμός των αναζητητών (agents) προκαθορίζεται στην αρχή του αλγόριθμου, όπως άλλωστε γίνεται σε κάθε αλγόριθμο τέτοιας λογικής, ώστε να μην μεταβάλλεται κατά τη διάρκεια της διαδικασίας και κατ' επέκταση να γλιτώσουμε υπολογιστική ισχύ (computational power). Θα μπορούσε να υποθέσει κανείς ότι όσο περισσότεροι αναζητητές υπάρχουν τόσο μεγαλύτερη είναι η πιθανότητα να προσεγγισθεί η καλύτερη λύση. Ωστόσο, όπως φαίνεται από τις παρακάτω υποενότητες που αναλύεται ο αλγόριθμος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό, η πληθώρα σημείων γύρω από τη θέση κάθε αναζητητή στον χώρο λύσεων, μπορεί να αντισταθμίσει αυτή την υπόθεση και να είναι εξίσου ανταγωνιστικός ο αλγόριθμος ακόμη και με μικρό πλήθος αναζητητών. Ένα από τα πλεονεκτήματα του αλγόριθμου αυτού είναι το γεγονός ότι μπορεί να σαρωθεί μεγαλύτερο μέρος του χώρου λύσεων σε σύγκριση με έναν άλλο αλγόριθμο Εμπνευσμένο από τη Φύση, εφόσον και οι δύο αλγόριθμοι έχουν το ίδιο πλήθος αναζητητών.

Πρωταρχικά, οι θέσεις των αναζητητών αρχικοποιούνται κάπου στον χώρο λύσεων. Ο ευκολότερος τρόπος για να γίνει αυτό είναι με την χρήση της ομοιόμορφης κατανομής. Ωστόσο, ανάλογα με το τι κατανομή ακολουθούν οι τιμές κάθε μεταβλητής απόφασης, μπορούν να χρησιμοποιηθούν εύκολα οι (προγραμματιστικές) εντολές για την αντίστοιχη κατανομή. Η γενικευμένη σχέση, ωστόσο, δίνεται από τον τύπο:

$$x_i^d = lower_bound^d + (upper_bound^d - lower_bound^d) \cdot rand \quad (4.2.1)$$

όπου $rand$ είναι ένας τυχαίος αριθμός στο διάστημα (0,1), $lower_bound^d$ και $upper_bound^d$ είναι το κάτω και άνω όριο για κάθε μεταβλητή απόφασης d , αντίστοιχα.

Ψευδοκώδικας 4.1: Αλγόριθμος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization)

- 1: Αρχικοποίηση αναζητητών
 - 2: Αρχικοποίηση Ενεργού Ακτίνας και Έντασης κάθε αναζητητή
 - 3: **Όσο** κριτήριο τερματισμού δεν ικανοποιείται
 - 4: **Εάν** μετρητής επαναλήψεων χωρίς καλύτερη λύση = στιγμιά επανατοποθέτησης
 - 5: Επανατοποθέτηση αναζητητή
 - 6: Επανυπολογισμός Έντασης και Ενεργού Ακτίνας
 - 7: **Τέλος_εάν**
 - 8: Ενημέρωση Ακτίνας για κάθε αναζητητή
 - 9: Υπολογισμός Έντασης για κάθε αναζητητή
 - 10: **Όσο** Σάρωση Πλήρους Κύκλου δεν έχει ολοκληρωθεί
 - 11: Ενημέρωση της Γωνίας Περιστροφής για κάθε διάσταση
 - 12: Υπολογισμός της Αντικειμενικής Συνάρτησης για την πιθανή νέα θέση
 - 13: Αποθήκευση της καλύτερης τιμής Αντικειμενικής Συνάρτησης για τα μέχρι τώρα βήματα της Σάρωσης
 - 14: **Τέλος**
 - 15: Ενημέρωση καλύτερης λύσης και Αντικειμενικής Συνάρτησης
 - 16: Ενημέρωση Έντασης και Σήματος Εκπομπής για κάθε αναζητητή
 - 17: **Τέλος**
-

Χρησιμοποιώντας τις σχέσεις (4.1.2) και (4.1.3) υπολογίζεται η αρχική τιμή της ακτίνας και της έντασης για κάθε αναζητητή. Ορίζουμε το Εισερχόμενο Σήμα (Power Input) ως την τιμή της αντικειμενικής συνάρτησης κι έτσι έχουμε:

$$Pe = fit_i, i \in \{1, 2, \dots, N\} \quad (4.2.2)$$

όπου με fit_i συμβολίζεται η απόδοση του αναζητητή i .

Από την άλλη, η σχέση (4.1.1) μεταβάλλεται ως εξής:

$$P = e^{Pe} \quad (4.2.3)$$

ώστε να μετατρέψουμε τις τιμές της αντικειμενικής συνάρτησης σε θετικούς αριθμούς, καθώς είναι αναγκαίο αυτό για την εισαγωγή τους σε μια σχέση που χρησιμοποιεί λογάριθμους, όπως αναλύεται στην υποενότητα 4.3.1. παρακάτω. Ο λογάριθμος δεν ορίζεται για αρνητικές τιμές, αλλά είναι πολύ πιθανό η αντικειμενική συνάρτηση να λαμβάνει αρνητικές τιμές. Γι' αυτό τροποποιείται κατάλληλα η σχέση (4.1.1) που ισχύει στο φυσικό ανάλογο.

Τα επόμενα βήματα εκτελούνται μέχρι να ικανοποιούνται τα κριτήρια τερματισμού. Πιο συχνό κριτήριο τερματισμού είναι ο μέγιστος αριθμός επαναλήψεων του εκάστοτε αλγόριθμου, όπου εδώ τον ονομάζουμε «πλήθος σαρώσεων» (“number of scans”). Υπολογίζεται η αντικειμενική συνάρτηση για κάθε «πλοίο» (λύση), ώστε να βρεθεί η καλύτερη λύση, η οποία αποθηκεύεται και ύστερα, όλοι οι αναζητητές μεταβάλλουν την Έντασή τους βάσει της καλύτερης λύσης. Εάν κάποιος αναζητητής έχει βρει καλύτερη λύση από την προηγούμενη ολική καλύτερη λύση (global best solution), μειώνει την Έντασή του για να σαρώσει τον χώρο γύρω από τη λύση αυτή, ώστε να ερευνηθεί μήπως υπάρχει κάποια ακόμη καλύτερη στον γύρω χώρο. Αντίθετα, αν έχει βρει χειρότερη λύση από την ήδη υπάρχουσα, αυξάνει την Έντασή του για να εκτελέσει μεγαλύτερα άλματα στον χώρο λύσεων, καθώς η Ένταση επηρεάζει την ακτίνα που ορίζει τη νοητή υπερ-σφαίρα που σαρώνεται γύρω από τον αναζητητή. Ο όρος υπερ-σφαίρα χρησιμοποιείται εδώ, καθώς σαρώνονται οι νοητοί κύκλοι γύρω από τον αναζητητή σε κάθε διάσταση του προβλήματος, όπου οι διαστάσεις συχνά είναι περισσότερες από τρεις.

Τέλος, ένας ακόμη χρήσιμος μηχανισμός που εφαρμόζεται στον αλγόριθμο είναι η επανατοποθέτηση των αναζητητών. Ένας εύκολος τρόπος επανατοποθέτησης των αναζητητών είναι να λαμβάνεται υπόψη η θέση της καλύτερης λύσης που έχει βρεθεί ως εκείνη τη στιγμή:

$$x_i^d = best^d + r_0^i \cdot rand \quad (4.2.4)$$

όπου το x_i^d παριστάνει τη θέση του i αναζητητή στη d διάσταση του προβλήματος, $best^d$ είναι η τιμή του x στην διάσταση d της καλύτερης λύσης που έχει βρεθεί από τον συγκεκριμένο αναζητητή, r_0^i είναι η ενεργός ακτίνα του i αναζητητή και με $rand$ συμβολίζουμε έναν τυχαίο αριθμό από την ομοιόμορφη κατανομή.

Ωστόσο, αυτό το βήμα εκτελείται όταν η τιμή της αντικειμενικής συνάρτησης του αναζητητή είναι κάτω από τον μέσο όρο των αντίστοιχων τιμών όλων των αναζητητών. Διαφορετικά, ο αναζητητής επανατοποθετείται τυχαία μέσω της σχέσης (4.2.1). Ο λόγος που εισήχθηκε αυτός ο κανόνας (Tzanetos and Dounias, 2018) ήταν να διατηρείται ισορροπία μεταξύ σάρωσης μεγαλύτερου μέρους του χώρου λύσεων (exploration) και ελέγχου του χώρου γύρω από κάθε λύση για πιθανές καλύτερες λύσεις (exploitation). Υιοθετώντας την λογική του ποσοστού μετάλλαξης (mutation

rate) που εμφανίζεται στα πρωταρχικά σχήματα γενετικών αλγόριθμων (Nilsson and Snoad, 2002) προκύπτει:

$$\mu_{opt} = \frac{1}{\tau} \quad (4.2.5)$$

όπου με τ συμβολίζεται ο αριθμός των γενιών μεταξύ περιβαλλοντικών αλλαγών (environmental changes). Θέτοντας το όριο των επαναλήψεων χωρίς αλλαγή (ή αλλιώς χωρίς περιβαλλοντική αλλαγή) ως το 1% του αριθμού των επαναλήψεων του αλγορίθμου (iterations), προκύπτει ο τύπος που δίνει την στιγμή επανατοποθέτησης (checkpoint):

$$checkpoint = \text{μέγιστος αριθμός επαναλήψεων} \cdot 0.01 \quad (4.2.6)$$

Το 1% που χρησιμοποιείται εδώ είναι ένα παράδειγμα. Γενικότερα, προτείνεται η χρήση του 5% του αριθμού των επαναλήψεων του αλγορίθμου, το οποίο υποστηρίζουν και τα πειράματα που παρουσιάζονται στα επόμενα κεφάλαια.

4.3 Βασικοί Μηχανισμοί Προτεινόμενης Μεθόδου

4.3.1 Παράμετρος της Έντασης (Intensity)

Η πιο σημαντική παράμετρος στον εν λόγω αλγόριθμο είναι η παράμετρος της Έντασης. Η Ένταση επηρεάζει την ενεργό ακτίνα και κατ' επέκταση την περιοχή την οποία θα ψάξει σε κάθε επανάληψη ο εκάστοτε αναζητητής. Η τιμή της Έντασης επαναπροσδιορίζεται στο τέλος κάθε επανάληψης βάσει της λύσης που έχει βρει ο κάθε αναζητητής. Χρησιμοποιώντας τις ιδιότητες της εκθετικής συνάρτησης, ο επαναπροσδιορισμός αυτός δίνεται από τη σχέση:

$$I_{i_{new}}^d = I_{i_{old}}^d \cdot e^{magnititude^d} \quad (4.3.1)$$

Το *magnitude* καθορίζει το μέγεθος της λύσης που έχει βρει ο αναζητητής και υπολογίζεται ως:

$$magnititude^d = [(scan_best_i - best) \times 10^{-b} + s] \cdot weighted^d \quad (4.3.2)$$

όπου $scan_best_i$ είναι η καλύτερη τιμή της αντικειμενικής συνάρτησης που έχει βρει ο αναζητητής i στην τρέχουσα σάρωση και $best$ ορίζεται η καλύτερη λύση που έχει βρεθεί γενικά (global best) τη δεδομένη στιγμή. Στην αρχική μορφή του αλγορίθμου (Tzanetos and Dounias, 2017a) το *magnitude* είχε οριστεί ως:

$$(scan_best_i - best) + s \quad (4.3.3)$$

κάτι που οδηγούσε την Ένταση (Intensity) να παίρνει την τιμή του άπειρου ύστερα από κάποια βήματα. Οι εξισώσεις ελέγχου (test functions) δεν είχαν δείξει αυτό το πρόβλημα, καθώς οι τιμές που μπορεί να πάρει η αντικειμενική συνάρτηση

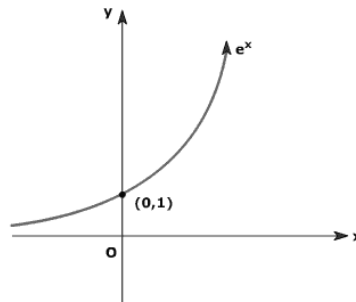
ήταν μικρότερες του ± 1000 . Αντιθέτως, πολλά πραγματικά προβλήματα παρουσιάζουν μεγάλες τιμές στην αντικειμενική συνάρτηση, με αποτέλεσμα να μην είναι διαχειρίσιμες. Γι' αυτό τον λόγο προστέθηκε ο όρος 10^{-b} , όπου το b αναπαριστά το πλήθος των ακέραιων ψηφίων της μεγαλύτερης τιμής αντικειμενικής συνάρτησης που έχει ο χειρότερος αναζητητής στην εκάστοτε επανάληψη. Η παράμετρος αυτή μειώνεται κάθε φορά που όλοι οι αναζητητές έχουν τιμές αντικειμενικής συνάρτησης με λιγότερα ψηφία. Σημειώνεται ότι η σχέση (4.3.2) χρησιμοποιείται σε προβλήματα ελαχιστοποίησης. Στα αντίστοιχα μεγιστοποίησης, οι όροι $scan_best_i$ και $best$ εναλλάσσουν πρόσημα.

Επειδή στην πορεία των πειραματισμών διαπιστώθηκε ότι ο όρος *magnitude* δεν μετέτρεπε την εκάστοτε λύση σε κάθε διάσταση του προβλήματος αναλογικά με το εύρος λύσεων της διάστασης, προτάθηκε (Tzanetos and Dounias, 2018) η στάθμιση του όρου αυτού πολλαπλασιάζοντάς τον με την ποσότητα:

$$weighted^d = \frac{accept_range^d}{\sum_1^d accept_range^d} \quad (4.3.4)$$

όπου $accept_range^d = upper_bound^d - lower_bound^d$ για κάθε διάσταση d του προβλήματος.

Τέλος, επειδή ο καλύτερος αναζητητής θα επέστρεφε μηδέν (0) στη σχέση (4.3.4), καθώς θα αφαιρούταν από τον εαυτό του, προστέθηκε μια πολύ μικρή τιμή $s > 0$, ώστε να απαλείψει το φαινόμενο αυτό. Η τιμή αυτή ορίζεται ανάλογα το πρόβλημα, αυθαίρετα.



Εικόνα 4.3: Γραφική παράσταση της συνάρτησης $y=e^x$

Η εξίσωση (4.3.1) δημιουργήθηκε βάσει των ιδιοτήτων της e^x . Όπως φαίνεται και στην Εικόνα 4.3, όταν το x λαμβάνει τιμές μικρότερες του μηδενός (0), η τιμή της e^x είναι μικρότερη της μονάδας (1). Συνεπακόλουθα, εάν ο όρος *magnitude* έχει αρνητική τιμή (που σημαίνει ότι ο αναζητητής βρήκε καλύτερη τιμή από την ήδη υπάρχουσα, αναφερόμενοι πάντα σε προβλήματα ελαχιστοποίησης), η Ένταση θα μειωθεί μέσω της (4.3.1), καθώς θα πολλαπλασιαστεί με κάποια τιμή μικρότερη της μονάδας. Αντιθέτως, εάν ο όρος *magnitude* έχει θετική τιμή, η Ένταση θα αυξηθεί πολλαπλασιαζόμενη με έναν αριθμό μεγαλύτερο της μονάδας. Με αυτό τον μετασχηματισμό της σχέσης (4.2.3) του φυσικού αναλόγου στην σχέση (4.3.1), όσο

πιο μακριά βρίσκεται ο αναζητητής από την καλύτερη λύση (βάσει αντικειμενικής συνάρτησης), τόσο περισσότερο θα μειωθεί η Ένταση, με αποτέλεσμα να κάνει μεγαλύτερα βήματα στον χώρο για την αναζήτηση κάποιας λύσης.

Ωστόσο, για να μετατραπούν οι πολύ υψηλές τιμές του I_i σε πιο διαχειρίσιμες, χρησιμοποιείται από το φυσικό ανάλογο η σχέση:

$$I_i = 10 \cdot \log \frac{I_i}{I_0} \quad (4.3.5)$$

όπου I_i είναι η Ένταση του i αναζητητή και I_0 συμβολίζουμε το Κατώφλι της Ακουστικής (Threshold of Hearing) (Lurton, 2002), το οποίο ορίζεται ως:

$$I_0 = 10^{-12} \text{ watts}/m^2 = 10^{-16} \text{ watts}/cm^2 \quad (4.3.6)$$

Η τιμή του Κατωφλιού της Ακουστικής (Threshold of Hearing) I_0 ορίζεται 10^{-12} . Αυτή η τιμή είναι σταθερή ασχέτως το πρόβλημα που λύνει ο αλγόριθμος. Πειράματα που είχαν ως στόχο την παραμετροποίηση του αλγορίθμου, έδειξαν ότι η τιμή αυτή δεν επηρεάζει την απόδοση του αλγορίθμου.

4.3.2 Ενεργός Ακτίνα (Effective radius)

Η ενεργός ακτίνα r_0 ορίζει την μέγιστη ακτίνα του νοητού κύκλου γύρω από τον αναζητητή που θα εκτελέσει τα βήματα. Η ενεργός ακτίνα πρέπει να είναι ανάλογη του χώρου λύσης που θα ερευνηθεί. Μια μικρή τιμή της ενεργού ακτίνας οδηγεί τον αλγόριθμο σε μικρότερα βήματα στον χώρο λύσης. Αντιθέτως, μια μεγαλύτερη τιμή της ενεργού ακτίνας οδηγεί σε μεγαλύτερα βήματα, αλλά με το ρίσκο να προσπεράσει πιθανές υποβέλτιστες λύσεις ο αναζητητής.

Αντιστρέφοντας την σχέση (4.1.3), η ενεργός ακτίνα r_0 υπολογίζεται:

$$r_{0_i}^d = \sqrt{\frac{area_i^{dk}}{4 \cdot \pi}} \quad (4.3.7)$$

όπου $area_i^{dk}$ είναι η περιοχή που θα ερευνηθεί από τον i αναζητητή στην d διάσταση όντας στην k επανάληψη του αλγορίθμου. Η πραγματική σχέση μεταξύ των παραμέτρων της Έντασης και της Ακτίνας χρησιμοποιείται και στον αλγόριθμο: σε χαμηλότερη ένταση σαρώνεται μεγαλύτερη περιοχή από ότι σε υψηλότερη ένταση. Ο στόχος είναι να μεγαλώνει η ακτίνα, εφόσον δεν βρίσκει ο αναζητητής καλύτερη λύση, ώστε να μετακινείται μακρύτερα από την εκάστοτε θέση του και κατ' επέκταση να καλυφθεί μεγαλύτερο μέρος του χώρου λύσης.

4.3.3 Σάρωση Πλήρους Κύκλου (Full Scan Loop)

Με απώτερο στόχο να εξερευνήσει ο αλγόριθμος μεγαλύτερο μέρος του χώρου λύσεων, σε κάθε επανάληψη κάθε αναζητητής σαρώνει τον χώρο γύρω του, ο οποίος

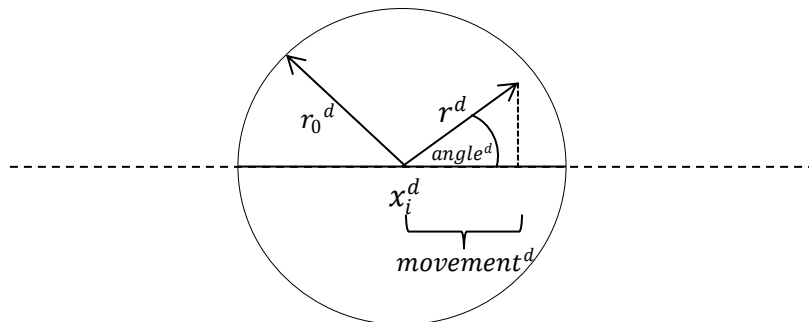
ορίζεται από την νοητή υπερ-σφαίρα ακτίνας r_0 . Αυτή η διαδικασία ονομάζεται Σάρωση Πλήρους Κύκλου (Full Scan Loop), διότι τα παρακάτω βήματα επαναλαμβάνονται έως ότου ολοκληρωθεί μια πλήρης περιστροφή (μια πλήρης σάρωση, όπως στο SONAR). Ξεκινώντας από γωνία 0° , εκτελούνται τυχαίες περιστροφές σε κάθε διάσταση του προβλήματος. Κάθε περιστροφή δεν μπορεί να ξεπεράσει μια προκαθορισμένη γωνία a° και υπολογίζεται ως:

$$angle^d = angle^d + rand \times a^\circ \quad (4.3.8)$$

όπου $rand$ είναι ένας τυχαίος αριθμός από την ομοιόμορφη κατανομή και $angle^d$ είναι η γωνία περιστροφής στη διάσταση d . Εάν κάποια από τις γωνίες περιστροφής $angle^d$ ξεπεράσει τις 360° , η Σάρωση Πλήρους Κύκλου σταματά. Το διάνυσμα γωνιών σε κάθε φάση της σάρωσης μετατρέπεται σε διάνυσμα μετατοπίσεων σε κάθε διάσταση μέσω του τύπου:

$$movement^d = r^d \cdot \cos(angle^d) \quad (4.3.9)$$

όπου r^d είναι μια τυχαία παραγόμενη ακτίνα, η οποία είναι μικρότερη της ενεργού ακτίνας r_0^d της d διάστασης του προβλήματος. Στην Εικόνα 4.4 παρακάτω, παρουσιάζεται σχεδιαστικά ένα παράδειγμα του τρόπου υπολογισμού του βήματος $movement^d$ όπως αυτό υπολογίζεται για κάθε διάσταση d του προβλήματος. Υποθέτουμε ότι το κέντρο του κύκλου, ο οποίος ορίζεται από την ενεργό ακτίνα r_0^d , είναι η παρούσα λύση του i αναζητητή που εξετάζεται. Οι πιθανές νέες λύσεις εξετάζονται σε κάθε φάση της Σάρωσης Πλήρους Κύκλου (Full Scan Loop) μέσω της εξίσωσης (4.3.11) με ένα παράδειγμα να φαίνεται στην Εικόνα 4.4 ως η προβολή του διανύσματος της τυχαίας ακτίνας r^d στην νοητή ευθεία του άξονα τιμών της μεταβλητής απόφασης x^d , δηλαδή της μεταβλητής απόφασης x για την d διάσταση του προβλήματος.



Εικόνα 4.4: Παράδειγμα υπολογισμού του βήματος $movement^d$

Μια μείωση της μέγιστης γωνίας περιστροφής a° (maximum rotation angle), οδηγεί σε μικρότερες περιστροφές και κατ' επέκταση σε περισσότερες παραγόμενες πιθανές λύσεις σε κάθε διάσταση.

Ένα σημαντικό στοιχείο παραμετροποίησης που προέκυψε μέσω των εφαρμογών του αλγόριθμου σε διάφορα πεδία ήταν η ανάγκη να μειωθεί ο υπολογιστικός χρόνος (computational time) χωρίς να εξαιρεθεί η καινοτομία της παραγωγής πολλών πιθανών λύσεων ανά επανάληψη. Στη διπλωματική εργασία του Κωνσταντίνου (2018) δείχθηκε ότι ελάττωση της γωνίας περιστροφής σε πολύ μικρές τιμές δεν είχε πάντα τα επιθυμητά αποτελέσματα, ενώ παράλληλα σπαταλούσε ο αλγόριθμος πολύτιμο υπολογιστικό χρόνο ψάχνοντας σε περιοχές του χώρου λύσεων που δεν έδειχνε να υπάρχει τοπικό βέλτιστο. Έτσι, αντί να κρατάει ο αλγόριθμος σταθερή την μέγιστη γωνία περιστροφής α° για όλους τους αναζητητές, τους ταξινομεί βάσει τιμής αντικειμενικής συνάρτησης και τους χωρίζει σε υπο-ομάδες (Tzanetos and Dounias, 2018; Tzanetos et al., 2018). Κάθε υπο-ομάδα χρησιμοποιεί την αντίστοιχη τιμή μέγιστης γωνίας περιστροφής α° από το διάνυσμα:

$$\vec{a}^\circ = [50 \ 40 \ 30 \ 20 \ 10 \ 5] \quad (4.3.10)$$

όπου οι τιμές του διανύσματος έχουν προκύψει μετά από εκτεταμένα πειράματα και προτείνονται ως προκαθορισμένες τιμές του εν λόγω διανύσματος.

Με αυτό τον μηχανισμό, κάθε αναζητητής ελέγχει περισσότερες πιθανές λύσεις σε κάθε επανάληψη του αλγορίθμου, ενώ στους άλλους αλγορίθμους ελέγχεται ένα σημείο ανά επανάληψη. Την ίδια στιγμή, ο κάθε αναζητητής ελέγχει το πλήθος πιθανών λύσεων που αντιστοιχεί στην σημαντικότητα της λύσης που εξετάζει σε εκείνη την επανάληψη: χειρότερη τιμή αντικειμενικής συνάρτησης οδηγεί δεξιότερα στο διάνυσμα μέγιστης γωνίας περιστροφής α° , όπου λαμβάνει μικρότερη τιμή μέγιστης γωνίας περιστροφής και άρα εξετάζει περισσότερα σημεία. Αυτό το γεγονός αυξάνει τις πιθανότητες να μεταπηδήσει ο αναζητητής σε καλύτερη λύση.

Η νέα θέση υπολογίζεται ως:

$$x_i^d_{new} = movement^d + x_i^d_{old} \quad (4.3.11)$$

όπου x_i^d είναι η τιμή του i αναζητητή στην d διάσταση και $movement^d$ είναι το d -οστό στοιχείο του διανύσματος που προκύπτει από τη σχέση (4.3.9). Σε κάθε βήμα της περιστροφής, η τιμή της αντικειμενικής συνάρτησης της νέας θέσης υπολογίζεται και εάν είναι καλύτερη από την ήδη καλύτερη που έχει βρει ο συγκεκριμένος αναζητητής, τότε ενημερώνεται η καλύτερη θέση και η τιμή της αντικειμενικής συνάρτησης που έχει.

4.3.4 Μηχανισμός Διόρθωσης Λύσεων (Correction mechanisms)

Πολλά προβλήματα χαρακτηρίζονται από περιορισμούς στις μεταβλητές απόφασης. Οι αλγόριθμοι δημιουργούν συχνά μη εφικτές (non-feasible) λύσεις μετά από κάποια βήματα. Για να αποφύγει ο αλγόριθμος τα κομμάτια του χώρου με τις μη εφικτές λύσεις και να γλιτώσει υπολογιστική ισχύ, εισήχθηκε ο παρακάτω

μηχανισμός διόρθωσης λύσεων. Εάν κάποιο x_i^d παραβιάσει τους περιορισμούς σε κάποιο βήμα του αλγορίθμου, επανατοποθετείται στον χώρο μέσω της σχέσης:

$$x_{i_{new}}^d = lower_bound^d + (upper_bound^d - lower_bound^d) \cdot \cos(x_{i_{old}}^d) \quad (4.3.12)$$

ώστε να ικανοποιείται η σχέση:

$$lower_bound^d \leq x_{i_{new}}^d \leq upper_bound^d \quad (4.3.13)$$

Ο ίδιος μηχανισμός χρησιμοποιείται και για την ενεργό ακτίνα r_0 . Εάν η τιμή της ενεργού ακτίνας r_0 κάποιου αναζητητή σε μια διάσταση d ξεπεράσει το $accept_range^d$ που αναφέρθηκε στην υποενότητα 4.3.1, τότε δημιουργείται μια νέα ενεργός ακτίνα με την ίδια λογική.

4.4 Παρατηρήσεις

Στόχος του 4^{ου} Κεφαλαίου ήταν να προταθεί ένας νέος Αλγόριθμος που είναι Εμπνευσμένος από τη Φύση, ο οποίος να επιλύει προβλήματα βελτιστοποίησης με υψηλή πολυπλοκότητα. Στο παρόν κεφάλαιο της διατριβής επεξηγείται λεπτομερώς ο προτεινόμενος αλγόριθμος, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH). Παρακάτω αναλύονται τα σημαντικότερα χαρακτηριστικά της BEH και επεξηγείται η πολυπλοκότητά της.

4.4.1 Εξερεύνηση και Εκμετάλλευση της BEH

Ένα χαρακτηριστικό των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση είναι η έλξη μεταξύ των αναζητητών, η οποία ωστόσο πολλές φορές δημιουργεί πρόωρη σύγκλιση (premature convergence). Για την αντιμετώπιση αυτού του φαινομένου έχει ενσωματωθεί ο μηχανισμός επανατοποθέτησης λύσεων (βήματα 4 έως 7 του Ψευδοκώδικα 4.1), ο οποίος στοχεύει στην διασπορά των αναζητητών στον χώρο λύσεων και η αποφυγή συγκέντρωσης γύρω από τοπικά βέλτιστα, τα οποία μπορεί να έχουν προσεγγίσει κάποιοι αναζητητές.

Ο παραπάνω μηχανισμός στοχεύει να προσφέρει υψηλό ποσοστό εξερεύνησης (exploration) στον προτεινόμενο αλγόριθμο. Παρόλα αυτά, ο τρόπος που δουλεύει στοχεύει να υπάρχει κι ένα ποσοστό εκμετάλλευσης (exploitation). Ανάλογα με το αν ξεπερνάει ο επανατοποθετούμενος αναζητητής την μέση ποιότητα (quality) των λύσεων σε όρους αντικειμενικής συνάρτησης, η επανατοποθέτηση γίνεται είτε τυχαία είτε με βάση την καλύτερη λύση που έχει βρεθεί. Για να επιτευχθεί ακόμη καλύτερη ισορροπία εξερεύνησης και εκμετάλλευσης, θα μπορούσε να αναπτυχθεί μια τροποποίηση της BEH η οποία να μειώνει σταδιακά την πιθανότητα επανατοποθέτησης βάσει της καλύτερης λύσης και να αυξάνεται η αντίστοιχη πιθανότητα τυχαίας επανατοποθέτησης.

Παράλληλα, η Σάρωση Πλήρους Κύκλου εστιάζει στην εκμετάλλευση (exploitation), καθώς ο μηχανισμός αυτός αναζητά άλλες πιθανές λύσεις στον χώρο γύρω από εκάστοτε λύση. Αυτή η αναζήτηση προσθέτει πολυπλοκότητα στον αλγόριθμο. Για να μειωθεί ο χρόνος εκτέλεσης της ΒΕΗ δημιουργήθηκε ένα διάλυμα γωνιών περιστροφής, το οποίο αντιστοιχεί τις μικρότερες γωνίες στις καλύτερες λύσεις και τις μεγαλύτερες στις χειρότερες, με σκοπό να μην σπαταλάει ο προτεινόμενος αλγόριθμος υπολογιστικό χρόνο και ισχύ σε λύσεις που δεν δείχνουν υποσχόμενες.

4.4.2 Πολυπλοκότητα χρόνου προτεινόμενης μεθόδου

Πίνακας 4.1: Απαιτούμενος χρόνος εκτέλεσης κάθε βήματος της ΒΕΗ

Βήμα	Απαιτούμενος χρόνος $T(n)$
1: Αρχικοποίηση αναζητητών	$N \times d$
2: Αρχικοποίηση Ενεργού Ακτίνας και Έντασης V αναζητητή	N
3: Όσο κριτήριο τερματισμού δεν ικανοποιείται	T
4: Για κάθε αναζητητή	N
5: Εάν μετρητής επαναλήψεων χωρίς καλύτερη λύση = στιγμή επανατοποθέτησης	1
6: Επανατοποθέτηση αναζητητή	$(T/k) \times d$
7: Επανυπολογισμός Έντασης και Ενεργού Ακτίνας	(T/k)
8: Τέλος_εάν	-
9: Ενημέρωση Ακτίνας	1
10: Υπολογισμός Έντασης	1
11: Όσο Σάρωση Πλήρους Κύκλου δεν έχει ολοκληρωθεί	$(360^\circ / \text{min } a^\circ)$
12: Ενημέρωση της Γωνίας Περιστροφής V διάσταση	d
13: Υπολογισμός της Αντικειμενικής Συνάρτησης για την πιθανή νέα θέση	1
14: Αποθήκευση της καλύτερης τιμής Αντικειμενικής Συνάρτησης για τα μέχρι τώρα βήματα της Σάρωσης	1
15: Τέλος	1
16: Ενημέρωση καλύτερης λύσης και Αντικειμενικής Συνάρτησης	1
17: Ενημέρωση Έντασης και Σήματος Εκπομπής	1
18: Τέλος	1
19: Τέλος	1

Σχετικά με την πολυπλοκότητα χρόνου του προτεινόμενου αλγορίθμου, είναι σαφές ότι η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) απαιτεί σχετικά περισσότερο χρόνο από άλλες μεθόδους που είναι Εμπνευσμένες από τη Φύση, καθώς ο μηχανισμός της Σάρωσης Πλήρους Κύκλου προσθέτει πολυπλοκότητα όντας ένας μηχανισμός τοπικής αναζήτησης.

Αναλυτικότερα, αν συμβολίσουμε με N το πλήθος των αναζητητών, με d τις διαστάσεις του προβλήματος (μεταβλητές απόφασης), με a° την μέγιστη γωνία περιστροφής, με T τον αριθμό των επαναλήψεων (το κριτήριο τερματισμού του αλγορίθμου), με k τις μέγιστες επαναλήψεις ως την στιγμή επανατοποθέτησης (checkpoint), ο υπολογισμός του απαιτούμενου χρόνου εκτέλεσης κάθε βήματος της BEH παρουσιάζεται στον Πίνακα 4.1.

Η αρχικοποίηση των αναζητητών είναι πολυπλοκότητας $O(N \times d)$, λόγω του ότι χρειάζεται να δημιουργηθεί μια τυχαία τιμή για κάθε μεταβλητή απόφασης σε κάθε αναζητητή. Η αρχικοποίηση των παραμέτρων του φυσικού αναλόγου του αλγορίθμου είναι $O(N)$ πολυπλοκότητας, διότι το χειρότερο σενάριο απαιτεί την αρχικοποίηση π.χ. της Έντασης (διεργασία $O(1)$ πολυπλοκότητας) για κάθε αναζητητή, δηλαδή N φορές.

Τα επόμενα βήματα επαναλαμβάνονται κατά μέγιστο T φορές, που είναι ο μέγιστος αριθμός επαναλήψεων. Επιπλέον, σε κάθε επανάληψη, όλες οι διεργασίες εκτελούνται για κάθε αναζητητή (N φορές), οπότε η πολυπλοκότητα της επαναληπτικής διαδικασίας είναι $O(T) \times O(N)$.

Ο μηχανισμός επανατοποθέτησης λύσεων αποτελείται από τον έλεγχο του αν έχει φτάσει ο αναζητητής τον μέγιστο αριθμό επαναλήψεων χωρίς βελτίωση (πολυπλοκότητας $O(1)$), την επανατοποθέτηση του αναζητητή (πολυπλοκότητας $O\left(\left(\frac{T}{k}\right) \times d\right)$) και τον επανυπολογισμό των παραμέτρων του αναζητητή που επανατοποθετήθηκε (πολυπλοκότητας $O\left(\frac{T}{k}\right)$), αφού άλλαξε η θέση του στο χώρο και κατ' επέκταση δημιουργήθηκε μια νέα λύση. Λόγω της ασυμπτωτικής ανάλυσης, η πολυπλοκότητα του μηχανισμού επανατοποθέτησης υπολογίζεται ως η μέγιστη πολυπλοκότητα μεταξύ της επανατοποθέτησης του αναζητητή και του επανυπολογισμού των παραμέτρων του αναζητητή που επανατοποθετήθηκε:

$$f_{\text{μηχανισμού επανατοποθέτησης}}(d, T, k) = \max\left(O\left(\left(\frac{T}{k}\right) \times d\right), O\left(\frac{T}{k}\right)\right) \quad (4.4.1)$$

Ο υπολογισμός της Ενεργού Ακτίνας και της Έντασης είναι δύο διεργασίες πολυπλοκότητας $O(1)$. Η πολυπλοκότητα χρόνου της Σάρωσης Πλήρους Κύκλου (Full Scan Loop) μπορεί να υπολογισθεί από το μέγιστο πηλίκο των 360° με κάθε τιμή μέγιστης γωνίας περιστροφής από το διάνυσμα a° :

$$f_{\text{Σάρωσης Πλήρους Κύκλου}}(a^\circ) = \max\left(O\left(\frac{360^\circ}{a_1^\circ}\right), O\left(\frac{360^\circ}{a_2^\circ}\right), \dots, O\left(\frac{360^\circ}{a_6^\circ}\right)\right) \quad (4.4.2)$$

όπου το μέγιστο πηλίκο ορίζεται από την ελάχιστη τιμή του παρονομαστή ή αλλιώς, την ελάχιστη τιμή του διανύσματος μέγιστων γωνιών περιστροφής a° . Άρα η σχέση (4.4.2) γίνεται:

$$f_{\text{Σάρωσης Πλήρους Κύκλου}}(a^\circ) = O\left(\frac{360^\circ}{\min a^\circ}\right) \quad (4.4.3)$$

Εντός της Σάρωσης Πλήρους Κύκλου εκτελούνται τρεις διεργασίες: (α) η ενημέρωση της γωνίας περιστροφής σε κάθε διάσταση, (β) ο υπολογισμός της αντικειμενικής συνάρτησης για την πιθανή νέα θέση και (γ) η αποθήκευση της καλύτερης τιμής αντικειμενικής συνάρτησης για τα μέχρι τώρα βήματα της Σάρωσης. Η διεργασία (α) είναι πολυπλοκότητας $O(d)$, ενώ οι διεργασίες (β) και (γ) είναι πολυπλοκότητας χρόνου $O(1)$. Αν σε αυτά συνυπολογίσουμε ότι στον έλεγχο «Όσο» γίνονται $w+1$ επαναλήψεις, όπου w είναι οι φορές που ικανοποιείται ο έλεγχος και άλλη μια φορά θα γίνει ο έλεγχος χωρίς να ικανοποιηθεί, η σχέση (4.4.3) εξελίσσεται σε:

$$f_{\text{Σάρωσης Πλήρους Κύκλου}}(a^\circ, d) = \left[O\left(\frac{360^\circ}{\min a^\circ}\right) \times (O(d) + O(1) + O(1)) \right] + O(1) \quad (4.4.4)$$

Τα βήματα 16 έως 19 είναι όλα υπολογιστικού χρόνου $O(1)$. Οπότε, σύμφωνα με όλα τα παραπάνω, η πολυπλοκότητα του χρόνου εκτέλεσης της ΒΕΗ εκτιμάται ως:

$$\begin{aligned} f(N, d, T, k, a^\circ) &= O(N \times d) + O(N) + O(T) \times O(N) \\ &\times \left(O(1) \times \max\left(O\left(\frac{T}{k}\right) \times d, O\left(\frac{T}{k}\right)\right) + O\left(\frac{360^\circ}{\min a^\circ}\right) \right) \\ &\times (O(d) + O(1) + O(1)) + O(1) + O(1) + O(1) \end{aligned} \quad (4.4.5)$$

Ακολούθως, η σχέση (4.4.5) γίνεται:

$$\begin{aligned} f(N, d, T, k, a^\circ) &= O(N \times d) + O(N) + O(T \times N) \\ &\times \left(O(1) \times O\left(\frac{T}{k}\right) \times d + O\left(d \times \frac{360^\circ}{\min a^\circ}\right) + O\left(2 \times \frac{360^\circ}{\min a^\circ}\right) + O(3) \right) \end{aligned} \quad (4.4.6)$$

και:

$$\begin{aligned}
f(N, d, T, k, a^\circ) = & O(N \times d) + O(N) + O\left(N \times \left(\frac{T^2}{k}\right) \times d\right) + O\left(T \times N \times d \times \frac{360^\circ}{\min a^\circ}\right) \\
& + O\left(2 \times T \times N \times \frac{360^\circ}{\min a^\circ}\right) + O(3 \times T \times N)
\end{aligned} \tag{4.4.7}$$

Στην παραπάνω σχέση είναι προφανές ότι ο γρηγορότερος όρος είναι ο $O(N)$. Επιπλέον, δεδομένου ότι συνήθως το πλήθος των μεταβλητών απόφασης είναι μικρότερο σε σύγκριση με τον μέγιστο αριθμό των επαναλήψεων ενός Αλγορίθμου Εμπνευσμένου από τη Φύση, ο όρος $O(N \times d)$ είναι γρηγορότερος από τον όρο $O(3 \times T \times N)$. Οπότε η πολυπλοκότητα χρόνου της ΒΕΗ δίνεται από έναν εκ των τριών υπόλοιπων όρων της σχέσης (4.4.7). Επιπλέον, ισχύει ότι:

$$\frac{360^\circ}{\min a^\circ} > 3 \tag{4.4.8}$$

καθώς θα έπρεπε η ελάχιστη τιμή του διανύσματος μέγιστης γωνίας περιστροφής a° θα έπρεπε να έχει τιμή μεγαλύτερη ή ίση των 120° . Κατ' επέκταση, ο όρος $O(3 \times T \times N)$ είναι γρηγορότερος εκ των δύο που περιέχουν την παράμετρο της γωνίας a° .

Τα προβλήματα που καλούνται να λύσουν οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση είναι υψηλής πολυπλοκότητας, οπότε δεν θα είχε νόημα να μιλάμε για πρόβλημα με μεταβλητές απόφασης 2 και λιγότερες, οπότε ισχύει $d > 2$. Με βάση αυτή την παραδοχή, ο γρηγορότερος μεταξύ των δύο όρων που περιέχουν την παράμετρο της γωνίας a° είναι ο $O\left(2 \times T \times N \times \frac{360^\circ}{\min a^\circ}\right)$.

Στη σύγκριση μεταξύ των δύο άλλων όρων παρατηρείται ότι το γινόμενο $T \times N \times d$ είναι κοινό και στους δύο. Κατ' επέκταση, η σύγκριση θα βασιστεί στους λόγους T/k και $360^\circ/\min a^\circ$. Όπως φαίνεται στην ενότητα 4.3.3, η μικρότερη τιμή της γωνίας a° στο διάστημα της σχέσης (4.3.10) είναι πέντε (5). Οπότε ο όρος $O\left(N \times \left(\frac{T^2}{k}\right) \times d\right)$ θα είναι γρηγορότερος του $O\left(T \times N \times d \times \frac{360^\circ}{\min a^\circ}\right)$ εφόσον:

$$T/k < 360^\circ/5^\circ \Rightarrow \tag{4.4.9}$$

$$T/k < 72 \Rightarrow \tag{4.4.10}$$

$$T < 72 \cdot k \tag{4.4.11}$$

Βάσει της (4.2.6) οι μέγιστες επαναλήψεις ως την στιγμή επανατοποθέτησης (checkpoint) είναι ίσες με $k = \mu \cdot T$. Άρα:

$$T < 72 \cdot \mu \cdot T \tag{4.4.12}$$

$$\frac{T}{T} < 72 \cdot \mu \tag{4.4.13}$$

$$\frac{1}{72} < \mu \quad (4.4.14)$$

Επίσης, βάσει της ίδιας σχέσης (4.2.6), ο όρος $O\left(N \times \left(\frac{T^2}{k}\right) \times d\right)$ μπορεί να γραφτεί:

$$O\left(N \times \left(\frac{T^2}{k}\right) \times d\right) = O\left(N \times \left(\frac{T^2}{\mu \cdot T}\right) \times d\right) = O\left(N \times \left(\frac{T}{\mu}\right) \times d\right) \quad (4.4.15)$$

Οπότε για $\mu \geq 0.014$ η πολυπλοκότητα του αλγορίθμου BEH είναι:

$$f(N, d, T, \mu) = O\left(N \times \left(\frac{T}{\mu}\right) \times d\right) \quad (4.4.16)$$

διαφορετικά είναι:

$$f(N, d, T, a^\circ) = O\left(T \times N \times d \times \frac{360^\circ}{\min a^\circ}\right) \quad (4.4.17)$$

4.4.3 Ανοικτές παράμετροι και αυτο-ρύθμιση BEH

Κάθε στοχαστικός αλγόριθμος έχει δύο βασικές ανοικτές παραμέτρους: τον πληθυσμό (population) και τις επαναλήψεις (generations). Τα πειράματα που γίνονται, συνήθως, αποσκοπούν στην εύρεση του κατάλληλου συνδυασμού πληθυσμού και επαναλήψεων για την επίλυση του εκάστοτε προβλήματος. Η πολυπλοκότητα του προβλήματος καθορίζει κατά κύριο λόγο το μέγεθος των δύο αυτών παραμέτρων.

Η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) έχει μια επιπλέον ανοικτή παράμετρο, τη στιγμή επανατοποθέτησης (checkpoint). Βασιζόμενοι στη βιβλιογραφία, μπορούμε να καθορίσουμε τη στιγμή επανατοποθέτησης ως ένα ποσοστό των συνολικών επαναλήψεων, το οποίο κυμαίνεται περίπου στο 10%. Ωστόσο, περαιτέρω έρευνα θα μπορούσε να διεξαχθεί, ώστε να δομηθεί ένας τρόπος υπολογισμού της κατάλληλης στιγμής επανατοποθέτησης βάσει της πορείας του εκάστοτε αναζητητή (agent) στον χώρο. Μια τέτοια θεωρία θα μπορούσε να στηριχθεί στον πίνακα απολαβών του Γκίτινς (Gittins, 1979; Gittins and Jones, 1979).

Μια ακόμη παράμετρος, που στην αρχική μορφή της προτεινόμενης μεθόδου ήταν ανοικτή παράμετρος, είναι η μέγιστη γωνία περιστροφής a° . Για την μείωση της πολυπλοκότητας και την ελαχιστοποίηση της απαιτούμενης παραμετροποίησης, δημιουργήθηκε ένα διάνυμα \bar{a}° μέγιστων γωνιών περιστροφής, το οποίο αναθέτει γωνίες περιστροφής σε κάθε αναζητητή (agent) σύμφωνα με την ποιότητας της λύσης του στην εκάστοτε επανάληψη. Με αυτό τον τρόπο, δεν εκτελούν όλοι οι αναζητητές μικρά (που απαιτεί υψηλό υπολογιστικό χρόνο) ή μεγάλα (κίνδυνος για χαμηλή

εκμετάλλευση) βήματα, αλλά ο κάθε ένας συμβάλει ξεχωριστά στην αναζήτηση ή στην εκμετάλλευση της προτεινόμενης μεθόδου.

4.4.4 Είδη προβλημάτων που λύνει η ΒΕΗ

Η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (ΒΕΗ) αποδεικνύεται καλή μέθοδος επίλυσης προβλημάτων βελτιστοποίησης με μεταβλητές απόφασης συνεχούς χώρου, όπως η Οικονομική Κατανομή Φορτίου (Economic Load Dispatch) και η εύρεση του βέλτιστου συνδυασμού (σε ποσοστά) επενδυόμενου κεφαλαίου σε ένα χαρτοφυλάκιο k μετοχών.

Για τα διακριτά προβλήματα, όπως το Πρόβλημα του Πλανόδιου Πωλητή (TSP), βρίσκεται υπό εξέλιξη έρευνα για την αποδοτική προσέγγιση τους από την προτεινόμενη μέθοδο της παρούσας διατριβής. Ωστόσο, με τις κατάλληλες τροποποιήσεις, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (ΒΕΗ) επέλυσε επιτυχώς προβλήματα Κατανομής Πόρων (Resource Leveling).

5. Υβριδικοί Αλγόριθμοι

5.1 Ανάπτυξη Υβριδικών Σχημάτων

Η σύμπραξη δύο ή περισσότερων μεθόδων με σκοπό την επίλυση ενός προβλήματος ονομάζεται υβριδικό σχήμα (hybrid scheme) ή υβριδική μέθοδος (hybrid method). Στη βιβλιογραφία εμφανίζονται πολύ συχνά υβριδικές μέθοδοι που αποτελούνται από Αλγόριθμους Εμπνευσμένους από τη Φύση (Gaitanidis et al., 2016; Naderi et al., 2017; X. Bao, H. Jia and C. Lang, 2019).

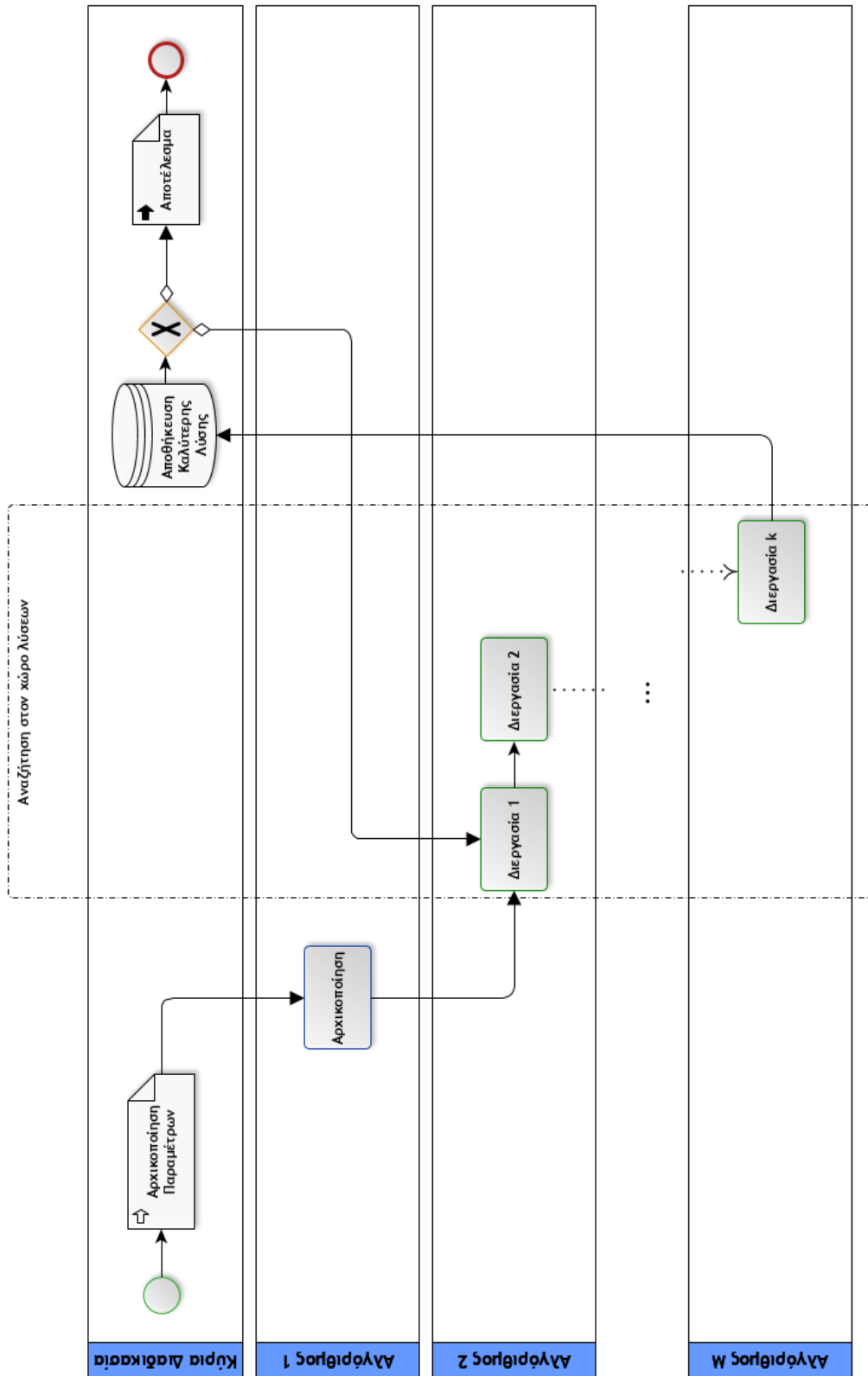
Η βασική φιλοσοφία πίσω από την ανάπτυξη υβριδικών σχημάτων είναι κάθε μέθοδος που απαρτίζει το υβριδικό σχήμα να υποκαθιστά τις αδυναμίες της/των άλλης/υπολοίπων μεθόδου/-ων. Ωστόσο, σε μερικές περιπτώσεις, κάθε μέθοδος καλείται να επιλύσει ένα συγκεκριμένο μέρος (στόχο) του προβλήματος στο οποίο εφαρμόζεται το υβριδικό σχήμα. Παρακάτω, αναλύονται τα συνηθέστερα υβριδικά μοντέλα της βιβλιογραφίας.

5.1.1 Υβριδικά Μοντέλα συνεργασίας μεταξύ των επιμέρους μεθόδων

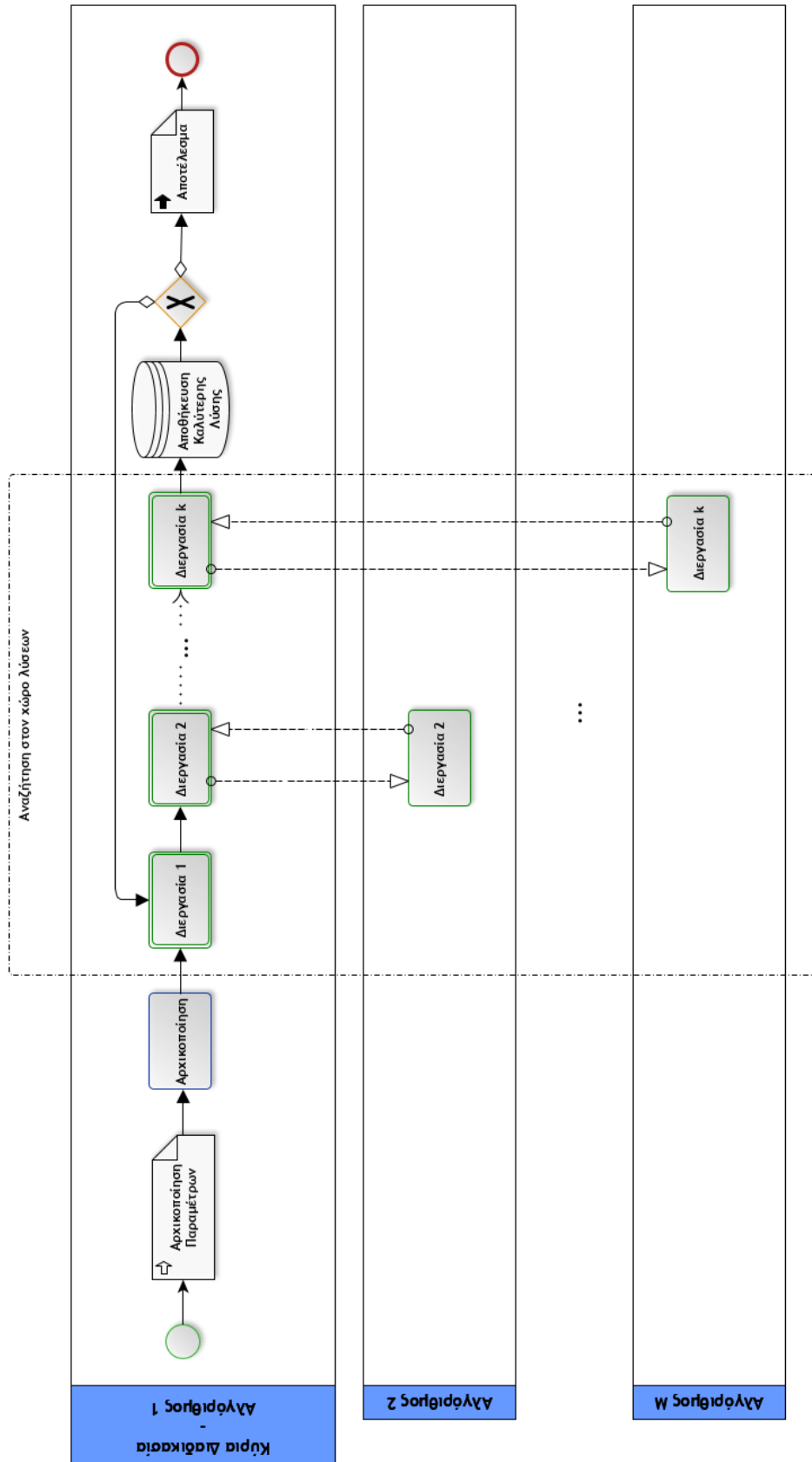
Πολύ συχνά, ξεχωριστές υπολογιστικές μέθοδοι καλούνται να συμβάλλουν εξίσου στην επίλυση ενός προβλήματος βελτιστοποίησης. Στις περισσότερες περιπτώσεις, το σημείο στο οποίο συνεργάζονται είναι η αναζήτηση στον χώρο λύσεων, σύμφωνα και με το Σχήμα 3-4 του Κεφαλαίου 3, όπου διακρίνονται οι ξεχωριστές φάσεις ενός Αλγορίθμου Εμπνευσμένου από τη Φύση.

Στο Σχήμα 5-1 παρακάτω, απεικονίζεται το μοντέλο όπου οι επιμέρους μέθοδοι εκτελούν διαφορετικές διεργασίες στο στάδιο της αναζήτησης στον χώρο λύσεων. Για παράδειγμα, οι F. Grimaccia, M. Mussetta and R. E. Zich (2007) παρουσίασαν ένα υβριδικό σχήμα που απαρτίζεται από τον Γενετικό Αλγόριθμο (ΓΑ) και τη Βελτιστοποίηση Σμήνους Σωματιδίων (ΒΣΣ). Σε αυτό το υβριδικό σχήμα, αρχικά ο ΓΑ χωρίζει τον πληθυσμό σε δύο υπο-ομάδες και ύστερα εκτελεί τα βήματα της διασταύρωσης και της μετάλλαξης στη μία υπο-ομάδα, ενώ η ΒΣΣ μετακινεί σε νέες θέσεις τις λύσεις της δεύτερης υπο-μάδας.

Μια άλλη περίπτωση είναι το μοντέλο που απεικονίζεται στο Σχήμα 5-2, όπου ένας εκ των αλγορίθμων είναι η κύρια μέθοδος και ο άλλος ή οι άλλοι αλγόριθμος/-οι εκτελούν συγκεκριμένες διεργασίες της κύριας μεθόδου. Οι Ravikumar Pandi and Panigrahi (2011) ανέπτυξαν ένα υβριδικό σχήμα με τον αλγόριθμο Αναζήτησης Αρμονίας (Harmony Search) και τη Βελτιστοποίηση Σμήνους Σωματιδίων (ΒΣΣ). Η φάση της αναζήτησης στον χώρο λύσεων της Αναζήτησης Αρμονίας (ΑΑ) γίνεται με μια εκ των εξής περιπτώσεων: της Αρμονίας από Μνήμης (Harmony Memory Consideration), τη Ρύθμιση Τόνου (Pitch Adjustment) και τον Αυτοσχεδιασμό (Improvisation). Στο παραπάνω υβριδικό σχήμα, ο ρόλος της ΒΣΣ είναι να γίνει η Ρύθμιση Τόνου με τις εξισώσεις μετακίνησης της ΒΣΣ, που παρουσιάστηκαν και στο Κεφάλαιο 3 στη σχέση (3.6.2).

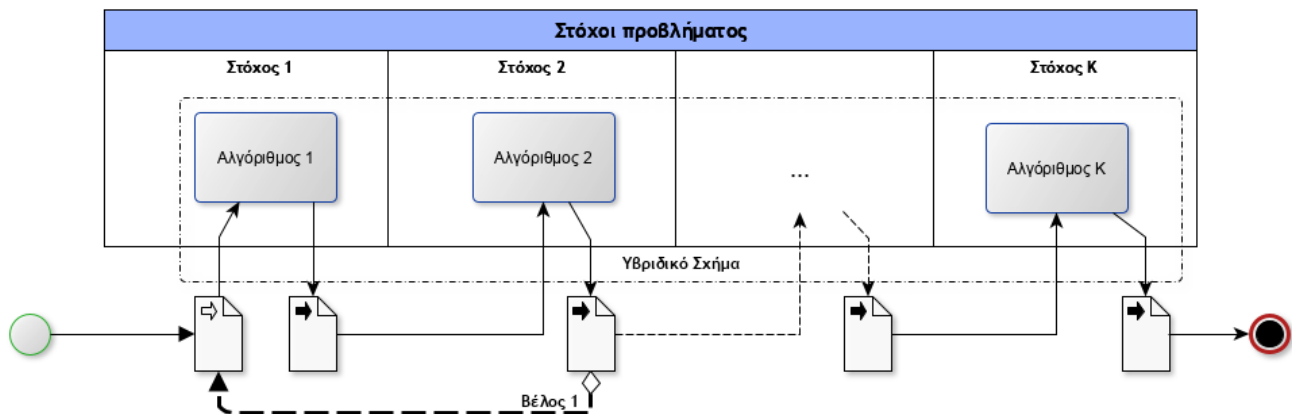


Σχήμα 5-1: Υβριδικό μοντέλο όπου οι επιμέρους αλγόριθμοι εκτελούν διαφορετικές διεργασίες



Σχήμα 5-2: Υβριδικό μοντέλο όπου οι επιμέρους αλγόριθμοι εκτελούν διεργασίες του κύριου αλγόριθμου

Σπανιότερα, συναντάται στη βιβλιογραφία η περίπτωση όπου ένας αλγόριθμος αναλαμβάνει το σημείο της αρχικοποίησης και ένας ή περισσότεροι αναλαμβάνουν την αναζήτηση στον χώρο λύσεων (Maaranen, Miettinen and Mäkelä, 2004; Rahnamayan, Tizhoosh and Salama, 2007). Αυτή η περίπτωση παρατηρείται όταν κάποια μέθοδος (συνήθως η απλούστερη μεταξύ των μεθόδων που απαρτίζουν το υβριδικό σχήμα) καλείται να τροφοδοτήσει την άλλη ή τις άλλες μεθόδους με υποδιαίρεσεις του χώρου λύσεων.



Σχήμα 5-3: Υβριδικό μοντέλο επίλυσης πολλών στόχων

Μια ακόμη περίπτωση όπου συνεργάζονται διαφορετικοί αλγόριθμοι για την επίλυση ενός προβλήματος είναι όταν το πρόβλημα είναι πολλών στόχων (multi-objective) και αναλαμβάνει κάθε αλγόριθμος ξεχωριστά να επιλύσει έναν εκ των στόχων (Giannakouris, Vassiliadis and Dounias, 2010). Η συγκεκριμένη περίπτωση φαίνεται στο Σχήμα 5-3, όπου κάθε αλγόριθμος λαμβάνει σαν είσοδο την έξοδο της προηγούμενης μεθόδου. Επιπλέον, υπάρχει και το φαινόμενο της ανάδρασης, όπου μια μέθοδος μπορεί να ανατροφοδοτήσει την προηγούμενη με κάποια δεδομένα, όπως δείχνει το Βέλος 1 του παραπάνω σχήματος.

5.1.2 Υβριδικά Μοντέλα βελτιστοποίησης άλλων μεθόδων Υπολογιστικής Νοημοσύνης

Όπως αναφέρεται και στην Ενότητα 3.7.1, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση είναι αρκετά χρήσιμοι στο να βρουν τις κατάλληλες τιμές των παραμέτρων άλλων μεθόδων Υπολογιστικής Νοημοσύνης (Computational Intelligent methods), όπως τα Τεχνητά Νευρωνικά Δίκτυα και ο Γενετικός Προγραμματισμός. Για παράδειγμα, οι Aljarah, Faris and Mirjalili (2018) χρησιμοποιούν τον Αλγόριθμο Βελτιστοποίησης Φαλαινών για να ρυθμίσουν κατάλληλα τα βάρη (weights) ενός Νευρωνικού Δικτύου Πολυστρωματικών Αντίληπτρων (Multilayer Perceptron Neural Network). Γενικότερα, η εκπαίδευση (training) ενός Τεχνητού Νευρωνικού Δικτύου ή παρόμοιων μεθόδων Υπολογιστικής Νοημοσύνης είναι αρκετά δύσκολη και απαιτητική. Με τη χρήση των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση, εφόσον το πρόβλημα βελτιστοποίησης είναι η

εύρεση του βέλτιστου σετ παραμέτρων της σχετικής μεθόδου που είναι προς εκπαίδευση, η διαδικασία αυτή απλοποιείται.

Επιπλέον, σε αρκετές περιπτώσεις θα μπορούσαν να τοποθετηθούν μετά από τις προαναφερόμενες μεθόδους Υπολογιστικής Νοημοσύνης, με σκοπό να βελτιώσουν περαιτέρω τη λύση τους.

5.2 Υβριδικά Μοντέλα που περιέχουν τη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό

Στην παρούσα διατριβή αναπτύχθηκαν και προτείνονται τρία υβριδικά σχήματα, τα οποία περιέχουν την Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (ΒΕΗ): το υβριδικό σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό – Προσομοιωμένης Ανόπτωσης (ΒΕΗ-ΠΑ), το υβριδικό σχήμα Γενετικού Αλγορίθμου – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΑ-ΒΕΗ) και το υβριδικό σχήμα Γενετικού Προγραμματισμού – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΠ-ΒΕΗ). Παρακάτω, γίνεται επιμέρους ανάλυση και παρουσίαση κάθε υβριδικού σχήματος.

5.2.1 Υβριδικό Σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό – Προσομοιωμένης Ανόπτωσης

Ο μηχανισμός επανατοποθέτησης που περιγράφεται στο Κεφάλαιο 4, αποτελεί μια μέθοδο επίτευξης μιας ισορροπίας μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation). Στον εν λόγω μηχανισμό, οι λύσεις που δεν βελτιώνουν τη θέση τους για κάποιες διαδοχικές επαναλήψεις, επανατοποθετούνται κάπου στον χώρο λύσεων με σκοπό να αποφευχθεί το φαινόμενο του εγκλωβισμού σε τοπικά βέλτιστα (local optimum trapping).

Ένας άλλος τρόπος επίτευξης ισορροπίας μεταξύ εξερεύνησης και εκμετάλλευσης είναι η σταδιακή μεταβολή από υψηλό ποσοστό εξερεύνησης σε υψηλό ποσοστό εκμετάλλευσης. Δηλαδή, στη διάρκεια των πρώτων επαναλήψεων της αλγοριθμικής διαδικασίας δίνεται έμφαση στην αναζήτηση στον χώρο και όσο τείνει να ολοκληρωθεί η διαδικασία δίνεται έμφαση στην τοπική αναζήτηση (local search) γύρω από τα τοπικά βέλτιστα που έχουν βρεθεί. Ένας μηχανισμός που το επιτυγχάνει αυτό αποτελεί μέρος της Προσομοιωμένης Ανόπτωσης (Simulated Annealing).

Ο μηχανισμός αυτός ονομάζεται κριτήριο Metropolis (Metropolis Criterion) και χρησιμοποιείται για να γίνονται αποδεκτές λύσεις χειρότερης ποιότητας με πιθανότητα p , έτσι ώστε να επιτυγχάνεται μεγαλύτερη διαφοροποίηση στις λύσεις. Η πιθανότητα αυτή υπολογίζεται ως:

$$p = e^{-\Delta q/T} \quad (5.2.1.1)$$

όπου με Δq συμβολίζεται η διαφορά μεταξύ της ποιότητας (σε όρους αντικειμενικής συνάρτησης) της ήδη υπάρχουσας λύσης και της ποιότητας της νέας λύσης και T είναι η παράμετρος της θερμοκρασίας της ΠΑ, η οποία μειώνεται σταδιακά κατά τη διάρκεια του αλγορίθμου βάσει της σχέσης:

$$T_k = T_{k-1} \cdot r_T \quad (5.2.1.2)$$

όπου r_T είναι ο ρυθμός μείωσης της θερμοκρασίας T σε κάθε επανάληψη k .

Η Προσομοιωμένη Ανόπτηση στηρίζεται στην ανάπτυξη μιας λύσης (trajectory-based) και βασίζεται στο φαινόμενο της μεταλλουργίας, την ανόπτηση, κατά την οποία γίνεται ελεγχόμενη ψύξη ενός μετάλλου που προηγουμένως είχε θερμανθεί σε πολλή υψηλή θερμοκρασία, ώστε να αυξηθεί το μήκος των κρυσταλλικών δομών του με αποτέλεσμα να γίνει πιο λεία η επιφάνειά του.

Ψευδοκώδικας 5.1: Προσομοιωμένη Ανόπτηση (Simulated Annealing)

- 1: Αρχικοποίηση παραμέτρων
 - 2: Αρχικοποίηση λύσης
 - 3: **Όσο** θερμοκρασία $\neq 0$
 - 4: Επιλογή μιας νέας θέσης
 - 5: Υπολογισμός ποιότητας της νέας θέσης
 - 6: Υπολογισμός πιθανότητας κριτηρίου Metropolis
 - 7: **Εάν** η ποιότητας της νέας θέσης είναι καλύτερη από την ποιότητας της παλιάς θέσης
 - 8: Αποδοχή λύσης
 - 9: **Αλλιώς_εάν** $rand \leq$ Πιθανότητα Κριτηρίου Metropolis
 - 10: Αποδοχή λύσης
 - 11: **Τέλος**
 - 12: Ενημέρωση καλύτερης λύσης
 - 13: Μείωση θερμοκρασίας
 - 14: **Τέλος**
-

Όπως φαίνεται στον Ψευδοκώδικας 5.1 παραπάνω, εάν η ποιότητα της νέας λύσης είναι καλύτερη από την ήδη υπάρχουσα, γίνεται αποδεκτή η νέα. Διαφορετικά, εάν η ποιότητά της είναι χειρότερη από την ήδη υπάρχουσα, εφαρμόζεται το κριτήριο Metropolis. Για να γίνει αποδεκτή σε αυτή την περίπτωση η νέα λύση, πρέπει ένας τυχαίος αριθμός $rand$ να είναι μικρότερος από την πιθανότητα του κριτηρίου Metropolis. Στην περίπτωση ενός προβλήματος ελαχιστοποίησης, χειρότερη ποιότητα σημαίνει:

$$q_{old} < q_{new} \quad (5.2.1.3)$$

όπου με q_{old} συμβολίζεται η ποιότητα της υπάρχουσας λύσης και με q_{new} η ποιότητα της νέας λύσης. Εφόσον, όπως ορίζεται παραπάνω:

$$\Delta q = q_{old} - q_{new} \quad (5.2.1.4)$$

και

$$(5.2.1.3) \Rightarrow q_{old} - q_{new} < 0 \quad (5.2.1.5)$$

Ψευδοκώδικας 5.2: Υβριδικό Σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization) με Προσομοιωμένη Ανόπτηση (Simulated Annealing)

- 1: Αρχικοποίηση αναζητητών
 - 2: Αρχικοποίηση Ενεργού Ακτίνας και Έντασης κάθε αναζητητή
 - 3: **Όσο** κριτήριο τερματισμού δεν ικανοποιείται
 - 4: **Εάν** μετρητής επαναλήψεων χωρίς καλύτερη λύση = στιγμή επανατοποθέτησης
 - 5: **Εάν** $rand \leq$ Πιθανότητα Κριτηρίου Metropolis
 - 6: Τυχαία Επανατοποθέτηση αναζητητή
 - 7: **Αλλιώς**
 - 8: Επανατοποθέτηση αναζητητή με βάση την ήδη γνωστή καλύτερη θέση
 - 9: **Τέλος**
 - 10: Επανυπολογισμός Έντασης και Ενεργού Ακτίνας
 - 11: **Τέλος_εάν**
 - 12: Ενημέρωση Ακτίνας για κάθε αναζητητή
 - 13: Υπολογισμός Έντασης για κάθε αναζητητή
 - 14: **Όσο** Σάρωση Πλήρους Κύκλου δεν έχει ολοκληρωθεί
 - 15: Ενημέρωση της Γωνίας Περιστροφής για κάθε διάσταση
 - 16: Υπολογισμός της Αντικειμενικής Συνάρτησης για την πιθανή νέα θέση
 - 17: Αποθήκευση της καλύτερης τιμής Αντικειμενικής Συνάρτησης για τα μέχρι τώρα βήματα της Σάρωσης
 - 18: **Τέλος**
 - 19: Ενημέρωση καλύτερης λύσης και Αντικειμενικής Συνάρτησης
 - 20: Ενημέρωση Έντασης και Σήματος Εκπομπής για κάθε αναζητητή
 - 21: **Τέλος**
-

όσο μεγαλύτερη είναι η παραπάνω διαφορά, τόσο αυξάνεται και η τιμή του κλάσματος της σχέσης (5.2.1.1). Επιπλέον, όσο μειώνεται η θερμοκρασία T , τόσο αυξάνεται η τιμή του εν λόγω κλάσματος. Από τις ιδιότητες της $y = e^{-x}$, ισχύει ότι η αύξηση της τιμής του x επιφέρει μείωση του y και άρα σύμφωνα με τη σχέση (5.2.1.1) μειώνεται η πιθανότητα του κριτηρίου Metropolis. Συνοψίζοντας, όσο χειρότερη είναι

η νέα λύση ή όσο εξελίσσεται ο αλγόριθμος (και άρα μειώνεται η παράμετρος της θερμοκρασίας), τόσο δυσκολότερο είναι να γίνει αποδεκτή μια νέα λύση με χειρότερη ποιότητα από την ήδη υπάρχουσα.

Στο προτεινόμενο υβριδικό σχήμα, η πιθανότητα του κριτηρίου Metropolis καθορίζει αν κατά την επανατοποθέτηση του αναζητητή θα δημιουργηθεί μια τυχαία νέα θέση ή αν θα επανατοποθετηθεί ο αναζητητής στον χώρο γύρω από την ήδη γνωστή καλύτερη λύση. Η τυχαία επανατοποθέτηση ενισχύει την εξερεύνηση, ενώ η επανατοποθέτηση με βάση την ήδη γνωστή καλύτερη λύση ενισχύει την εκμετάλλευση. Οπότε, στην αρχή της υβριδικής μεθόδου είναι πιο πιθανό να επανατοποθετηθεί τυχαία ο αναζητητής και άρα υπάρχει υψηλό ποσοστό εξερεύνησης, ενώ σταδιακά αυξάνεται η πιθανότητα να επανατοποθετηθεί γύρω από την καλύτερη γνωστή λύση και άρα αυξάνεται το ποσοστό της εκμετάλλευσης, ενώ μειώνεται αυτό της εξερεύνησης.

Για να διερευνηθεί εάν βελτιώνεται η απόδοση της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (BEH) μέσα από το υβριδικό σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό – Προσομοιωμένης Ανόπτησης (BEH-ΠΑ), εφαρμόστηκε το εν λόγω υβριδικό σχήμα στο πρόβλημα της Κατανομής Πόρων, όπου και συγκρίνονται τα αποτελέσματά του με τη BEH.

5.2.2 Υβριδικό Σχήμα Γενετικού Αλγορίθμου – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό

Το υβριδικό σχήμα Γενετικού Αλγορίθμου – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΑ-BEH) χρησιμοποιήθηκε στην επίλυση ενός προβλήματος πολλών στόχων (multi-objective), πιο συγκεκριμένα στην Βελτιστοποίηση Χαρτοφυλακίου η οποία περιγράφεται σε επόμενο κεφάλαιο.

Οι στόχοι του προβλήματος είναι η εύρεση των μετοχών που θα απαρτίζουν το χαρτοφυλάκιο, δεδομένου ενός συνόλου μετοχών ενός χρηματιστηριακού δείκτη και η εύρεση του ποσού που θα επενδυθεί από το διαθέσιμο κεφάλαιο σε κάθε μετοχή του χαρτοφυλακίου. Το συγκεκριμένο πρόβλημα έχει λυθεί στο παρελθόν με άλλα υβριδικά σχήματα, όπως αυτό της Βελτιστοποίησης Αποικίας Μυρμηγκιών – Αλγόριθμου Βαρυτικής Αναζήτησης (BAM-ABA) (Tzanetos, Vassiliadis and Dounias, 2017) και Βελτιστοποίησης Αποικίας Μυρμηγκιών – Αλγόριθμου Λαμπυρίδων (BAM-ΑΛ) (Giannakouris, Vassiliadis and Dounias, 2010). Η Βελτιστοποίηση Αποικίας Μυρμηγκιών (BAM) επιλύει τον πρώτο στόχο, ενώ το δεύτερο μέρος κάθε υβριδικού σχήματος επιλύει τον δεύτερο στόχο, αντίστοιχα.

Ωστόσο, η BAM είναι μια αρκετά αυστηρή μέθοδος, η οποία συχνά συγκλίνει πολύ γρήγορα σε ένα τοπικό βέλτιστο (premature convergence). Για να αντιμετωπιστεί αυτή η δυσκολία, προτάθηκε η χρήση του Γενετικού Αλγορίθμου (ΓΑ) αντί της BAM, καθώς ο ΓΑ είναι πιο απλοϊκός (naïve) και δεν εγκλωβίζεται σε τοπικά βέλτιστα.

Ο Γενετικός Αλγόριθμος (Holland, 1992b) βασίζεται στον τρόπο εξέλιξης των ειδών, όπου οι απόγονοι έχουν γενετική πληροφορία και από τους δύο γονείς, ενώ ένα πολύ μικρό ποσοστό του πληθυσμού μπορεί να υποστεί μετάλλαξη. Υποθέτοντας ότι αρχικά δημιουργείται ένας τυχαίος πληθυσμός:

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}$$

όπου x_{ij} είναι η τιμή της j μεταβλητής απόφασης της λύσης του i αναζητητή για ένα πρόβλημα με k μεταβλητές απόφασης και με n συμβολίζεται ο πληθυσμός του ΓΑ. Επιλέγοντας δύο τυχαίες λύσεις από τον πληθυσμό:

Γονέας 1	x_{i1}	x_{i2}	x_{i3}	...	x_{ik-1}	x_{ik}
Γονέας 2	$x_{i'1}$	$x_{i'2}$	$x_{i'3}$...	$x_{i'k-1}$	$x_{i'k}$

Η διασταύρωση μεταξύ αυτών των λύσεων θα γίνει σε ένα τυχαίο σημείο και θα έχει ως αποτέλεσμα να δημιουργηθούν δύο νέες λύσεις:

Παιδί 1	x_{i1}	x_{i2}	x_{i3}	...	$x_{i'r}$...	$x_{i'k-1}$	$x_{i'k}$
Παιδί 2	$x_{i'1}$	$x_{i'2}$	$x_{i'3}$...	x_{ir}	...	x_{ik-1}	x_{ik}

Οι αρχικές λύσεις ονομάζονται γονείς (parent solutions) και εκείνες που είναι αποτέλεσμα της διασταύρωσης ονομάζονται παιδιά (children solutions).

Κατά τη μετάλλαξη, μια ή περισσότερες τιμές των μεταβλητών απόφασης μιας λύσης μπορεί να μεταβληθούν, λαμβάνοντας μια τυχαία τιμή.

Στο υβριδικό σχήμα ΓΑ-BEH που αναπτύχθηκε και προτείνεται στα πλαίσια της παρούσας διατριβής, η επιλογή των γονέων γίνεται με τη μέθοδο του Ελιτισμού (elitism), δηλαδή επιλέγονται οι καλύτερες λύσεις μεταξύ του πληθυσμού, βάσει της ποιότητάς τους. Οι λύσεις του ΓΑ στο υβριδικό σχήμα αντιπροσωπεύουν τα χαρτοφυλάκια και δίνονται ως είσοδος στη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH). Για κάθε χαρτοφυλάκιο, η BEH αναζητά τη βέλτιστη κατανομή του διαθέσιμου κεφαλαίου και επιστρέφει το αποτέλεσμα στον ΓΑ. Το αποτέλεσμα αυτό λαμβάνεται υπόψη στη δημιουργία των νέων χαρτοφυλακίων.

Επιπλέον, στο παραπάνω υβριδικό σχήμα έχουν προστεθεί δύο ακόμη βήματα. Για να διασφαλιστεί ότι η τυχειότητα του ΓΑ δε θα δημιουργεί δύο ή περισσότερα ίδια χαρτοφυλάκια σε κάθε πληθυσμό, έχει προστεθεί ένας σχετικός έλεγχος στην αρχή κάθε γενιάς. Επίσης, στο τέλος κάθε γενιάς ελέγχεται η διαφοροποίηση του πληθυσμού σε σχέση με την προηγούμενη γενιά. Δηλαδή απαριθμούνται τα χαρτοφυλάκια που δεν εμφανίζονται στην προηγούμενη γενιά, με σκοπό να ενισχυθεί η εξερεύνηση του χώρου.

Ψευδοκώδικας 5.3: Υβριδικό Σχήμα Γενετικού Αλγορίθμου - Βελτιστοποίησης
 Εμπνευσμένης από τον Ηχοεντοπισμό (Sonar Inspired Optimization)

- 1: Αρχικοποίηση παραμέτρων
 - 2: **Για** 1: σύνολο γενεών
 - 3: Δημιουργία Πληθυσμού
 - 4: Έλεγχος Μοναδικότητας Χαρτοφυλακίων
 - 5: **Για** 1: σύνολο πληθυσμού
 - 6: Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό ()
 - 7: **Τέλος**
 - 8: Ενημέρωση καλύτερου χαρτοφυλακίου
 - 9: Υπολογισμός Ρυθμού Μετάλλαξης
 - 10: Επιλογή μέσω Ελιτισμού
 - 11: Διασταύρωση Πληθυσμού
 - 12: Μετάλλαξη
 - 13: Μέτρηση διαφοροποίησης πληθυσμού από την προηγούμενη γενιά
 - 14: **Τέλος**
-

Όπως γίνεται αντιληπτό, αυτό το υβριδικό σχήμα είναι εκθετικού χρόνου ($O(2^n)$), καθώς για σε κάθε γενιά για κάθε λύση του πληθυσμού του ΓΑ, η ΒΕΗ εκτελείται για κάποιο αριθμό επαναλήψεων έχοντας έναν πληθυσμό λύσεων. Γενικότερα, τα υβριδικά σχήματα είναι σαφώς πιο χρονοβόρα σε σχέση με τις επιμέρους μεθόδους που τα απαρτίζουν. Ωστόσο, είναι καλύτερα από πλευράς απόδοσης, καθώς ενσωματώνουν νοημοσύνη από πολλές μεθόδους και οι αδυναμίες κάθε μεθόδου καλύπτονται από τις υπόλοιπες.

5.2.3 Υβριδικό Σχήμα Γενετικού Προγραμματισμού – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό

Όπως αναλύεται και στα προηγούμενα Κεφάλαια, οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση εφαρμόζονται κυρίως σε προβλήματα βελτιστοποίησης. Στα προβλήματα ταξινόμησης, κατηγοριοποίησης και πρόβλεψης είναι πιο αποδοτικές άλλες μέθοδοι Υπολογιστικής Νοημοσύνης και ειδικότερα οι τεχνικές Μηχανικής Μάθησης. Ωστόσο, ενώ υπερτερούν στην ανάπτυξη μοντέλων που θα περιγράφουν τη σχέση μεταξύ κάποιων μεταβλητών απόφασης, σε αρκετές περιπτώσεις είναι απαραίτητη η σωστή ρύθμιση των παραμέτρων τους (parameter tuning). Επίσης, όντας μέθοδοι “μαύρου κουτιού” (black box) και άρα μη γνωρίζοντας τις εσωτερικές διεργασίες που λαμβάνουν μέρος κατά την ανάπτυξη ενός μοντέλου, είναι δύσκολο να βελτιωθεί η απόδοσή τους.

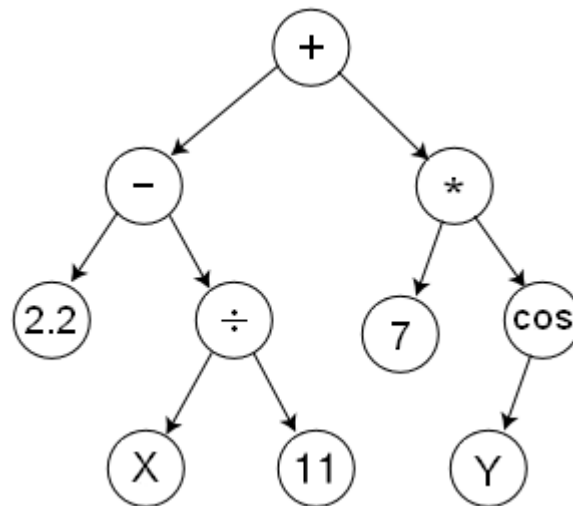
Το υβριδικό σχήμα Γενετικού Προγραμματισμού – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΠ-ΒΕΗ) αποσκοπεί στην περαιτέρω βελτίωση του μοντέλου που παράγει ο Γενετικός Προγραμματισμός (ΓΠ) και

ειδικότερα στο να βρεθούν οι βέλτιστες τιμές των συντελεστών του παραγόμενου μοντέλου.

Ο Γενετικός Προγραμματισμός (ΓΠ) προτάθηκε από τον (Koza, 1992), ο οποίος στηρίχτηκε στις αρχές του Γενετικού Αλγόριθμου. Είναι μια μέθοδος Μηχανικής Μάθησης που αναπτύσσει μοντέλα που περιγράφουν τη σχέση μεταξύ των δεδομένων ενός προβλήματος. Οι λύσεις του ΓΠ αναπαρίστανται με δενδρικές δομές, ένα χαρακτηριστικό που τον καθιστά ιδιαίτερα χρήσιμο στην ανάπτυξη μοντέλων που κατανοεί ο λήπτης απόφασης. Για παράδειγμα, η παρακάτω μαθηματική σχέση:

$$\left(2.2 - \frac{X}{11}\right) + (7 \cdot \cos Y) \quad (5.2.3.1)$$

αποτυπώνεται με δενδρική δομή ως ακολούθως:



Σχήμα 5-4: Δενδρική Δομή Γενετικού Προγραμματισμού

Τα X και Y αντιπροσωπεύουν τις μεταβλητές απόφασης του προβλήματος, ενώ οι αριθμοί είναι οι σταθεροί όροι (ή συντελεστές) του μοντέλου. Εφόσον έχει αναπτυχθεί το μοντέλο (η μαθηματική σχέση) που περιγράφει ένα σύστημα, η εύρεση των βέλτιστων τιμών των συντελεστών αυτών αποτελεί ένα ξεχωριστό πρόβλημα βελτιστοποίησης.

Η ύπαρξη των παραμέτρων της Έντασης και της Ενεργού Ακτίνας της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (BEH) προσφέρουν υψηλή απόδοση στην αναζήτηση στον χώρο λύσεων, καθώς η προτεινόμενη μέθοδος μπορεί να εστιάζει σε βάθος πολλών δεκαδικών στ κάθε μεταβλητή απόφασης.

Κατ' επέκταση, η BEH χρησιμοποιείται στο προτεινόμενο υβριδικό σχήμα για την εύρεση των βέλτιστων τιμών των συντελεστών του μοντέλου που παράγει ο ΓΠ.

5.3 Συμπεράσματα

Σκοπός αυτού του κεφαλαίου ήταν να εισάγει τον αναγνώστη στη λειτουργία και την επιχειρηματολογία γύρω από την συνεργασία της προτεινόμενης μεθόδου με άλλες μεθόδους Υπολογιστικής Νοημοσύνης με απώτερο σκοπό την περαιτέρω βελτίωση των λύσεων που παράγονται, καθώς επίσης να διερευνήσει πώς θα επηρεαστεί η απόδοση του προτεινόμενου αλγόριθμου εάν αναπτυχθούν υβριδικές μέθοδοι, στις οποίες θα συνεργαστεί με άλλους Αλγόριθμους Εμπνευσμένους από τη Φύση ή άλλες μεθόδους Υπολογιστικής Νοημοσύνης.

Αρχικά, γίνεται μια αναφορά στους υβριδικούς αλγόριθμους και γενικότερα στις υβριδικές μεθόδους Υπολογιστικής Νοημοσύνης, όπου αναλύονται τα μοντέλα υβριδικών μεθόδων που συναντώνται συχνότερα στη βιβλιογραφία. Σε όλες τις περιπτώσεις που αναλύονται, είναι σαφές ότι ο υπολογιστικός χρόνος εκτέλεσης των υβριδικών μεθόδων είναι κατά πολύ μεγαλύτερος του αντίστοιχου μιας απλής μεθόδου.

Τρία υβριδικά σχήματα αναπτύχθηκαν και προτείνονται στην παρούσα διατριβή, τα οποία περιέχουν την Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH).

Το υβριδικό σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό – Προσομοιωμένης Ανόπτησης (BEH-ΠΑ) ενσωματώνει το κριτήριο Metropolis της ΠΑ στην προτεινόμενη μέθοδο (BEH) με σκοπό να δημιουργηθεί μια αυτορρυθμιζόμενη σταδιακή εναλλαγή από υψηλή εξερεύνηση (exploration) σε υψηλή εκμετάλλευση (exploitation),

Το υβριδικό σχήμα Γενετικού Αλγορίθμου – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΑ-BEH) αναπτύχθηκε για την επίλυση της Βελτιστοποίησης Χαρτοφυλακίου που αποτελεί πρόβλημα πολλών στόχων (multi-objective problem). Η απλοϊκότητα (naiveness) του ΓΑ προσφέρει μεγαλύτερη εξερεύνηση του χώρου λύσεων στο υβριδικό σχήμα σε σχέση με παλαιότερα υβριδικά σχήματα που αναπτύχθηκαν για την επίλυση του παραπάνω προβλήματος.

Η υψηλή απόδοση της BEH στην αναζήτηση στον χώρο λύσεων λόγω των παραμέτρων της Ενεργού Ακτίνας και της Έντασης, την καθιστούν χρήσιμη στην βελτίωση των τιμών που θα έχουν οι σταθεροί όροι ενός μοντέλου που αναπτύσσεται από μια άλλη μέθοδο Υπολογιστικής Νοημοσύνης. Το υβριδικό σχήμα Γενετικού Προγραμματισμού – Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΠ-BEH) συνθέτει τη δυνατότητα ανάπτυξης μοντέλων που περιγράφουν ένα σύστημα του ΓΠ με την υψηλή απόδοση στην βελτιστοποίηση μιας σειράς μεταβλητών απόφασης της BEH, που στην προκειμένη περίπτωση είναι οι συντελεστές του προηγούμενου μοντέλου.

6. Γενικές Εφαρμογές

Όλα τα πειράματα για τις εφαρμογές που αναφέρονται παρακάτω έγιναν σε προγραμματιστικό περιβάλλον Matlab σε υπολογιστές με μνήμη 4GB, επεξεργαστή 3.6GHz Intel Core i7 και λειτουργικό Windows 10 Pro. Σε όλα τα προβλήματα που παρουσιάζονται παρακάτω, ο αριθμός των ανεξάρτητων πειραμάτων που εκτελέστηκαν, είναι ίδιος με τον αριθμό των αντίστοιχων πειραμάτων που έχουν γίνει από τις συγκρινόμενες μεθόδους της βιβλιογραφίας και χρησιμοποιείται για να υπάρχει μια βάση σύγκρισης.

6.1 Συναρτήσεις Ελέγχου Αποδοτικότητας

Για τον έλεγχο λειτουργικότητας και αποδοτικότητας των νοημόνων μεθόδων έχουν προταθεί μια σειρά από συναρτήσεις ελέγχου (benchmark functions). Κατά την παρουσίασή της (Tzanetos and Dounias, 2017a), η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό δοκιμάστηκε με επιτυχία σε επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας.

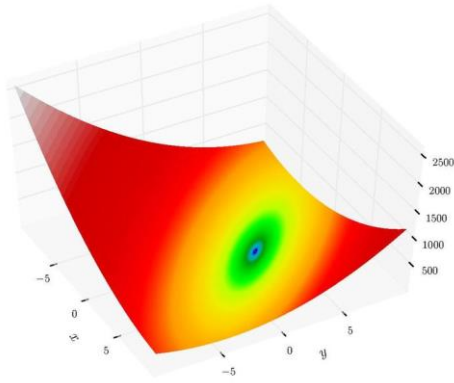
Κατά τη διάρκεια της παρούσας διατριβής βελτιώθηκε σταδιακά η προτεινόμενη μέθοδος, οπότε παρακάτω έχουν γίνει επαναληπτικά πειράματα σε επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας, χρησιμοποιώντας την πιο πρόσφατη μορφή του αλγορίθμου, η οποία διατίθεται στον ιστότοπο της MathWorks ανεπτυγμένη σε προγραμματιστικό περιβάλλον Matlab².

6.1.1 Επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας

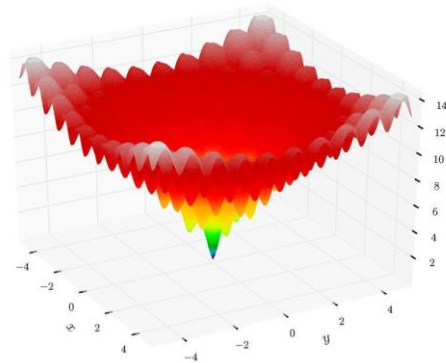
Γενικά, διάφορες συναρτήσεις ελέγχου αποδοτικότητας (benchmark function) έχουν προταθεί με σκοπό τη διερεύνηση της στατιστικής απόδοσης και την καταλληλότητα των μεθόδων βελτιστοποίησης (Jamil and Yang, 2013; Hellwig and Beyer, 2019). Κάθε συνάρτηση έχει κάποιες ιδιαιτερότητες που έχουν να κάνουν με το πλήθος των κορυφών (modes) που έχει ο χώρος λύσης τους, το αν είναι συνεχείς ή όχι, κυρτές κλπ.

Όταν ο χώρος λύσεων μιας συνάρτησης έχει ένα τοπικό βέλτιστο, η συνάρτηση ονομάζεται μονοκόρυφη (unimodal), ενώ σε αντίθετη περίπτωση λέγεται πολλών κορυφών (multimodal). Ένα παράδειγμα μονοκόρυφης συνάρτησης (unimodal) είναι η συνάρτηση Booth (Εικόνα 6.1α), ενώ ένα αντίστοιχο παράδειγμα συνάρτησης πολλών κορυφών (multimodal) είναι η συνάρτηση N.4 του Ackley (Εικόνα 6.1β).

² <https://www.mathworks.com/matlabcentral/fileexchange/64302-sonar-inspired-optimization-sio-generalized-version>



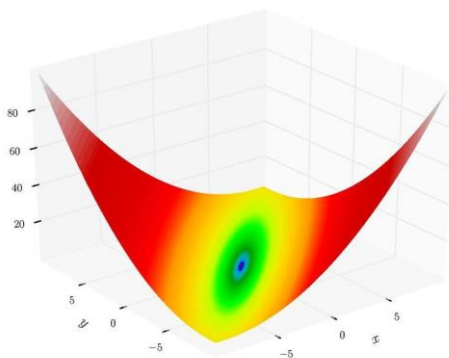
(α) Συνάρτηση Booth (Booth function)



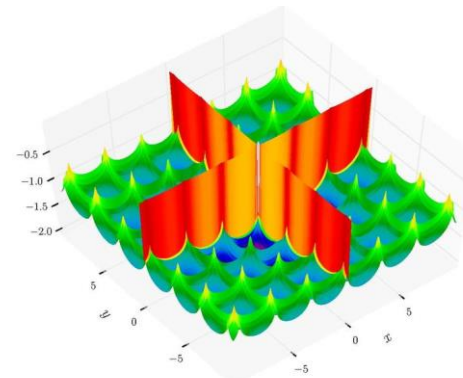
(β) Συνάρτηση N.4 του Ackley (Ackley's function N.4)

Εικόνα 6.1: Παραδείγματα χώρου λύσεων με ένα και πολλαπλά τοπικά βέλτιστα

Ένας άλλος διαχωρισμός που γίνεται είναι με βάση την κυρτότητα του χώρου λύσεων σε κυρτές (convex) και μη-κυρτές (non-convex) συναρτήσεις. Κυρτή συνάρτηση είναι η συνάρτηση Booth (Εικόνα 6.1α) και η συνάρτηση Matyas (Εικόνα 6.2α), ενώ μη-κυρτή είναι η συνάρτηση cross-in-tray (cross-in-tray function), όπως είναι εμφανές στην Εικόνα 6.2β.



(α) Συνάρτηση Matyas (Matyas function)



(β) Συνάρτηση cross-in-tray (Cross-in-tray function)

Εικόνα 6.2: Παραδείγματα χώρου λύσεων κυρτής (convex) και μη-κυρτής (non-convex) συνάρτησης

Επιπλέον, η δυσκολία επίλυσης μιας συνάρτησης εξαρτάται και από τις διαστάσεις της, δηλαδή το πλήθος των μεταβλητών απόφασης για τις οποίες ορίζεται, καθώς και από το πλήθος των αντικειμενικών συναρτήσεων που πρέπει να βελτιστοποιηθούν. Παρακάτω επιλέχθηκαν και παρουσιάζονται 11 επιλεγμένες συναρτήσεις ελέγχου αποτελεσματικότητας με σκοπό να ελεγχθεί η απόδοση της προτεινόμενης μεθόδου σε διάφορων ειδών συναρτήσεις.

Πίνακας 6.1: Ολικά βέλτιστα και πεδία ορισμού των συναρτήσεων ελέγχου αποδοτικότητας

Συνάρτηση	Ολικό βέλτιστο	Πεδίο τιμών
Ackley's function	$f_{min} = 0$	$-5 \leq x_i \leq 5$
Beale's function	$f_{min} = 0$	$-4.5 \leq x_1, x_2 \leq 4.5$
Booth's function	$f_{min} = 0$	$-10 \leq x_1, x_2 \leq 10$
Bukin function N.6	$f_{min} = 0$	$-15 \leq x_1 \leq 5$ $-3 \leq x_2 \leq 3$
Himmelblau's function	$f_{min} = 0$	$-6 \leq x_1, x_2 \leq 6$
Lévi function N.13	$f_{min} = 0$	$-10 \leq x_1, x_2 \leq 10$
Matyas function	$f_{min} = 0$	$-10 \leq x_1, x_2 \leq 10$
Rastrigin function	$f_{min} = 0$	$-5.12 \leq x_i \leq 5.12$
Schaffer function N. 2	$f_{min} = 0$	$-100 \leq x_1, x_2 \leq 100$
Schaffer function N. 4	$f_{min} = 0.292579$	$-100 \leq x_1, x_2 \leq 100$
Three-hump camel function	$f_{min} = 0$	$-5 \leq x_i \leq 5$

6.1.1.1 Συνάρτηση του Ackley (Ackley's Function)

Η συνάρτηση του Ackley προτάθηκε από τον David Ackley στη διατριβή του (Ackley, 1987) και είναι μια συνεχής (continuous), μη-κυρτή (non-convex) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal), η οποία ορίζεται σε n διαστάσεις (μεταβλητές απόφασης):

$$f(x_1, x_2, \dots, x_n) = -20 \cdot e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - \frac{1}{e} \sum_{i=1}^n \cos 2\pi x_i + e + 20 \quad (6.1.1)$$

όπου $x_i \in [-5,5]$, αν και γενικά μπορούν να δοθούν διάφορα διαστήματα ορισμού των μεταβλητών απόφασης. Στην παρούσα διατριβή ελέγχθηκε η αποδοτικότητα της προτεινόμενης μεθόδου για 2 διαστάσεις με το προαναφερθέν πεδίο ορισμού.

6.1.1.2 Συνάρτηση του Beale (Beale's function)

Η συνάρτηση του Beale ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), μη-κυρτή (non-convex) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal), όπως φαίνεται και στην Εικόνα 6.1α. Η σχέση που την περιγράφει είναι η:

$$f(x_1, x_2) = (1.5 - x_1 + x_1 \cdot x_2)^2 + (2.25 - x_1 + x_1 \cdot x_2^2)^2 + (2.625 - x_1 + x_1 \cdot x_2^3)^2 \quad (6.1.2)$$

όπου $x_1, x_2 \in [-4.5,4.5]$.

6.1.1.3 Συνάρτηση του Booth (Booth's function)

Η συνάρτηση του Booth ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), κυρτή (convex) συνάρτηση με ένα ολικό βέλτιστο (unimodal), όπως φαίνεται και στην Εικόνα 6.1α . Η σχέση που την περιγράφει είναι η:

$$f(x_1, x_2) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2 \quad (6.1.3)$$

όπου $x_1, x_2 \in [-10, 10]$.

6.1.1.4 Συνάρτηση N.6 του Bukin (Bukin function N.6)

Η συνάρτηση N.6 του Bukin (1997) ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), μη-διαφοροποιήσιμη (non-differentiable) και μη-διαχωρίσιμη (non-separable), κυρτή (convex) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal):

$$f(x_1, x_2) = 100\sqrt{|x_2 - 0.01 \cdot x_1^2|} + 0.01 \cdot |x_1 + 10| \quad (6.1.4)$$

όπου $x_1 \in [-15, 5]$ και $x_2 \in [-3, 3]$.

6.1.1.5 Συνάρτηση του Himmelblau (Himmelblau's function)

Η συνάρτηση του Himmelblau (1972) ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), μη-κυρτή (non-convex) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal) και περιγράφεται από τη σχέση:

$$f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 \quad (6.1.5)$$

όπου $x_1, x_2 \in [-6, 6]$.

6.1.1.6 Συνάρτηση N.13 του Lévi (Lévi function N.13)

Η συνάρτηση N.13 του Lévi ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), μη-κυρτή (non-convex) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal):

$$f(x_1, x_2) = \sin^2(3\pi x_1) + (x_1 - 1)^2(1 + \sin^2(3\pi x_2)) + (x_2 - 1)^2(1 + \sin^2(2\pi x_2)) \quad (6.1.6)$$

όπου $x_1, x_2 \in [-10, 10]$.

6.1.1.7 Συνάρτηση του Matyas (Matyas function)

Η συνάρτηση του Matyas ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), κυρτή (convex) συνάρτηση με ένα ολικό βέλτιστο (unimodal):

$$f(x_1, x_2) = 0.26(x_1^2 + x_2^2) - 0.48 \cdot x_1 \cdot x_2 \quad (6.1.7)$$

όπου $x_1, x_2 \in [-10, 10]$.

6.1.1.8 Συνάρτηση του Rastrigin (Rastrigin function)

Η συνάρτηση του Rastrigin αρχικά προτάθηκε από τον Leonard Andreevich Rastrigin (Rastrigin, 1974) ως συνάρτηση 2 διαστάσεων, ενώ αργότερα παρουσιάστηκε μια πιο γενικευμένη μορφή της από τον Rudolph (1990). Η συγκεκριμένη συνάρτηση είναι συνεχής (continuous), μη-κυρτή (non-convex), διαχωρίσιμη (separable) και διαφοροποιήσιμη (differentiable) συνάρτηση με πολλά τοπικά βέλτιστα (multimodal), η οποία ορίζεται σε n διαστάσεις (μεταβλητές απόφασης):

$$f(x_1, x_2, \dots, x_n) = 10 \cdot n + \sum_{i=1}^n [x_i^2 - 10 \cdot \cos 2\pi x_i] \quad (6.1.8)$$

όπου $x_i \in [-5.12, 5.12]$. Στην παρούσα διατριβή ελέγχθηκε η αποδοτικότητα της προτεινόμενης μεθόδου για 2, 5 και 10 διαστάσεις με το προαναφερθέν πεδίο ορισμού.

6.1.1.9 Συνάρτηση N.2 του Schaffer (Schaffer function N. 2)

Η συνάρτηση N.2 του Schaffer ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), κυρτή (convex) συνάρτηση με ένα ολικό βέλτιστο (unimodal):

$$f(x_1, x_2) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (6.1.9)$$

όπου $x_1, x_2 \in [-100, 100]$.

6.1.1.10 Συνάρτηση N.4 του Schaffer (Schaffer function N. 4)

Η συνάρτηση N.4 του Schaffer ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι, όπως και η N.2, μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), κυρτή (convex) συνάρτηση με ένα ολικό βέλτιστο (unimodal):

$$f(x_1, x_2) = 0.5 + \frac{\sin^2(\cos(|x_1^2 - x_2^2|)) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} \quad (6.1.10)$$

όπου $x_1, x_2 \in [-100, 100]$.

6.1.1.11 Συνάρτηση καμήλας με τρεις καμπούρες (Three-hump camel function)

Η συνάρτηση καμήλας με τρεις καμπούρες ορίζεται σε 2 διαστάσεις (μεταβλητές απόφασης) και είναι μια συνεχής (continuous), διαφοροποιήσιμη (differentiable), μη-διαχωρίσιμη (non-separable), μη-κυρτή (non-convex) συνάρτηση με ένα ολικό βέλτιστο (unimodal). Η σχέση που την περιγράφει είναι η:

$$f(x_1, x_2) = 2x_1^2 - 1.05 \cdot x_1^4 + \frac{x_1^6}{6} + x_1 \cdot x_2 + x_2^2 \quad (6.1.11)$$

όπου $x_1, x_2 \in [-5, 5]$.

6.1.2 Αποτελέσματα

Σε όλες τις συναρτήσεις έγιναν 50 ανεξάρτητα πειράματα, όπου χρησιμοποιήθηκαν 100 αναζητητές, έγιναν 100 επαναλήψεις, ο συντελεστής στιγμής επανατοποθέτησης είχε τιμή 0.01 και τέλος, το διάνυσμα τιμών μέγιστης γωνίας περιστροφής α° είχε τις τιμές που φαίνονται στη σχέση (4.3.10).

Πίνακας 6.2: Απόδοση της προτεινόμενης μεθόδου στις επιλεγμένες συναρτήσεις ελέγχου αποδοτικότητας

Συνάρτηση	Γνωστό βέλτιστο	Καλύτερη τιμή	Μέση τιμή	Τυπ. Απόκλ.
Ackley's function	0	2.9132e-13	2.7047e-08	1.1742e-07
Beale's function	0	5.0805e-11	0.0021	0.0128
Booth's function	0	9.5956e-11	7.5782e-04	0.0054
Bukin function N.6	0	0.0276	0.1879	0.1357
Himmelblau's function	0	1.5825e-10	7.8493e-07	3.6130e-06
Lévi function N.13	0	1.1865e-10	3.9080e-06	2.3906e-05
Matyas function	0	5.7713e-14	5.9857e-08	3.8526e-07
Rastrigin function ($n = 2$)	0	1.6162e-08	5.0777e-06	1.6570e-05
Rastrigin function ($n = 5$)	0	2.9970e-06	0.6343	1.5352
Rastrigin function ($n = 10$)	0	4.3734e-05	0.9979	2.2520
Schaffer function N. 2	0	0	4.2380e-12	1.8542e-11
Schaffer function N. 4	0.292579	0.2926	0.2929	3.8981e-04
Three-hump camel function	0	7.8450e-12	2.4032e-05	1.0423e-04

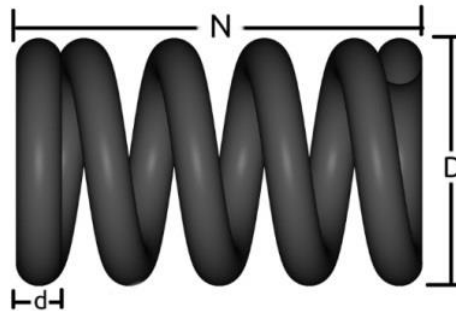
Σύμφωνα με τα αποτελέσματα του Πίνακα 6.2, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) κατάφερε να προσεγγίσει με μεγάλη ακρίβεια το γνωστό ολικό βέλτιστο κάθε συνάρτησης ελέγχου αποδοτικότητας.

6.2 Πρότυπα Προβλήματα Μηχανικού

Μια πρόκληση για τον προτεινόμενο αλγόριθμο ήταν η επίλυση πραγματικών προβλημάτων μηχανικού, που λόγω των περιορισμών που υπάρχουν στον φορμαλισμό τους, θεωρούνται Προβλήματα Βελτιστοποίησης υπό Περιορισμούς (Constrained Optimization Problems). Για τον σκοπό αυτό, επιλέχθηκαν δύο γνωστά προβλήματα βελτιστοποίησης από τον χώρο της Μηχανικής: ο Σχεδιασμός Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/compression Spring Design) και ο Σχεδιασμός Συγκολλημένης Δοκού (Welded Beam Design).

6.2.1 Σχεδιασμός Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/compression Spring Design)

Ο στόχος είναι η ελαχιστοποίηση του βάρους ενός ελατηρίου (Εικόνα 6.3), το οποίο υποβάλλεται σε εφελκυσμό/θλίψη (tension/compression spring). Περιορισμοί του προβλήματος είναι η διατμητική τάση λόγω στρέψης (shear stress), η συχνότητα κύματος ταλάντωσης (surge frequency) και η ελάχιστη παραμόρφωση (minimum deflection). Τρεις κατασκευαστικοί παράγοντες αποτελούν τις μεταβλητές απόφασης του προβλήματος: η διάμετρος του ελατηρίου (d), η μέση διάμετρος της σπείρας (D) και ο αριθμός των σπειρών (N).



Εικόνα 6.3: Σχεδιασμός Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/compression Spring Design)

Με την παραδοχή ότι η λύση είναι ένα διάνυσμα $\vec{x} = [x_1, x_2, x_3] = [d, D, N]$, ο μαθηματικός φορμαλισμός του προβλήματος έχει ως εξής:

Ελαχιστοποίηση της
δεδομένου ότι:

$$f(\vec{x}) = (x_3 + 2)x_2x_1^2$$

$$g_1(\vec{x}) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0$$

$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

και εύρος μεταβλητών:

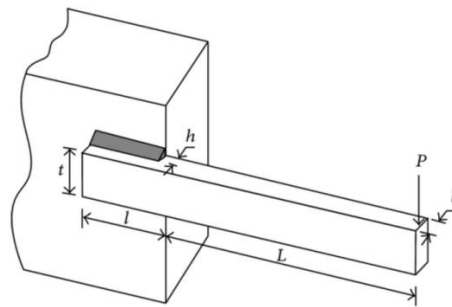
$$0.05 \leq x_1 \leq 2$$

$$0.25 \leq x_2 \leq 1.3$$

$$2 \leq x_3 \leq 15$$

6.2.2 Σχεδιασμός Συγκολλημένης Δοκού (Welded Beam Design)

Ο στόχος σε αυτό το πρόβλημα είναι η ελαχιστοποίηση του κατασκευαστικού κόστους (fabrication cost) μιας συγκολλημένης δοκού (welded beam), όπως αυτή φαίνεται σχεδιαστικά στην Εικόνα 6.4. Το εν λόγω πρόβλημα παρουσιάζει περιορισμούς: η διατμητική τάση λόγω στρέψης (shear stress) (τ), τάση κάμψης της δοκού (the bending stress in the beam) (θ), το φορτίο στην άκρη της δοκού (the buckling load on the bar) (P_c), το βέλος κάμψης στην άκρη της δοκού (the end deflection of the beam) (δ) και επιμέρους περιορισμούς, όπως περιγράφονται παρακάτω.



Εικόνα 6.4: Σχεδιασμός Συγκολλημένης Δοκού (Welded Beam Design)

Οι μεταβλητές απόφασης του προβλήματος είναι το πάχος συγκόλλησης (thickness of the weld) (h), το μήκος του συγκολλημένου μέρους (length of the attached part) (l), το ύψος της δοκού (height) (t) και το πάχος (thickness) της δοκού (b). Υποθέτοντας ότι η λύση είναι ένα διάνυσμα $\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$, το πρόβλημα βελτιστοποίησης μπορεί να περιγραφεί ως:

Ελαχιστοποίηση της
δεδομένου ότι:

$$f(\vec{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{max} \leq 0$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{max} \leq 0$$

$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{max} \leq 0$$

$$g_4(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_5(\vec{x}) = P - P_c(\vec{x}) \leq 0$$

$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$

όπου

$$g_7(\vec{x}) = 1.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \leq 0$$

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{MR}{J}, M = P(L + \frac{x_2}{2})$$

$$R = \sqrt{\frac{x_2^2}{2} + \left(\frac{x_1 + x_3}{2}\right)^2}$$

$$J = 2 \left\{ \sqrt{2}x_1x_2 \left[\frac{x_2^2}{2} + \left(\frac{x_1 + x_3}{2}\right)^2 \right] \right\}$$

$$\sigma(\vec{x}) = \frac{6PL}{x_4x_3^2}, \delta(\vec{x}) = \frac{6PL^3}{x_4x_3^2}$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2} \left(1 - \frac{x_3}{2L} \sqrt{\frac{E}{4G}} \right)$$

δεδομένου ότι

$$P = 6000 \text{ lb}, L = 14 \text{ in.}, E = 30 \times 10^6 \text{ psi}, G = 12 \times 10^6 \text{ psi}$$

$$\delta_{max} = 0.25 \text{ in.}, \tau_{max} = 13600 \text{ psi}, \sigma_{max} = 30000 \text{ psi}$$

Το εύρος των μεταβλητών δίνεται παρακάτω:

$$0.1 \leq x_1 \leq 2 \stackrel{g_6}{\Rightarrow} 0.125 \leq x_1 \leq 2$$

$$0.1 \leq x_2 \leq 10$$

$$0.1 \leq x_3 \leq 10$$

$$0.1 \leq x_4 \leq 2$$

6.2.3 Αποτελέσματα

Για κάθε πρόβλημα έγιναν 40 ανεξάρτητα τρεξίματα, ώστε να μετρηθεί και στατιστικά η απόδοση του αλγορίθμου. Τα αποτελέσματα συγκρίθηκαν με αυτά της βιβλιογραφίας, τα οποία προέρχονται από άλλους αλγόριθμους Εμπνευσμένους από τη Φύση, καθώς και ευρετικές μεθόδους. Στον Πίνακα 6.3 παρακάτω φαίνονται οι παράμετροι που χρησιμοποιήθηκαν στον αλγόριθμο για όλα ανεξαιρέτως τα πειράματα.

Πίνακας 6.3: Παράμετροι της BEH για όλα τα πειράματα

Παράμετρος	τιμή
Αριθμός επαναλήψεων (iterations):	1000
Πληθυσμός:	25
Ανεξάρτητα πειράματα:	40
s	0.008

Τα αποτελέσματα του αλγόριθμου για το πρόβλημα Σχεδιασμού του Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/Compression Spring Design) παρουσιάζονται στον Πίνακα 6.4, όπου ο στόχος του προβλήματος είναι η ελαχιστοποίηση του βάρους της κατασκευής και δίνεται στη στήλη «Βέλτιστο Βάρος». Τα αποτελέσματα αυτά συγκρίνονται με τα αντίστοιχα άλλων Αλγορίθμων Εμπνευσμένων από τη Φύση, όπως του Βελτιστοποιητή Γκρίζου Λύκου (Grey Wolf Optimizer - ΒΓΛ) (Mirjalili, Mirjalili and Lewis, 2014) και της συν-εξελικτικής Βελτιστοποίησης Σμήνους Σωματιδίων (co-evolutionary Particle Swarm Optimization - σΒΣΣ) (He and Wang, 2007).

Επιπλέον, τα αποτελέσματα των ευρετικών μεθόδων: Εξελικτική Στρατηγική (Evolutionary Strategy – ΕΣ) (Mezura-Montes and Coello, 2008), Γενετικός Αλγόριθμος (Genetic Algorithm - GA) (Coello Coello, 2000b), Αναζήτησης Αρμονίας (Harmony Search - AA) (Mahdavi, Fesanghary and Damangir, 2007), συν-εξελικτική Διαφορική Εξέλιξη (co-evolutionary Differential Evolution - σΔΕ) (Huang, Wang and He, 2007), αλλά και τα αντίστοιχα αποτελέσματα μαθηματικών προσεγγίσεων (Belegundu and Arora, 1985; Arora, 2016) χρησιμοποιήθηκαν ως σημεία σύγκρισης (benchmarks).

Η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (Sonar Inspired Optimization - SIO) κατάφερε να ξεπεράσει την απόδοση των άλλων αλγορίθμων και τα κατασκευαστικά σημεία (design points) που πρότεινε δεν παραβίαζαν τους περιορισμούς του προβλήματος.

Πίνακας 6.4: Σύγκριση αποτελεσμάτων για το πρόβλημα του Σχεδιασμού Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη

Αλγόριθμος / Μέθοδος	Βέλτιστη Λύση			Βέλτιστο Βάρος	Μέση Τιμή	Τυπική Απόκλιση
	d	D	L			
ΒΕΗ	0.051354	0.405225	7.554277	0.0102104	0.0116751	7.8400e-3
ΒΓΛ	0.05169	0.356737	11.28885	0.012666	N/A	N/A
σΒΣΣ	0.051728	0.357644	11.244543	0.0126747	0.012730	5.1985e-5
ΕΣ	0.051989	0.363965	10.890522	0.0126810	0.012923	5.9200e-4
ΓΑ	0.051480	0.351661	11.632201	0.0127048	0.012769	5.9000e-5
ΑΑ	0.051154	0.349871	12.076432	0.0126706	N/A	N/A
σΔΕ	0.051609	0.354714	11.410831	0.0126702	0.012703	2.7000e-5
Μαθηματική Βελτιστοποίηση	0.053396	0.399180	9.1854000	0.0127303	N/A	N/A
Constrained Correction	0.050000	0.315900	14.250000	0.0128334	N/A	N/A

Ο Πίνακας 6.5 περιέχει τα αποτελέσματα του αλγόριθμου για το πρόβλημα Σχεδιασμού της Συγκολλημένης Δοκού (Welded Beam Design), όπου ο στόχος του προβλήματος είναι η ελαχιστοποίηση του κόστους της κατασκευής και δίνεται στη στήλη «Βέλτιστο Κόστος». Για λόγους σύγκρισης υπάρχουν τα αποτελέσματα άλλων

αλγορίθμων, όπως του Βελτιστοποιητή Γκρίζου Λύκου (Grey Wolf Optimizer - GWO) (Mirjalili, Mirjalili and Lewis, 2014), του Γενετικού Αλγόριθμου (Genetic Algorithm - GA) (Deb, 1991, 2000; Coello Coello, 2000a), της Αναζήτησης Αρμονίας (Harmony Search - HS) (Lee and Geem, 2005), αλλά και μαθηματικών προσεγγίσεων (Ragsdell and Phillips, 1976).

Η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό ξεπέρασε σε απόδοση όλες τις άλλες μεθόδους εκτός του Βελτιστοποιητή Γκρίζου Λύκου, από την οποία βρέθηκε να είναι ελάχιστα χειρότερη. Ωστόσο, η απουσία στατιστικής ανάλυσης της συγκρινόμενης μεθόδου δεν επιτρέπει την περαιτέρω σύγκριση των αποτελεσμάτων.

Πίνακας 6.5: Σύγκριση αποτελεσμάτων για το πρόβλημα του Σχεδιασμού Συγκολλημένης Δοκού

Αλγόριθμος / Μέθοδος	Βέλτιστη Λύση				Βέλτιστο Κόστος	Μέση Τιμή	Τυπική Απόκλιση
	h	l	t	b			
BEH	0.331482	2.017442	9.045926	0.208865	1.7621466	2.075521	0.449052
ΒΓΛ	0.205676	3.478377	9.03681	0.205777	1.72624	N/A	N/A
ΓΑ	N/A	N/A	N/A	N/A	1.8245	N/A	N/A
ΓΑ	N/A	N/A	N/A	N/A	2.3810	2.383	N/A
ΓΑ	0.2489	6.1730	8.1789	0.2533	2.4331	N/A	N/A
ΑΑ	0.2442	6.2231	8.2915	0.2443	2.3807	N/A	N/A
Richardson's Random	0.4575	4.7313	5.0853	0.6600	4.1185	N/A	N/A
Simplex	0.2792	5.6256	7.7512	0.2796	2.5307	N/A	N/A
Davidon-Fletcher-Powell	0.2434	6.2552	8.2915	0.2444	2.3841	N/A	N/A
Linear Approximation	0.2444	6.2189	8.2915	0.2444	2.3815	N/A	N/A

Και στα δύο προβλήματα, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό παρείχε καλές υποβέλτιστες λύσεις (near optimal solutions) χωρίς να παραβιάζουν οι λύσεις που πρότείνει κάποιον περιορισμό. Στο πρόβλημα Σχεδιασμού του Ελατηρίου που υποβάλλεται σε Εφελκυσμό/Θλίψη (Tension/Compression Spring Design problem) ξεπέρασε σε απόδοση όλες τις άλλες μεθόδους. Στο πρόβλημα Σχεδιασμού της Συγκολλημένης Δοκού (Welded Beam Design problem) ήταν μακράν καλύτερη από όλες τις άλλες μεθόδους εκτός του Βελτιστοποιητή Γκρίζου Λύκου, αλλά η απουσία στατιστικής ανάλυσης του Βελτιστοποιητή Γκρίζου Λύκου δεν προσφέρει τη δυνατότητα εξαγωγής ασφαλούς συμπεράσματος για την σύγκριση της απόδοσης μεταξύ των δύο μεθόδων. Εντούτοις, η στατιστική ανάλυση αποδεικνύει ότι η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό είναι ένα ποιοτικό εργαλείο βελτιστοποίησης με σταθερή απόδοση, το οποίο παρέχει βέλτιστες ή υποβέλτιστες λύσεις.

7. Εφαρμογές σε πραγματικά προβλήματα

Στις παρακάτω εφαρμογές τα πειράματα έγιναν σε προγραμματιστικό περιβάλλον Matlab σε υπολογιστές με μνήμη 4GB, επεξεργαστή 3.6GHz Intel Core i7 και λειτουργικό Windows 10 Pro, εκτός της περίπτωσης των τεχνητών έργων 90 δραστηριοτήτων του προβλήματος κατανομής πόρων (Tzanetos et al., 2018), όπου χρησιμοποιήθηκε Gnome Terminal με μνήμη 8GB, επεξεργαστή 3.8GHz AMD FX-4300 σε λειτουργικό Linux Arch και με μνήμη 3.7GB, επεξεργαστή 2.27GHz Intel® Core™ i5 M430 σε λειτουργικό Linux Ubuntu 16.04.3. Σε όλα τα προβλήματα που παρουσιάζονται παρακάτω, ο αριθμός των ανεξάρτητων πειραμάτων που εκτελέστηκαν, είναι ίδιος με τον αριθμό των αντίστοιχων πειραμάτων που έχουν γίνει από τις συγκρινόμενες μεθόδους της βιβλιογραφίας και χρησιμοποιείται για να υπάρχει μια βάση σύγκρισης.

7.1 Βελτιστοποίηση Χαρτοφυλακίου

Ένα πολύ γνωστό πρόβλημα βελτιστοποίησης από τον χώρο της Χρηματοοικονομικής Μηχανικής είναι αυτό της Βέλτιστης Διαχείρισης Χαρτοφυλακίου. Βασικός στόχος της διαχείρισης χαρτοφυλακίου είναι η μεγιστοποίηση της απόδοσής του, ενώ παράλληλα ο συνολικός κίνδυνος του χαρτοφυλακίου να είναι ο μικρότερος δυνατός (Markowitz, 1952). Η απόδοση και ο κίνδυνος του χαρτοφυλακίου εξαρτώνται από τις μετοχές που το απαρτίζουν. Μετοχές με υψηλότερη αναμενόμενη απόδοση συντελούν σε χαρτοφυλάκιο με εξίσου υψηλό αναμενόμενο κέρδος, αλλά παράλληλα και υψηλό κίνδυνο. Αντιθέτως ένα χαρτοφυλάκιο που απαρτίζεται από μετοχές με μικρότερη αναμενόμενη απόδοση έχει χαμηλό αναμενόμενο κέρδος και παράλληλα χαμηλό κίνδυνο.

Η αναμενόμενη απόδοση του χαρτοφυλακίου υπολογίζεται ως το άθροισμα των γινομένων της αναμενόμενης απόδοσης r_i κάθε περυσιακού στοιχείου i επί του επενδύμενου κεφαλαίου w_i στο i περιουσιακό στοιχείο. Οπότε για k περιουσιακά στοιχεία ισχύει:

$$E(r_p) = \sum_{i=1}^k w_i \cdot E(r_i) \quad (7.1.1)$$

Ο κίνδυνος του χαρτοφυλακίου υπολογίζεται ως η διακύμανση της απόδοσης του χαρτοφυλακίου:

$$\sigma_p^2 = \sum_{i=1}^k w_i^2 \cdot \sigma_i^2 + \sum_{i=1}^k w_i \cdot \sigma_i \cdot \sigma_j \cdot \rho_{ij} \quad (7.1.2)$$

όπου με σ_i^2 είναι η τυπική απόκλιση των περιοδικών αποδόσεων του περιουσιακού στοιχείου i και ρ_{ij} είναι ο συντελεστής συσχέτισης μεταξύ των αποδόσεων των περιουσιακών στοιχείων i και j .

Το πρόβλημα της Βελτιστοποίησης Χαρτοφυλακίου (Portfolio Optimization), θα μπορούσε να οριστεί ως η εύρεση των κατάλληλων μετοχών που θα απαρτίζουν ένα χαρτοφυλάκιο και παράλληλα η εύρεση του ποσού που θα επενδυθεί στην κάθε μια από το αρχικό κεφάλαιο (Thomaidis et al., 2009; Vassiliadis, Thomaidis and Dounias, 2009). Είναι, δηλαδή, πρόβλημα πολλών στόχων (multi-objective) βάσει του παραπάνω ορισμού.

Στα πλαίσια της παρούσας διατριβής μελετήθηκε η περίπτωση όπου όλο το διαθέσιμο κεφάλαιο επενδύεται στα επιμέρους περυσιακά στοιχεία του χαρτοφυλακίου:

$$\sum_{i=1}^k w_i = 1 \quad (7.1.3)$$

όπου w_i είναι το ποσοστό του επενδύμενου κεφαλαίου στο i περιουσιακό στοιχείο (asset) και με k συμβολίζεται ο μέγιστος αριθμός περιουσιακών στοιχείων (assets) που απαρτίζουν ένα χαρτοφυλάκιο. Επιπλέον, λαμβάνεται υπόψη μόνο η περίπτωση επένδυσης στα περιουσιακά στοιχεία (δηλαδή αγορά αυτών) και όχι η περίπτωση της πώλησης περιουσιακών στοιχείων:

$$0 \leq w_i \leq 1 \quad (7.1.4)$$

Για την αξιολόγηση των λύσεων χρησιμοποιείται ένας χρηματοοικονομικός δείκτης, που ονομάζεται Sortino ratio (Kühn, 2006), ο οποίος λαμβάνει υπόψη του την υπερβάλλουσα απόδοση του χαρτοφυλακίου (excess return of the portfolio) και την μεταβλητότητα των αρνητικών αποδόσεων του χαρτοφυλακίου (volatility of the portfolio's negative returns). Επιπλέον, ο συγκεκριμένος δείκτης περιέχει έναν όρο που δίνει ποινές (penalty term) στις λύσεις που παραβιάζουν το κατώφλι (threshold) της μεταβλητότητας των αποδόσεων. Αυτός ο όρος είναι η διακύμανση των αρνητικών αποδόσεων του χαρτοφυλακίου:

$$\theta_0(r_P) = \sqrt{\int_{-\infty}^{r_f} (r_f - r_P)^2 \cdot f(r_P) dr_P} \quad (7.1.5)$$

όπου r_f είναι η απόδοση της επένδυσης χωρίς κίνδυνο (the risk-free return) και $f(r_P)$ είναι η συνάρτηση πυκνότητας πιθανότητας των αποδόσεων του χαρτοφυλακίου με:

$$f(r_P) = \frac{e^{-\frac{(r_P - E(r_P))^2}{2 \cdot \sigma^2}}}{\sigma \cdot \sqrt{2\pi}} \quad (7.1.6)$$

Για να εντοπιστούν οι λύσεις που παραβιάζουν το κατώφλι (threshold) της μεταβλητότητας των αποδόσεων, χρησιμοποιείται ένας άλλος δείκτης, που

ονομάζεται Παρακολούθηση Μεταβλητότητας Σφάλματος (tracking error volatility - TEV). Ο TEV μετράει τη διακύμανση της απόκλισης μεταξύ των αποδόσεων του χαρτοφυλακίου και ενός δείκτη σημείου αναφοράς της αγοράς (benchmark index). Γενικότερα, ο δείκτης TEV θεωρείται μέτρο παθητικών στρατηγικών στη διαχείριση χαρτοφυλακίου (passive portfolio management strategies) (Sharpe, 1994; Kühn, 2006) και υπολογίζεται ως:

$$\sqrt{\text{Var}(r_P - r_B)} \leq H \quad (7.1.7)$$

όπου r_B είναι η ημερήσια απόδοση του σημείου αναφοράς (benchmark) και H είναι το άνω όριο (threshold) της παρακολούθησης μεταβλητότητας του σφάλματος (tracking error volatility).

Συνδυάζοντας όλα τα παραπάνω, το πρόβλημα της Βελτιστοποίησης Χαρτοφυλακίου (Portfolio Optimization) που μελετήθηκε μπορεί να περιγραφεί, όπως φαίνεται παρακάτω:

Μεγιστοποίηση του
$$\text{Sortino Ratio} = \frac{E(r_P) - r_f}{\theta_0(r_P)}$$

δεδομένου ότι:
$$\sum_{i=1}^k w_i = 1$$

$$0 \leq w_i \leq 1$$

$$k = c$$

$$\sqrt{\text{Var}(r_P - r_B)} \leq H$$

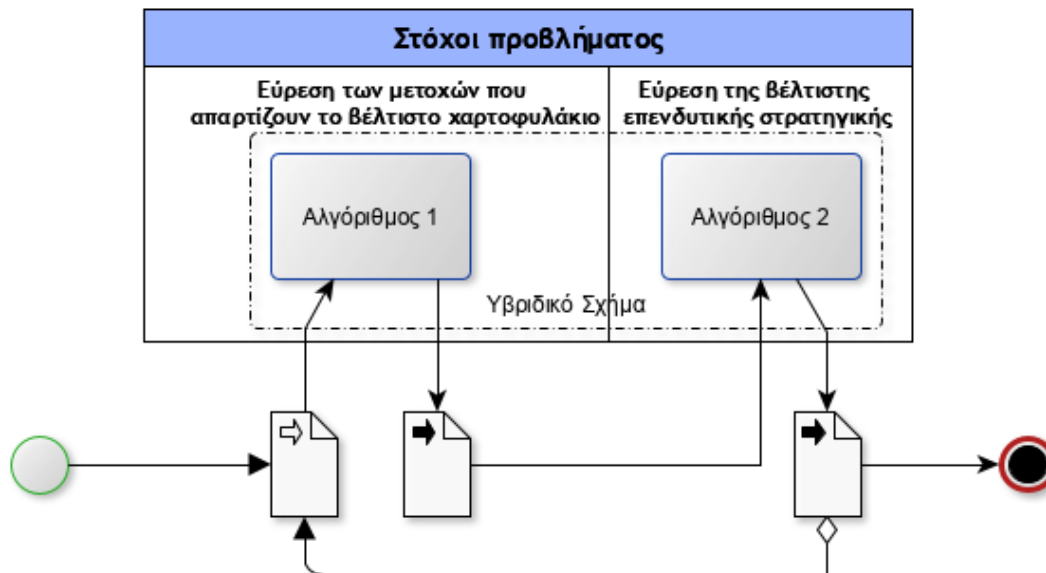
όπου $E(r_P)$ είναι η αναμενόμενη απόδοση του χαρτοφυλακίου και υπολογίζεται μέσω της σχέσης (7.1.1), r_f είναι η απόδοση της επένδυσης χωρίς κίνδυνο (the risk-free return), $\theta_0(r_P)$ είναι η μεταβλητότητα των αποδόσεων (volatility of returns) που παραβιάζουν το κατώφλι H , w_i είναι το ποσοστό του επενδύομένου κεφαλαίου στο i περιουσιακό στοιχείο (asset) και με k συμβολίζεται ο μέγιστος αριθμός περιουσιακών στοιχείων (assets) που απαρτίζουν ένα χαρτοφυλάκιο.

Ο περιορισμός των περιουσιακών στοιχείων (assets) που απαρτίζουν ένα χαρτοφυλάκιο ονομάζεται δεσμευτικός περιορισμός (binding constraint), δηλαδή τα περιουσιακά στοιχεία πρέπει να είναι αυστηρά ίσα με έναν σταθερό αριθμό c – ούτε περισσότερα, ούτε λιγότερα.

7.1.1 Προτεινόμενο υβριδικό σχήμα

Στο παρελθόν έχουν προταθεί διάφορα υβριδικά σχήματα για την επιτυχή βελτιστοποίηση χαρτοφυλακίου, τα οποία αποτελούνταν από Μεθόδους Εμπνευσμένες από τη Φύση. Οι Giannakouris, Vassiliadis and Dounias (2010), χρησιμοποίησαν τη Βελτιστοποίηση Αποικίας Μυρμηγκιών (An Colony Optimization) για το διακριτό μέρος του προβλήματος και τον Αλγόριθμο

Λαμπυρίδων (Firefly Algorithm) για το συνεχές. Στο παρακάτω Σχήμα 7-1 απεικονίζεται ο τρόπος επίλυσης του προβλήματος από τα υβριδικά σχήματα που χρησιμοποιήθηκαν κατά καιρούς από το Εργαστήριο Διοίκησης Επιχειρήσεων και Λήψης Αποφάσεων.

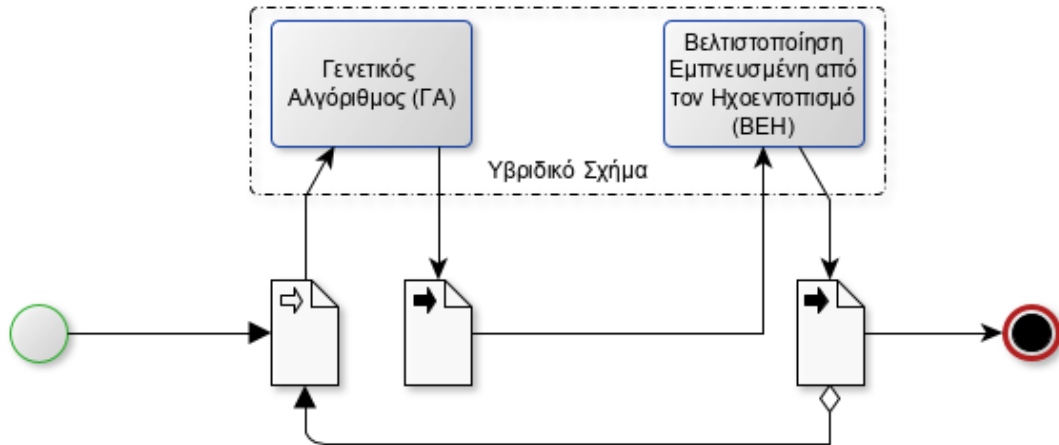


Σχήμα 7-1: Διαγραμματική απεικόνιση προσέγγισης βελτιστοποίησης χαρτοφυλακίου με υβριδικό σχήμα αλγορίθμων

Ο πρώτος αλγόριθμος που απαρτίζει το υβριδικό σχήμα δέχεται σαν είσοδο τις διαθέσιμες μετοχές του δείκτη με σκοπό να προτείνει μια k -άδα μετοχών που θα απαρτίζουν το βέλτιστο χαρτοφυλάκιο. Αυτό το χαρτοφυλάκιο δίνεται σαν είσοδος στον δεύτερο αλγόριθμο του υβριδικού σχήματος με σκοπό να αναζητήσει το ποσό του διαθέσιμου κεφαλαίου που πρέπει να επενδυθεί σε κάθε μετοχή που απαρτίζει το χαρτοφυλάκιο. Το αποτέλεσμα ανατροφοδοτεί τον πρώτο αλγόριθμο, ο οποίος πλέον έχει μια πληροφορία για την σημαντικότητα των μετοχών και προτείνει νέα χαρτοφυλάκια βάσει αυτής.

Πιο πρόσφατα, χρησιμοποιήθηκε ο Αλγόριθμος Βαρυτικής Αναζήτησης (ABA) αντί του Αλγόριθμου Λαμπυρίδων (ΑΛ) ως το δεύτερο μέρος του υβριδικού σχήματος που λύνει το συγκεκριμένο πρόβλημα (Tzanetos, Vassiliadis and Dounias, 2017). Αυτό που παρατηρήθηκε είναι ότι η απόδοση του δεύτερου αλγόριθμου του υβριδικού σχήματος περιοριζόταν αισθητά από τα χαρτοφυλάκια με τα οποία τον τροφοδοτούσε ο πρώτος αλγόριθμος. Η Βελτιστοποίηση Αποικίας Μυρμηγκιών λειτουργούσε πιο αυστηρά στην επιλογή των μετοχών που θα απαρτίζουν το καλύτερο χαρτοφυλάκιο και πολύ γρήγορα (στα πρώτα βήματα της αλγοριθμικής διαδικασίας) είχε αυξήσει τις πιθανότητες εμφάνισης συγκεκριμένων μετοχών με αποτέλεσμα τα προτεινόμενα χαρτοφυλάκια να τις περιέχουν σε ένα μεγάλο βαθμό και έτσι να μην γίνεται αρκετή εξερεύνηση (exploration) από το υβριδικό σχήμα.

Στα πλαίσια της παρούσας διατριβής δοκιμάστηκε με επιτυχία ένα διαφορετικό υβριδικό σχήμα, το οποίο αντί της Βελτιστοποίησης Αποικίας Μυρμηγκιών (BAM) είχε τον Γενετικό Αλγόριθμο (ΓΑ) ως μέθοδο επίλυσης του διακριτού προβλήματος. Η λογική πίσω από αυτή την αλλαγή ήταν να χρησιμοποιηθεί ένα απλοϊκό (naive) εργαλείο που δεν θα σπρώχνει όλο το υβριδικό σχήμα να εξετάζει τις ίδιες μετοχές συνέχεια.



Σχήμα 7-2: Προτεινόμενο υβριδικό σχήμα για Βελτιστοποίηση Χαρτοφυλακίου

Το διάλυμα της λύσης στον Γενετικό Αλγόριθμο (ΓΑ) για το συγκεκριμένο πρόβλημα έχει τη μορφή:

$$X_i^{GA} = \{asset_1, asset_2, asset_3, \dots, asset_k\} \quad (7.1.1.1)$$

όπου με $asset_j$ συμβολίζεται το j -οστό περιουσιακό στοιχείο του χαρτοφυλακίου, k είναι το πλήθος των περιουσιακών στοιχείων του χαρτοφυλακίου και το $i \in 1, 2, \dots, N$ και N είναι το πλήθος των αναζητητών του ΓΑ. Το αντίστοιχο διάλυμα της BEH είναι της μορφής:

$$X_i^{BEH} = \{w_1, w_2, w_3, \dots, w_k\} \quad (7.1.1.2)$$

όπου w_j είναι το ποσοστό του επενδύμενου κεφαλαίου στην μετοχή j , το $i \in 1, 2, \dots, N$ και N είναι το πλήθος των αναζητητών της BEH, ενώ k είναι το πλήθος των περιουσιακών στοιχείων του χαρτοφυλακίου.

Στην αρχικοποίηση του Γενετικού Αλγορίθμου (ΓΑ), δημιουργείται τυχαία ένας πληθυσμός αποτελούμενος από πενήντα (50) χαρτοφυλάκια των πέντε (5) μετοχών. Ο αριθμός αυτός των πενήντα χαρτοφυλακίων επιλέχθηκε έτσι ώστε κάθε μετοχή να εμφανίζεται αρχικά τουλάχιστον σε ένα χαρτοφυλάκιο.

Για να διασφαλιστεί ότι δεν θα εμφανίζεται δύο φορές η ίδια μετοχή σε ένα χαρτοφυλάκιο, καθώς και ότι κάθε χαρτοφυλάκιο (κάθε συνδυασμός μετοχών) εμφανίζεται μια φορά στον πληθυσμό, έχει εισαχθεί ένας σχετικός έλεγχος στο

υβριδικό σχήμα. Ο εν λόγω έλεγχος, εφόσον εντοπίσει κάποιο από τα δύο προαναφερόμενα προβλήματα, επαναδημιουργεί το χαρτοφυλάκιο με μετοχές από τις εναπομείναντες.

Ύστερα, ο πληθυσμός αυτός που περιέχει τα χαρτοφυλάκια δίνεται σαν είσοδος στην ΒΕΗ με σκοπό να βρει τη βέλτιστη κατανομή του κεφαλαίου για κάθε χαρτοφυλάκιο και να υπολογίσει την ποιότητα κάθε υβριδικής (πλέον) λύσης. Η υβριδική λύση αποτελείται από δύο διανύσματα, όπως αυτά περιγράφονται στις σχέσεις (7.1.1.1) και (7.1.1.2) παραπάνω.

Ο ρυθμός μετάλλαξης (mutation rate) του ΓΑ μεταβάλλεται κατά τη διάρκεια της εκτέλεσης του υβριδικού σχήματος, αντίθετα με τον κλασικό Γενετικό Αλγόριθμο όπου ο ρυθμός μετάλλαξης παραμένει σταθερός καθ' όλη τη διάρκεια της αλγοριθμικής διαδικασίας (Holland, 1992b). Εάν δεν έχει βρεθεί καλύτερη λύση για κάποιες συνεχόμενες γενιές ο ρυθμός μετάλλαξης μεταβάλλεται ως ακολούθως: η αρχική τιμή ορίστηκε 0.15, ενώ αυξάνεται στο 0.20 αν ύστερα από δύο συνεχόμενες γενιές δεν έχει βρεθεί καλύτερη λύση και αυξάνεται περαιτέρω στο 0.25 αν ύστερα από τρεις ακόμη συνεχόμενες επαναλήψεις δεν βελτιώνεται η ήδη υπάρχουσα λύση. Το 0.25 είναι η μέγιστη τιμή που μπορεί να πάρει στο υβριδικό σχήμα ο ρυθμός μετάλλαξης.

Είναι σημαντικό να αναφερθεί εδώ ότι συνολικά ο Γενετικός Αλγόριθμος εκτελείται για 10 γενιές, καθώς ένα υβριδικό σχήμα όπου η μία μέθοδος καλεί την άλλη για να υπολογίσει κάποιο μέγεθος για κάθε στοιχείο του πληθυσμού της πρώτης, εκτελείται σε εκθετικό χρόνο. Κατ' επέκταση, όπως φαίνεται και παρακάτω, οι παράμετροι και των δύο μεθόδων που απαρτίζουν το υβριδικό σχήμα έχουν επιλεγεί με τέτοιο τρόπο ώστε να είναι όσο το δυνατόν λιγότερο χρονοβόρα η υβριδική μέθοδος.

Η επιλογή των καλύτερων χαρτοφυλακίων τα οποία θα παραμείνουν στον επόμενο πληθυσμό γίνεται μέσω ελιτισμού (elitism). Οι νέες λύσεις του επόμενου πληθυσμού που δημιουργούνται μέσω διασταύρωσης (crossover) προέρχονται από τα καλύτερα χαρτοφυλάκια (σύμφωνα με τη ΒΕΗ) τα οποία έχουν την υψηλότερη ποιότητα λύσης (σε τιμές Sortino). Τα ζεύγη που θα διασταυρωθούν επιλέγονται μέσω ενός τουρνουά (tournament).

Στη συνέχεια, σύμφωνα με τον ρυθμό μετάλλαξης που αναφέρθηκε παραπάνω, εκτελείται το βήμα της μετάλλαξης (mutation) του ΓΑ. Σε αυτό το σημείο της μεθόδου, έχει προστεθεί ένας μηχανισμός ο οποίος συγκρίνει τον πληθυσμό της εκάστοτε γενιάς με αυτόν της προηγούμενης. Αυτό γίνεται με σκοπό να διευρυνθεί ο χώρος αναζήτησης του ΓΑ και να αποφευχθεί η πιθανότητα να εγκλωβιστεί σε τοπικά βέλτιστα, καθώς με την προαναφερόμενη σύγκριση ελέγχεται εάν εμφανίζονται ίδια χαρτοφυλάκια από τη μία γενιά στην επόμενη. Τόσο στο βήμα της διασταύρωσης όσο και σε αυτό της μετάλλαξης, γίνεται σύγκριση του εκάστοτε πληθυσμού με τον

προηγούμενο και υπολογίζεται το επίπεδο διαφοροποίησης (differentiation level) του πληθυσμού από την προηγούμενη γενιά.

Αφού ολοκληρωθεί το βήμα της μετάλλαξης, εάν το επίπεδο διαφοροποίησης είναι χαμηλό, δηλαδή λιγότερο από τον μισό πληθυσμό αποτελείται από μοναδικά χαρτοφυλακία, τα χαρτοφυλάκια της τρέχουσας γενιάς που είναι ίδια με την προηγούμενη αντικαθιστούνται με χαρτοφυλάκια που τα συνιστούν μετοχές οι οποίες δεν εμφανίζονται στον τρέχον πληθυσμό.

Στο τέλος κάθε επανάληψης του ΓΑ εκτελείται ξανά ο έλεγχος για μετοχές που εμφανίζονται δύο φορές σε ένα χαρτοφυλάκιο και για χαρτοφυλάκια που εμφανίζονται δύο ή περισσότερες φορές στον πληθυσμό, καθώς η τυχαιότητα του βήματος της μετάλλαξης μπορεί να οδηγήσει σε μια από τις δύο προαναφερόμενες περιπτώσεις.

7.1.2 Εισαγωγή Ευρετικής Πληροφορίας

Μελετώντας το πρόβλημα, παρατηρήθηκε ότι τα καλύτερα χαρτοφυλάκια αποτελούνταν συνήθως από κάποιες συγκεκριμένες μετοχές (Tzanetos, Vassiliadis and Dounias, 2017). Κατ' επέκταση, προτάθηκε σε σχετική εργασία η χρήση αυτής της πληροφορίας (Tzanetos, Vassiliadis and Dounias, 2018), η οποία λόγω του ότι προέρχεται από παρατήρηση επί των πρωταρχικών αποτελεσμάτων μιας πειραματικής διαδικασίας, ονομάζεται ευρετική πληροφορία.

Για να εισηχθεί η ευρετική πληροφορία στον ΓΑ, οι τέσσερις μετοχές που παρατηρήθηκε να εμφανίζονται στα καλύτερα χαρτοφυλάκια (συγκεκριμένα, οι μετοχές 171, 174, 242 και 438) είχαν περισσότερες πιθανότητες να επιλεγούν σε ένα χαρτοφυλάκιο. Ειδικότερα, σε ξεχωριστά σεντ πειραμάτων που έγιναν, οι εν λόγω μετοχές είχαν πιθανότητα εμφάνισης 20% και 40%, αντίστοιχα.

7.1.3 Αποτελέσματα Πειραματικής Διαδικασίας

Το πρόβλημα που μελετήθηκε στα πλαίσια της παρούσας διατριβής περιγράφεται ως:

$$\begin{aligned} \text{Μεγιστοποίηση του} & \quad \text{Sortino Ratio} = \frac{E(r_P) - r_f}{\theta_0(r_P)} \\ \text{δεδομένου ότι:} & \quad \sum_{i=1}^k w_i = 1 \\ & \quad 0 \leq w_i \leq 1 \\ & \quad k = 5 \\ & \quad \sqrt{\text{Var}(r_P - r_B)} \leq 0,0001 \end{aligned}$$

Δηλαδή, ο δεσμευτικός περιορισμός των περιουσιακών στοιχείων (assets) που απαρτίζουν το χαρτοφυλάκιο στο πρόβλημα ορίζει τα περιουσιακά στοιχεία να είναι

πέντε (5) και το άνω όριο (threshold) της παρακολούθησης μεταβλητότητας του σφάλματος (tracking error volatility) ορίζεται ίσο με 0,0001. Η μεταβλητότητα των αποδόσεων (volatility of returns) που παραβιάζουν το κατώφλι H είναι ίση με $2.8 \cdot 10^{-4}$.

Πίνακας 7.1: Τιμές παραμέτρων των αλγορίθμων που απαρτίζουν το προτεινόμενο υβριδικό σχήμα

Παράμετρος	Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό	Γενετικός Αλγόριθμος
Πληθυσμός	50	50
Γενιές	30	10
μ	0.05	-
α^o	18	-
Ρυθμός μετάλλαξης	-	0.15, 0.20, 0.25 (αυτορυθμιζόμενος)
Γονιδίωμα	-	5

Τα δεδομένα προέρχονται από το “Yahoo Finance” και αποτελούνται από ημερήσιες τιμές κλεισίματος των μετοχών του χρηματιστηριακού δείκτη S&P’s 500, ο οποίος αποτελεί έναν αντιπροσωπευτικό δείκτη αναφοράς των χρηματιστηριακών αγορών. Η περίοδος των δεδομένων εκτείνεται στην περίοδο των έξι μηνών από 01/12/2008 έως 01/05/2009, η οποία ήταν κρίσιμη περίοδος των παγκόσμιων αγορών, λόγω των επιπτώσεων της παγκόσμιας οικονομικής κρίσης που ξέσπασε το 2008.

Πίνακας 7.2: Σύγκριση αποτελεσμάτων προτεινόμενης μεθόδου με άλλες μεθόδους, χωρίς και με ευρετική πληροφορία

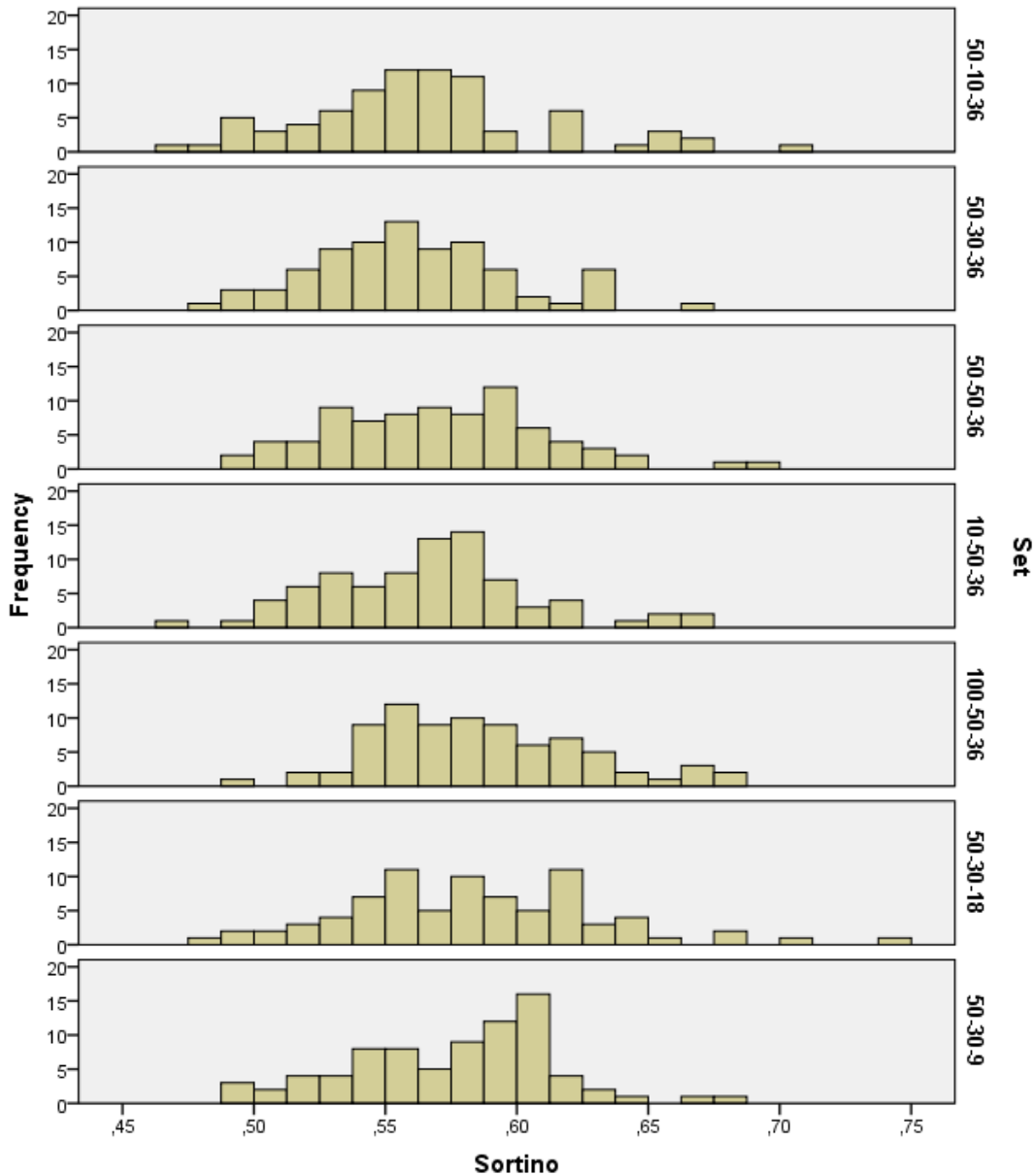
Μέθοδος	Μέση Τιμή	Τυπική Απόκλιση	Λοξότητα	Sortino	Χαρτοφυλάκιο
ΓΑ-BEH	0.5815	0.0458	0.2314	0.7092	[45,171,438,242,428]
ΓΑ-BEH με ευρ. πληρ. (20%)	0.6816	0.0287	-1.1472	0.7484	[171,174,242,438,18]
ΓΑ-BEH με ευρ. πληρ. (40%)	0.6996	0.0163	0.3616	0.7619	[438,171,242,174,18]
Συγκρινόμενες Μέθοδοι					
BAM-ΑΛ	0.5024	0.0284	0.8055	0.5996	[81,115,174,242,438]
BAM-ΑΒΑ με ευρ. πληρ.	0.6620	0.0215	0.5550	0.6952	[45,171,242,283,438]
BAM-ΑΛ με ευρ. πληρ.	0.6689	0.0252	0.0831	0.7134	[30,45,171, 242,438]
Monte Carlo & Μη Γραμμ. Προγραμματισμός	-	-	-	0.5548	N/A

Όπως φαίνεται στον Πίνακα 7.2, το προτεινόμενο υβριδικό σχήμα (ΓΑ-BEH) ξεπέρασε την απόδοση των υπόλοιπων υβριδικών, αλλά και μιας κλασικής μεθόδου αποτελούμενης από Monte Carlo και Μη Γραμμικό Προγραμματισμό, όπως και της τακτικής να επενδύσει κανείς το διαθέσιμο κεφάλαιο στις πέντε (5) μετοχές με την

υψηλότερη κεφαλαιοποίηση. Η σύγκριση γίνεται ως προς την τιμή του Sortino ratio, όπου ο στόχος είναι η ελαχιστοποίησή του.

Μάλιστα, παρατηρεί κανείς (Πίνακας 7.2) ότι η εισαγωγή της ευρετικής πληροφορίας βελτίωσε περαιτέρω την απόδοση του υβριδικού σχήματος Γενετικού Αλγορίθμου-Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΓΑ-BEH).

7.1.4 Ανάλυση Παραμέτρων της Προτεινόμενης Μεθόδου στο Πρόβλημα Βελτιστοποίησης Χαρτοφυλακίου



Γράφημα 7-1: Κατανομές ανεξάρτητων πειραμάτων για διάφορα σετ παραμέτρων

Στα πλαίσια της ρύθμισης παραμέτρων (parameter tuning) της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (BEH) έγιναν 80 ανεξάρτητα πειράματα για διαφορετικά σετ παραμέτρων. Ο Πίνακας 7.3 περιέχει τις πληροφορίες κάθε σετ:

πόσοι αναζητητές χρησιμοποιήθηκαν, για πόσες γενιές εκτελέστηκε ο αλγόριθμος και ποια ήταν η μέγιστη γωνία περιστροφής. Οι παράμετροι του Γενετικού Αλγόριθμου παρέμειναν ίδιες με αυτές της Βελτιστοποίησης Αποικίας Μυρμηγκιών από παλαιότερες εργασίες (Tzanetos, Vassiliadis and Dounias, 2018, 2017; Giannakouris, Vassiliadis and Dounias, 2010).

Όπως ήταν αναμενόμενο, η αύξηση των γενεών στη ΒΕΗ είχε ως συνέπεια τη βελτίωση της κατανομής των αποτελεσμάτων του υβριδικού σχήματος, όπως φαίνεται και στο Γράφημα 7-1.

Πίνακας 7.3: Παράμετροι ΒΕΗ σεκάθε σετ πειραμάτων

Σετ Παραμέτρων	1	2	3	4	5	6	7
Αναζητητές	50	50	50	10	100	50	50
Γενιές	10	30	50	50	50	30	30
Μέγιστη γωνίας περιστροφής	36	36	36	36	36	18	9
Καλύτερο αποτέλεσμα	0.7004	0.6713	0.6908	0.6717	0.6818	0.7482	0.6761
Μέση τιμή	0.5646	0.5608	0.5712	0.5666	0.5863	0.5834	0.5763
Λοξότητα	0.6352	0.3496	0.4202	0.3381	0.4346	0.5600	-0.1397

Το καλύτερο αποτέλεσμα (Sortino), όπως φαίνεται και στον Πίνακα 7.3 παραπάνω, επιτεύχθηκε στο σετ των 50 αναζητητών, 30 γενεών και 18° μέγιστης γωνίας περιστροφής. Η κατανομή του συγκεκριμένου σετ παραμέτρων παρουσίασε μεγαλύτερη δεξιά λοξότητα και είχε μια εκ των δύο υψηλότερων μέσων αποδόσεων.

7.2 Βέλτιστη Διαχείριση Έργων

Η Διαχείριση Έργων (Project Management) αποτελεί ένα πεδίο στο οποίο πολύ συχνά καλείται ένας Μηχανικός να αντιμετωπίσει προβλήματα εύρεσης βέλτιστης στρατηγικής. Το πρόβλημα της Κατανομής Πόρων (Resource Leveling) είναι ένα τέτοιο πρόβλημα εύρεσης βέλτιστης στρατηγικής και μπορεί να περιγραφεί ως η βέλτιστη κατανομή πόρων στις χρονικές στιγμές ενός έργου συναρτήσει διάφορων παραγόντων, όπως ο χρόνος περάτωσης του έργου κλπ.

Είναι ευρέως γνωστό πως για τον υπολογισμό της ενωρίτερης (Early Start - ES) και αργότερης έναρξης (Late Start - LS) κάθε δραστηριότητας ενός έργου χρησιμοποιείται η Μέθοδος του Κρίσιμου Μονοπατιού (Critical Path Method - CPM). Για όσες δραστηριότητες ισχύει:

$$ES^d = LS^d \quad (7.2.1)$$

δηλαδή η ενωρίτερη και η αργότερη έναρξη της d δραστηριότητας συμπίπτουν, οι δραστηριότητες αυτές δημιουργούν το Κρίσιμο Μονοπάτι. Κατ' επέκταση, το Κρίσιμο Μονοπάτι αποτελείται από τις δραστηριότητες του έργου, οι οποίες πρέπει να ξεκινήσουν κάποια συγκεκριμένη ημέρα, ώστε να μην υπάρξει καθυστέρηση στην ολοκλήρωση του έργου. Οπότε, στο πρόβλημα της βέλτιστης Κατανομής Πόρων, ο στόχος είναι να βρεθεί η κατάλληλη χρονική στιγμή έναρξης κάθε δραστηριότητας

που δεν ανήκει στο Κρίσιμο Μονοπάτι, ώστε να υπάρξει πιο ομοιόμορφη κατανομή των διαθέσιμων πόρων κατά τη διάρκεια του έργου.

Η πολυπλοκότητα του προβλήματος της Κατανομής Πόρων δίνεται από το σύνολο των διαθέσιμων χρονικών ενάρξεων των μη-κρίσιμων δραστηριοτήτων. Έστω ότι μελετάμε ένα έργο 13 δραστηριοτήτων, όπου η ενωρίτερη και αργότερη έναρξη κάθε δραστηριότητας φαίνεται στον παρακάτω Πίνακας 7.4.

Πίνακας 7.4: Ενωρίτερη και Αργότερη Έναρξη ενός έργου 13 δραστηριοτήτων

<i>d</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
ES	1	1	1	1	3	5	5	2	8	8	11	16	16
LS	1	7	1	4	11	9	5	5	15	12	11	16	16

Με κόκκινο σημειώνονται οι δραστηριότητες που απαρτίζουν το Κρίσιμο Μονοπάτι. Ο υπολογισμός των διαθέσιμων χρονικών ενάρξεων (ΔXE) των υπόλοιπων επτά μη-κρίσιμων δραστηριοτήτων γίνεται ως ακολούθως:

$$\Delta XE^d = LS^d - ES^d + 1 \tag{7.2.2}$$

Οπότε οι διαθέσιμες χρονικές ενάρξεις κάθε δραστηριότητας *d* είναι:

Πίνακας 7.5: Διαθέσιμες Χρονικές Ενάρξεις κάθε μη-κρίσιμης δραστηριότητας *d*

<i>d</i>	2	4	5	6	8	9	10
ΔXE^d	7	4	9	5	4	8	5

Για να υπολογίσουμε την πολυπλοκότητα του προβλήματος της Κατανομής Πόρων στο παραπάνω έργο, αφού οι πιθανές λύσεις δημιουργούνται από τον συνδυασμό κάθε διαθέσιμης χρονικής έναρξης των μη-κρίσιμων δραστηριοτήτων, έχουμε:

$$\text{Συνδυασμοί} = 7 \times 4 \times 9 \times 5 \times 4 \times 8 \times 5 = 201.600 \tag{7.2.3}$$

Δεδομένου ότι τα πραγματικά έργα είναι πολλών δραστηριοτήτων, γίνεται σαφές ότι το πρόβλημα της Κατανομής Πόρων ανήκει στην κλάση NP, ενώ αν οι μη-κρίσιμες δραστηριότητες είναι πάρα πολλές, το εν λόγω πρόβλημα μπορεί να αναχθεί στην κλάση NP-δυσμενή.

7.2.1 Αντικειμενικές Συναρτήσεις

Σύμφωνα με τον Boctor (1996), άλλες αντικειμενικές συναρτήσεις πέρα από όσες περιγράφονται παρακάτω είναι η ελαχιστοποίηση της κατανομής (distribution) των πόρων (R^2), η ελαχιστοποίηση του λόγου μεταξύ του γινομένου της διάρκειας του έργου με το τετράγωνο της ημερήσιας χρήσης των πόρων και του τετραγώνου

του αθροίσματος όλων των πόρων του έργου (*RIC*) και η Εντροπία των Πόρων (*Entropy of Resources*). Κάθε μια από τις αντικειμενικές συναρτήσεις, οι οποίες επεξηγούνται παρακάτω, μπορεί να χρησιμοποιηθεί ξεχωριστά για την επίλυση του προβλήματος της Κατανομής Πόρων ή ακόμη και σε συνεργασία με οποιαδήποτε από τις άλλες.

A) Ελαχιστοποίηση της μέγιστης χρήσης πόρων

Η πιο διαδεδομένη αντικειμενική συνάρτηση στο πρόβλημα της Κατανομής Πόρων είναι η ελαχιστοποίηση της μέγιστης χρήσης πόρων για μια μονάδα του χρόνου (Neumann and Zimmermann, 2000; S. Zhao et al., 2006; N. Pang, Y. Shi and Y. You, 2008):

$$G_f = \max\{F(t), t = 1, 2, \dots, T\} \quad (7.2.1.1)$$

όπου $F(t)$ είναι το άθροισμα των πόρων που χρειάζονται την χρονική στιγμή t . Μπορεί να υπολογιστεί προσθέτοντας τους πόρους r_i που χρειάζονται για κάθε δραστηριότητα i του έργου, οι οποίες είναι σε εξέλιξη κατά την χρονική στιγμή t :

$$F(t) = \sum_{i \in \{1, \dots, n\}} r_i(t) \quad (7.2.1.2)$$

B) Ελαχιστοποίηση της μέσης χρήσης πόρων

Η διαφορά μεταξύ πραγματικής (*actual*) και επιθυμητής χρήσης πόρων (*desirable resource usage*) συγκρίνεται στο κριτήριο *RLI*, όπου η πραγματική χρήση αντιπροσωπεύει τους πόρους που απαιτούνται σε μια μονάδα χρόνου βάσει του χρονοπρογραμματισμού που έχει γίνει, ενώ ως επιθυμητή χρήση πόρων χρησιμοποιείται η μέση χρήση που παρατηρείται στο έργο. Δηλαδή, ο στόχος σε αυτό το κριτήριο είναι να ελαχιστοποιηθούν οι πόροι που είναι πάνω από την μέση χρήση (Younis and Saad, 1996; Elwany et al., 1998; Leu, Yang and Huang, 2000; Rieck, Zimmermann and Gather, 2012):

$$RLI = \sum_{t \in \{1, 2, \dots, T\}} \left| \left(\sum_{i \in \{1, 2, \dots, n\}} r_i(t) \right) - \left(\frac{\sum_{i \in \{1, 2, \dots, n\}} r_i d_i}{T} \right) \right| \quad (7.2.1.3)$$

όπου d_i είναι η διάρκεια της i δραστηριότητας και με r_i συμβολίζονται οι πόροι που χρειάζεται η i δραστηριότητα στη δεδομένη μονάδα του χρόνου.

Γ) Ελαχιστοποίηση της τυπικής απόκλισης της κατανομής πόρων

Η παρακάτω σχέση ελαχιστοποιεί την τυπική απόκλιση της κατανομής των πόρων από τη μέση χρήση πόρων (Bandelloni, Tucci and Rinaldi, 1994; Neumann and Zimmermann, 2000; N. Pang, Y. Shi and Y. You, 2008; Anagnostopoulos and Koulinas, 2010; Huang, Wang and Chen, 2010; Rieck, Zimmermann and Gather, 2012):

$$StD = \frac{1}{T-1} \sum_{i \in \{1,2,\dots,n\}, t \in \{1,2,\dots,T\}} \left(\left(\sum_{i \in \{1,2,\dots,n\}} r_i(t) \right) - \left(\frac{\sum_{i \in \{1,2,\dots,n\}} r_i d_i}{T} \right) \right)^{\frac{1}{2}} \quad (7.2.1.4)$$

όπου d_i είναι η διάρκεια της i δραστηριότητας και με r_i συμβολίζονται οι πόροι που χρειάζεται η i δραστηριότητα στη δεδομένη μονάδα του χρόνου.

Δ) Κριτήριο STEP

Το συγκεκριμένο κριτήριο έχει ως στόχο την πιο ομαλή μετάβαση στους συνολικούς πόρους που παρατηρείται από μια χρονική στιγμή στην επόμενη (Younis and Saad, 1996; Neumann and Zimmermann, 1999, 2000; Z. Li, P. Wuliang and Z. Zhongliang, 2010):

$$STEP = \sum_{t \in \{1,2,\dots,T\}} |F(t) - F(t-1)| \quad (7.2.1.5)$$

Στην ουσία, αυτό το κριτήριο αξιολόγησης των προφίλ πόρων διασφαλίζει ότι οι συνολικοί πόροι που χρησιμοποιούνται σε μια μονάδα χρόνου δεν θα απέχουν πολύ από το αντίστοιχο σύνολο της επόμενης χρονικής στιγμής. Αποκλείει δηλαδή τα προφίλ που έχουν π.χ. 10 πόρους την χρονική στιγμή $t-1$ και 2 πόρους την χρονική στιγμή t ή το ανάποδο.

7.2.2 Τροποποιήσεις του Αλγορίθμου

Δύο βασικοί περιορισμοί υπάρχουν στο πρόβλημα της Βέλτιστης Κατανομής Πόρων, όπως αυτό έχει αναλυθεί παραπάνω:

- η συνολική διάρκεια (total duration) του έργου δεν πρέπει να ξεπερνάει τον ελάχιστο χρόνο περάτωσης του έργου, όπως αυτός υπολογίζεται με την Μέθοδο του Κρίσιμου Μονοπατιού (Critical Path Method - CPM) και
- κάθε δραστηριότητα πρέπει να ξεκινάει αφού έχουν ολοκληρωθεί όλες οι προαπαιτούμενες της.

Λόγω αυτών των περιορισμών πρέπει να αποφεύγει ο αλγόριθμος να παράγει μη εφικτές λύσεις, ώστε να μην χάνεται πολύτιμος υπολογιστικός χρόνος και ισχύς (computational time and power). Η ύπαρξη του μηχανισμού διόρθωσης λύσεων συντελεί σε αυτό.

Στην προτεινόμενη μέθοδο, λοιπόν, κάθε αναζητητής θεωρείται μια λύση:

$$X_i = \{st_1, st_2, st_3, \dots, st_n\} \quad (7.2.2.1)$$

όπου st_j είναι η χρονική στιγμή έναρξης της δραστηριότητας j , το $i \in 1, 2, \dots, N$ και N είναι το πλήθος των αναζητητών, ενώ n είναι το πλήθος των δραστηριοτήτων του έργου. Υπολογίζοντας την Νωρίτερη (Early) και Αργότερη Έναρξη (Late Start) κάθε

δραστηριότητας j , ορίζονται αντίστοιχα ως άνω και κάτω όριο για κάθε διάσταση του προβλήματος:

$$upper_bound_j = ES_j$$

$$lower_bound_j = LS_j$$

Κάθε διάσταση του προβλήματος αντιστοιχεί σε κάθε δραστηριότητα, όπως ορίζονται αυτές στο έργο με αύξουσα σειρά. Με αυτό τον τρόπο, οι δραστηριότητες που συντελούν το Κρίσιμο Μονοπάτι (Critical Path) θα έχουν σταθερή χρονική στιγμή έναρξης, με αποτέλεσμα πολλές μη εφικτές (non-feasible) λύσεις να μην ελέγχονται. Οπότε, κάθε λύση παράγεται από τη σχέση:

$$st_j^d = lower_bound^d + (upper_bound^d - lower_bound^d) \cdot rand \quad (7.2.2.2)$$

Ωστόσο, όταν μια δραστηριότητα έχει προαπαιτούμενες, η σχέση αυτή δεν παράγει εφικτές λύσεις. Σε αυτή την περίπτωση, λοιπόν, ελέγχονται όλες οι χρονικές στιγμές ολοκλήρωσης όλων των προαπαιτούμενων δραστηριοτήτων και η μέγιστη τιμή αυτών (ώστε να έχουν ολοκληρωθεί όλες) χρησιμοποιείται ως το νέο κάτω όριο της δραστηριότητας j :

$$st_j^d = temporary_lower_bound^d + (upper_bound^d - lower_bound^d) \cdot rand \quad (7.2.2.3)$$

όπου το προσωρινό κάτω όριο $temporary_lower_bound^d$ μπορεί να περιγραφεί από τη σχέση:

$$temporary_lower_bound^d = \max\{st_j^k + duration^d, k \in (\text{Προαπαιτούμενες})\} \quad (7.2.2.4)$$

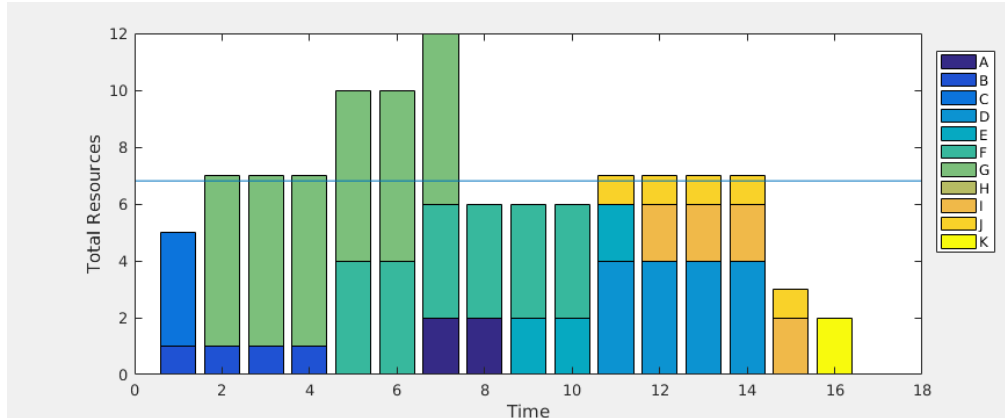
όπου με $duration^d$ συμβολίζεται η διάρκεια της d δραστηριότητας και st_j^k συμβολίζεται η χρονική στιγμή έναρξης της k δραστηριότητας, η οποία είναι προαπαιτούμενη της δραστηριότητας d που εξετάζεται. Βάσει της (4.2.2.2), ο Μηχανισμός Διόρθωσης Λύσεων που περιγράφεται στην προηγούμενη ενότητα, πλέον τροποποιείται ως:

$$st_{j_new}^d = temporary_lower_bound^d + (upper_bound^d - lower_bound^d) \cdot \cos(st_{j_old}^d) \quad (7.2.2.5)$$

7.2.3 Επίλυση τεχνητών έργων της βιβλιογραφίας

Για τον έλεγχο αποδοτικότητας της προτεινόμενης μεθόδου στο πρόβλημα της Βέλτιστης Κατανομής Πόρων χρησιμοποιήθηκαν δύο μικρά έργα της βιβλιογραφίας με γνωστές διακριτές λύσεις (Leu, Yang and Huang, 2000; Leu and Hung, 2002), καθώς και επιλεγμένα τεχνητά έργα 90 και 120 δραστηριοτήτων, των οποίων τα δεδομένα βρίσκονται στην βιβλιοθήκη PSPLib (Sprecher and Kolisch, 1996). Τα πειράματα των μικρών έργων και των επιλεγμένων έργων 90 δραστηριοτήτων έγιναν χρησιμοποιώντας Gnome Terminal σε λειτουργικό 8GB, 3.8GHz AMD FX-4300

Linux Arch και 3.7GB, 2.27GHz Intel® Core™ i5 M430 Linux Ubuntu 16.04.3. Για τα πειράματα των τεχνητών έργων 120 δραστηριοτήτων χρησιμοποιήθηκε το παράθυρο εντολών (command window) σε υπολογιστή με μνήμη 4GB, επεξεργαστή 3.6GHz Intel Core i7 και λειτουργικό Windows 10 Pro



Εικόνα 7.1: Προφίλ πόρων για το Πρόβλημα 1 (Leu et al., 2000)

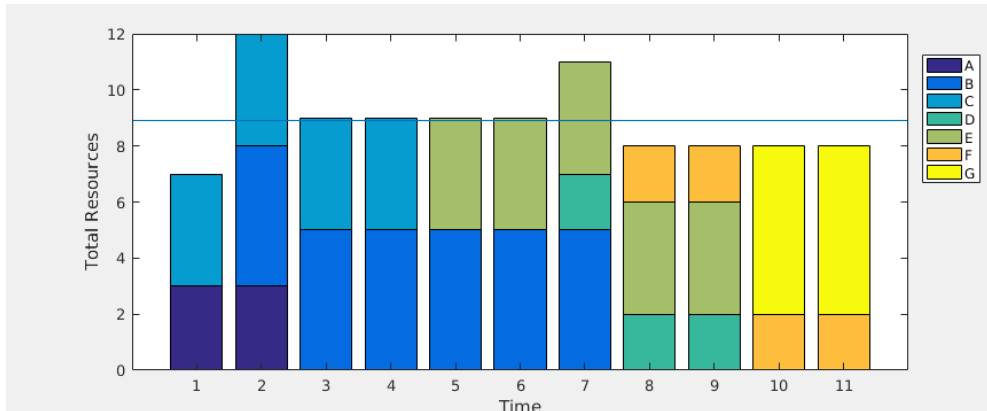
Αρχικά, εκτελέστηκαν 40 ανεξάρτητα πειράματα για κάθε ένα από τα έργα με γνωστές λύσεις (benchmark problems). Για κάθε πείραμα χρησιμοποιήθηκαν 20 αναζητητές και έγιναν 20 επαναλήψεις. Στις Εικόνες Εικόνα 7.1 και Εικόνα 7.2 φαίνονται τα προφίλ πόρων που πρότεινε η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) για κάθε πρόβλημα. Επιπλέον, στον Πίνακα 7.6 συγκρίνεται η γνωστή καλύτερη λύση, με την αντίστοιχη που έδωσε για κάθε πρόβλημα ο προτεινόμενος αλγόριθμος (BEH), καθώς και η στατιστική απόδοσή του (ποσοστιαία επιτυχία εύρεσης της γνωστής βέλτιστης τιμής σε σύνολο 40 ανεξάρτητων τρεξιμάτων). Ως μέτρο σύγκρισης έχουν χρησιμοποιηθεί τα αντίστοιχα αποτελέσματα του Γενετικού Αλγόριθμου (ΓΑ) από παλαιότερες εργασίες του Εργαστηρίου Διοίκησης Επιχειρήσεων και Λήψης Αποφάσεων (Kyriklidis and Dounias, 2014; Kyriklidis et al., 2014).

Πίνακας 7.6: Αποτελέσματα σε μικρά έργα με γνωστές λύσεις

Πρόβλημα	Αντικειμενική Συνάρτηση	Βέλτιστη Τιμή	ΓΑ	Απόδοση ΓΑ	BEH	Απόδοση BEH
1 - Leu et al., 2000	RLI	25	25	74%	25	100%
2 - Leu and Hung, 2002	RLI	11	11	100%	11	100%

Και στα δύο προβλήματα, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) είχε 100% επιτυχία (40 στα 40 ανεξάρτητα πειράματα) στην εύρεση της ήδη γνωστής καλύτερης λύσης. Αντιθέτως, όπως φαίνεται στον Πίνακα 7.6, η απόδοση του Γενετικού Αλγόριθμου ήταν 74% για το πρώτο πρόβλημα. Επιπλέον, αξίζει να σημειωθεί ότι ο μηχανισμός παραγωγής εφικτών λύσεων, που περιγράφεται στην υποενότητα 4.2.2., αποτελεί σημαντικό πλεονέκτημα του τρόπου προσέγγισης του αλγόριθμου σε σχέση με άλλους. Με 20 επαναλήψεις και 20 αναζητητές κατάφερε να βρίσκει πάντα την καλύτερη λύση, χωρίς να δημιουργεί μη

εφικτές λύσεις, ενώ ο Γενετικός Αλγόριθμος (Kyriklidis and Dounias, 2014) χρησιμοποίησε 40 αναζητητές σε 50 επαναλήψεις και έχανε αρκετό υπολογιστικό χρόνο σε μη εφικτές λύσεις.



Εικόνα 7.2: Προφίλ πόρων για το Πρόβλημα 2 (Leu and Hung, 2002)

Ύστερα, ο αλγόριθμος εφαρμόστηκε σε επιλεγμένες περιπτώσεις τεχνητών έργων 90 και 120 δραστηριοτήτων από την βιβλιοθήκη PSPLib. Τα αποτελέσματα παρουσιάζονται στους Πίνακες 7.7 και 7.8. Ο προτεινόμενος αλγόριθμος (BEH) έτρεξε 50 ανεξάρτητες φορές, όπου χρησιμοποιήθηκαν 40 αναζητητές και 50 γενιές σε κάθε ανεξάρτητο πείραμα. Οι τιμές των παραμέτρων επιλέχθηκαν έτσι ώστε να γίνει δίκαια σύγκριση με τα αποτελέσματα του Γενετικού Αλγόριθμου (ΓΑ) και του Υβριδικού Γενετικού Αλγόριθμου (ΥΓΑ) που χρησιμοποιήθηκαν σε παλαιότερη εργασία (Kyriklidis and Dounias, 2016).

Πίνακας 7.7: Αποτελέσματα στα τεχνητά προβλήματα της PSPLib (90 δραστηριοτήτων)

Έργο	BEH			ΥΓΑ			ΓΑ		
	G_f	Std	STEP	G_f	Std	STEP	G_f	Std	STEP
j901_1	63	14,4067	233	57	10,5831	228	61	9,5378	236
j901_2	41	7,4185	237	38	7,9458	243	42	7,4931	235
j901_3	63	15,8973	201	57	13,6071	221	58	13,5003	203
j901_4	64	17,5290	229	53	12,6695	239	55	12,1604	247
j901_5	47	10,2852	214	46	10,4648	238	48	11,2022	219
j901_6	59	15,0633	189	58	13,9444	181	60	14,9280	189
j901_7	52	12,5102	245	50	11,3426	256	50	11,4613	221
j901_8	58	14,2916	286	58	15,5474	302	59	14,8351	289
j901_9	65	17,9966	239	62	17,3777	246	65	16,9910	241
j901_10	51	10,8906	266	52	11,2312	334	54	11,8841	304

Πίνακας 7.8: Αποτελέσματα στα τεχνητά προβλήματα της PSPLib (120 δραστηριοτήτων)

Έργο	BEH	ΥΓΑ	ΓΑ	Έργο	BEH	ΥΓΑ	ΓΑ
j1201_1	62	56	56	j1203_6	58	61	61
j1201_2	67	64	64	j1203_7	115	55	57
j1201_3	65	70	73	j1203_8	71	70	72
j1201_4	63	65	66	j1203_9	59	63	66
j1201_5	58	60	60	j1203_10	59	56	60
j1201_6	73	74	76	j1204_1	76	76	75
j1201_7	58	54	54	j1204_2	58	54	58
j1201_8	68	64	67	j1204_3	70	64	63
j1201_9	69	70	71	j1204_4	74	73	75
j1201_10	63	64	63	j1204_5	80	80	81
j1202_1	81	87	87	j1204_6	60	66	68
j1202_2	68	71	73	j1204_7	70	73	73
j1202_3	84	87	91	j1204_8	65	66	66
j1202_4	76	60	62	j1204_9	66	67	67
j1202_5	72	67	70	j1204_10	63	66	66
j1202_6	75	81	80	j1205_1	62	60	62
j1202_7	62	65	64	j1205_2	73	75	77
j1202_8	67	70	67	j1205_3	69	75	78
j1202_9	73	66	67	j1205_4	54	60	61
j1202_10	85	87	87	j1205_5	66	69	71
j1203_1	80	75	78	j1205_6	74	77	78
j1203_2	57	56	57	j1205_7	70	67	70
j1203_3	59	62	62	j1205_8	74	74	73
j1203_4	69	75	75	j1205_9	58	56	57
j1203_5	66	73	70	j1205_10	61	61	63

Στο κριτήριο STEP, ο προτεινόμενος αλγόριθμος (BEH) παρείχε καλύτερες λύσεις σε 6 από τα 10 προβλήματα, ενώ στις υπόλοιπες περιπτώσεις ήταν καλύτερος από τον έναν εκ των δύο συγκρινόμενων αλγορίθμων ή ίσης απόδοσης (περίπτωση j901_6). Στην ελαχιστοποίηση της τυπικής απόκλισης της κατανομής των πόρων (Std), σε 4 από τις 10 περιπτώσεις ο αλγόριθμος είχε καλύτερη απόδοση από τους άλλους δύο.

Τέλος, στην ελαχιστοποίηση της μέγιστης χρήσης πόρων (G_f) ο αλγόριθμος βρήκε καλύτερη λύση σε 2 περιπτώσεις, όπου στην μια (περίπτωση j901_8) εκ των δύο προσέγγισε την ήδη γνωστή βέλτιστη λύση του Υβριδικού Γενετικού Αλγόριθμου (ΥΓΑ). Σε τρεις (3) ακόμη περιπτώσεις είναι δεύτερος σε απόδοση. Η απουσία στατιστικής ανάλυσης των δύο μεθόδων που χρησιμοποιήθηκαν ως βάση σύγκρισης δεν επιτρέπει την εξαγωγή περαιτέρω συμπερασμάτων. Παρόλα αυτά, ενθαρρυντικό είναι το γεγονός ότι οι λύσεις της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό δεν απέχουν πολύ από αυτές των δύο άλλων μεθόδων.

Στα τεχνητά έργα 120 δραστηριοτήτων, η προτεινόμενη μέθοδος παρείχε καλύτερες λύσεις από τις συγκρινόμενες μεθόδους σε 27 από τα 50 τεχνητά έργα που επιλέχθηκαν, ενώ σε 2 ακόμη περιπτώσεις (j1204_5, j1205_10) είχε εξίσου καλή απόδοση με την καλύτερη συγκρινόμενη μέθοδο. Στις υπόλοιπες περιπτώσεις, η απόδοσή του ήταν ελάχιστα χειρότερη από τον Γενετικό Αλγόριθμο (ΓΑ) και τον Υβριδικό Γενετικό Αλγόριθμο (ΥΓΑ), των οποίων τα αποτελέσματα προέρχονται από τη βιβλιογραφία (Kyriklidis et al., 2014).

7.2.4 Βέλτιστη Κατανομή Πόρων σε έργο κατασκευής Υποσταθμών Υψηλών Τάσεων

Η Κοινοτική Οδηγία 2010/75/EK δρομολόγησε το κλείσιμο αρκετών εργοστασίων παραγωγής ενέργειας με σκοπό τη μείωση των εκπομπών ρύπων. Αυτό είχε ως συνέπεια την αντικατάσταση των κεντρικών μονάδων παραγωγής ενέργειας (που βρίσκονται εκτός αστικού ιστού και μακριά από το δίκτυο) από μικρότερους σταθμούς διανομής κοντά στο δίκτυο. Για τη διασύνδεση της παραγωγής με το δίκτυο χρησιμοποιούνται Υποσταθμοί Υψηλών Τάσεων (High Voltage Substations).

Οι Υποσταθμοί Υψηλών Τάσεων μετατρέπουν την τάση της παραγωγής, της τάξεως των 20 με 30 kV, σε Τάση Δικτύου Διανομής (Transmission Grid Voltage) της τάξης των 150 kV. Για την αποφυγή απώλειας ενέργειας οι Υποσταθμοί Υψηλών Τάσεων κατασκευάζονται κοντά στις μονάδες παραγωγής. Για παράδειγμα, η παραγωγή αιολικής ενέργειας γίνεται συνήθως σε περιοχές με υψηλό υψόμετρο, όπου το αιολικό δυναμικό είναι υψηλότερο. Οι Υποσταθμοί Υψηλών Τάσεων σε αυτή την περίπτωση πρέπει να έχουν κατασκευαστεί κοντά στο αιολικό πάρκο. Αυτό το γεγονός έχει ως συνέπεια το έργο της κατασκευής Υποσταθμών Υψηλών Τάσεων να απαιτεί μετακίνηση προσωπικού, εξοπλισμού και μηχανημάτων σε μεγάλες αποστάσεις. Για την μείωση του κόστους του έργου, πρέπει να γίνει σωστή κατανομή πόρων με σκοπό να μην απαιτούνται πολλές μετακινήσεις εξοπλισμού ή μηχανημάτων από και προς τον χώρο της κατασκευής.

Το έργο κατασκευής των Υποσταθμών Υψηλών Τάσεων (YYT) έχει αρκετές δραστηριότητες που εκτελούνται σειριακά ανά ομάδες εργασιών. Κατ' επέκταση, αυτές οι δραστηριότητες δε θα μπορούσαν να μετακινηθούν χρονικά για την εύρεση της βέλτιστης κατανομής των διαθέσιμων πόρων. Για την απλοποίηση του εν λόγω

έργου, οι δραστηριότητες συμπύχτηκαν σε 40 ομάδες εργασιών, οι οποίες παρουσιάζονται στον Πίνακα 7.9 παρακάτω.

Πίνακας 7.9: Δεδομένα έργου κατασκευής Υποσταθμών Υψηλών Τάσεων

A/A δραστηριότητας	Διάρκεια (μέρες)	Πόροι	Προαπαιτούμενη δραστηριότητα
1	0	0	-
2	270	12	1
3	60	6	2
4	60	4	3FS-30 μέρες
5	270	9	4
6	14	6	5FS-30 μέρες
7	450	16	6
8	420	13	7FS-30 μέρες
9	660	3	6
10	660	6	6
11	60	4	10FS-10 μέρες
12	20	8	5
13	120	9	5
14	270	8	6FS+90 μέρες
15	240	12	7
16	20	11	14
17	90	8	15
18	480	15	5
19	90	8	8
20	20	8	19
21	15	1	15
22	20	1	14FS+30 μέρες
23	15	1	15FS+30 μέρες
24	5	1	16
25	10	1	17
26	5	15	18
27	10	11	17
28	210	3	8FS-60 μέρες
29	90	2	8FS-60 μέρες
30	180	2	23
31	105	6	24
32	105	8	27
33	75	14	22
34	75	2	26
35	75	11	25
36	60	5	11
37	270	14	8FS-30 μέρες
38	105	4	37
39	75	6	38
40	0	0	39

Η ένδειξη FS στην τελευταία στήλη του Πίνακα εκφράζει τη σχέση ολοκλήρωσης-για-την-έναρξη (finish-to-start) της εκάστοτε δραστηριότητας με την

προαπαιτούμενή της. Για παράδειγμα, η ένδειξη 3FS-30 μέρες σημαίνει ότι η δραστηριότητα 4 μπορεί να ξεκινήσει 30 ημέρες πριν την ολοκλήρωση της δραστηριότητας 3 που είναι η προαπαιτούμενή της.

7.2.4.1 Προτεινόμενο υβριδικό σχήμα

Για την επίλυση του παραπάνω έργου χρησιμοποιήθηκε η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH), καθώς και ένα υβριδικό σχήμα που αποτελείται από την BEH και την Προσομοιωμένη Ανόπτωση (ΠΑ). Τα πειράματα έγιναν σε προγραμματιστικό περιβάλλον Matlab σε υπολογιστές με μνήμη 4GB, επεξεργαστή 3.6GHz Intel Core i7 και λειτουργικό Windows 10 Pro και σε έναν υπολογιστή με μνήμη 16GB, επεξεργαστή 3.6GHz Intel Core i7-9700KF και λειτουργικό Windows 10 Pro.

Πίνακας 7.10: Τιμές των παραμέτρων της BEH και του υβριδικού σχήματος BEH-ΠΑ

Παράμετρος	BEH	BEH-ΠΑ
συντελεστής στιγμής επανατοποθέτησης	0,08	0,08
T (θερμοκρασία)	-	1000
Ρυθμός μείωσης θερμοκρασίας T	-	0,95

Όπως αναφέρθηκε και στα προηγούμενα κεφάλαια, η επίτευξη μιας ισορροπίας μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation) είναι στόχος μέγιστης σημασίας. Ο μηχανισμός επανατοποθέτησης των λύσεων που δεν βελτιώνουν τη θέση τους, ο οποίος περιέχεται στην BEH, αποτελεί μια μέθοδο επίτευξης της παραπάνω ισορροπίας. Ωστόσο, αυτή η ισορροπία μπορεί να επιτευχθεί και με την σταδιακή μεταβολή από υψηλό ποσοστό εξερεύνησης σε υψηλό ποσοστό εκμετάλλευσης, όπως συμβαίνει και στην Προσομοιωμένη Ανόπτωση (Simulated Annealing) (Kirkpatrick, Gelatt and Vecchi, 1983).

Το υβριδικό σχήμα που προτάθηκε για τη Βέλτιστη Κατανομή Πόρων στο έργο της κατασκευής Υποσταθμών Υψηλών Τάσεων εισάγει στη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) τον παραπάνω μηχανισμό της Προσομοιωμένης Ανόπτωσης (ΠΑ). Ο μηχανισμός αυτός ονομάζεται κριτήριο Metropolis (Metropolis Criterion) και χρησιμοποιείται για να γίνονται με πιθανότητα p αποδεκτές λύσεις που δεν είναι καλύτερες από την ήδη υπάρχουσα, έτσι ώστε να επιτυγχάνεται μεγαλύτερη διαφοροποίηση στις λύσεις. Η πιθανότητα αυτή υπολογίζεται ως:

$$p = e^{-q_{best} - q_i / T} \quad (7.2.4.1)$$

όπου με q_{best} συμβολίζεται η ποιότητα (με όρους αντικειμενικής συνάρτησης) της καλύτερης λύσης, q_i είναι η ποιότητα της λύσης του αναζητητή i και T είναι η παράμετρος της θερμοκρασίας της ΠΑ, η οποία μειώνεται σταδιακά κατά τη διάρκεια του αλγορίθμου.

Στο προτεινόμενο υβριδικό σχήμα, αυτή η πιθανότητα καθορίζει αν κατά την επανατοποθέτηση του αναζητητή θα δημιουργηθεί μια τυχαία νέα θέση ή αν θα επανατοποθετηθεί ο αναζητητής στον χώρο γύρω από την ήδη γνωστή καλύτερη λύση. Η τυχαία επανατοποθέτηση ενισχύει την εξερεύνηση, ενώ η επανατοποθέτηση με βάση την ήδη γνωστή καλύτερη λύση ενισχύει την εκμετάλλευση. Οπότε, στην αρχή της υβριδικής μεθόδου είναι πιο πιθανό να επανατοποθετηθεί τυχαία ο αναζητητής και άρα υπάρχει υψηλό ποσοστό εξερεύνησης, ενώ σταδιακά αυξάνεται η πιθανότητα να επανατοποθετηθεί γύρω από την καλύτερη γνωστή λύση και άρα αυξάνεται το ποσοστό της εκμετάλλευσης, ενώ μειώνεται αυτό της εξερεύνησης.

Στον αντίποδα, στη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH) το κριτήριο επιλογής του τρόπου επανατοποθέτησης είναι το αν η ποιότητα του αναζητητή που επανατοποθετείται είναι μεγαλύτερη από τη μέση ποιότητα του πληθυσμού, όπως περιγράφεται στο Κεφάλαιο 4.

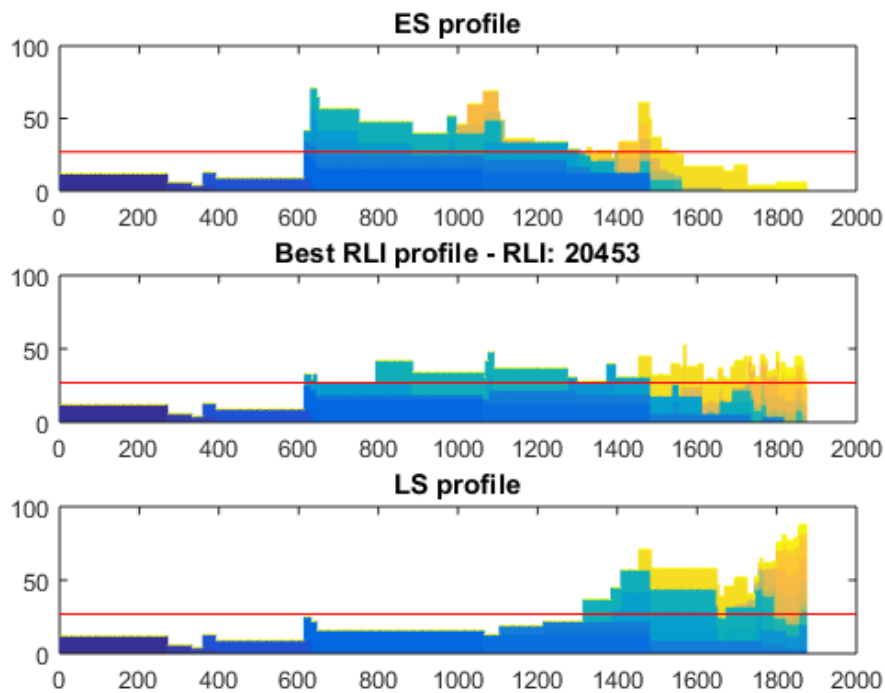
7.2.4.2 Αποτελέσματα

Πέντε διαφορετικά σεντ παραμέτρων επιλέχθηκαν για την παραμετροποίηση τόσο της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (BEH) όσο και του υβριδικού σχήματος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό-Προσομοιωμένης Ανόπτησης (BEH-ΠΑ). Αρχικά, διατηρώντας σταθερό τον πληθυσμό στους 50 αναζητητές (agents) δοκιμάστηκαν διαφορετικές τιμές γενιών: 10, 50 και 100. Η ίδια διαδικασία έγινε διατηρώντας σταθερό τον αριθμό των γενιών στις 50 και δοκιμάζοντας τις προαναφερόμενες τιμές στην παράμετρο του πληθυσμού πλέον. Για τη μέτρηση της στατιστικής απόδοσης κάθε σεντ παραμέτρων πραγματοποιήθηκαν 50 ανεξάρτητα πειράματα για κάθε περίπτωση. Όλες οι υπόλοιπες παράμετροι των αλγορίθμων που συνιστούν το υβριδικό σχήμα παρέμειναν σταθερές σε όλα τα πειράματα και οι τιμές τους παρουσιάζονται στον Πίνακα 7.10.

Πίνακας 7.11: Αποτελέσματα BEH και BEH-ΠΑ για το κριτήριο RLI

	Καλύτερο Προφίλ	Μέση Απόδοση	Τυπική Απόκλιση
Νωρίτερη Έναρξη (ES)	30795	-	-
Αργότερη Έναρξη (LS)	33689	-	-
BEH (10-50)	20613	20848,68	110,616
BEH (50-10)	20671	20972,60	138,341
BEH (50-50)	20578	20749,02	80,9468
BEH (50-100)	20483	20722,48	82,3938
BEH (100-50)	20491	20690,76	79,0816
BEH-ΠΑ (10-50)	20515	20788,38	104,813
BEH-ΠΑ (50-10)	20708	20986,40	106,413
BEH-ΠΑ (50-50)	20517	20694,38	82,8128
BEH-ΠΑ (50-100)	20496	20673,12	91,8252
BEH-ΠΑ (100-50)	20453	20667,82	89,4805

Στους Πίνακες Πίνακας 7.11 και Πίνακας 7.12 παρουσιάζονται τα αποτελέσματα της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΒΕΗ) και του υβριδικού σχήματος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό-Προσομοιωμένης Ανόπτωσης (ΒΕΗ-ΠΑ) σε κάθε σετ παραμέτρων. Οι αριθμοί στην παρένθεση αντιστοιχούν στον πληθυσμό (population) και τις γενιές (generations) σε κάθε σετ. Δηλαδή (50-10) σημαίνει ότι χρησιμοποιήθηκαν 50 αναζητητές και 10 επαναλήψεις. Όπως φαίνεται στους Πίνακες Πίνακας 7.11 και Πίνακας 7.12, τόσο η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (ΒΕΗ) όσο και το υβριδικό σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό-Προσομοιωμένης Ανόπτωσης (ΒΕΗ-ΠΑ) παρείχαν μακράν καλύτερες λύσεις από τις μεθόδους της Ενωρίτερης (Early) και της Αργότερης Έναρξης (Late Start). Όπως ήταν αναμενόμενο, η αύξηση των τιμών των παραμέτρων είχε ως αποτέλεσμα την βελτίωση της απόδοσης και των δύο μεθόδων.



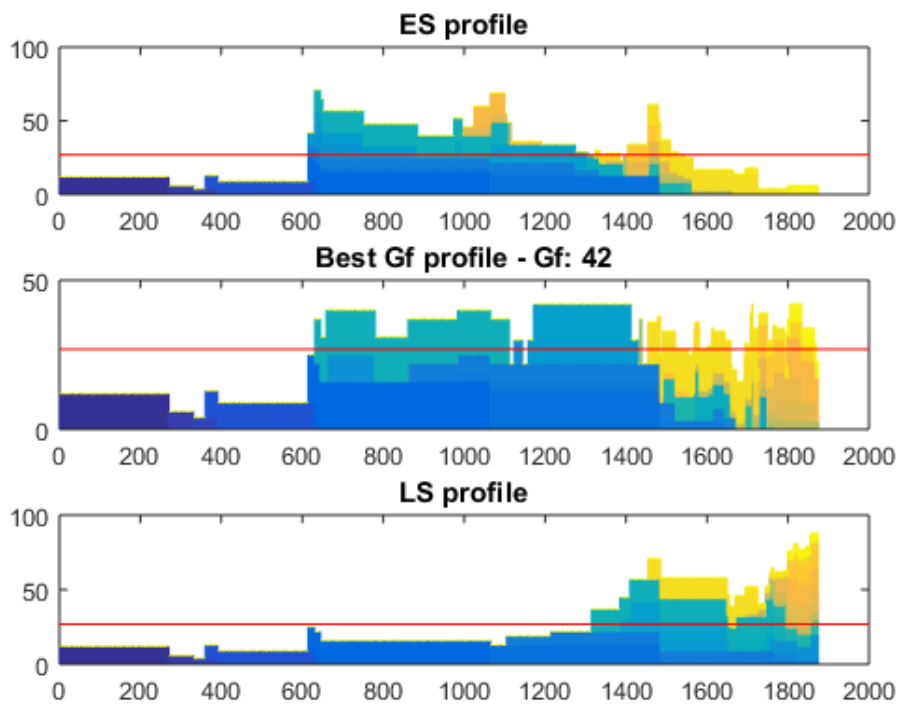
Εικόνα 7.3: Σύγκριση του καλύτερου προφίλ πόρων για το κριτήριο RLI με τα προφίλ της Ενωρίτερης και Αργότερης Έναρξης

Στην Εικόνα 7.3 το προφίλ πόρων της καλύτερης λύσης σύμφωνα με το κριτήριο RLI, η οποία βρέθηκε από το υβριδικό σχήμα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό-Προσομοιωμένης Ανόπτωσης (ΒΕΗ-ΠΑ) με 100 αναζητητές και 50 επαναλήψεις, συγκρίνεται με τα αντίστοιχα προφίλ των μεθόδων της Ενωρίτερης (Early) και της Αργότερης Έναρξης (Late Start). Η κόκκινη γραμμή σε κάθε προφίλ αντικατοπτρίζει την μέση χρήση πόρων στο εν λόγω προφίλ πόρων. Η λύση που επιτεύχθηκε από το υβριδικό σχήμα είναι εμφανές ότι δίνει ένα πιο ομοιόμορφο προφίλ πόρων σε σχέση με τις συγκρινόμενες μεθόδους.

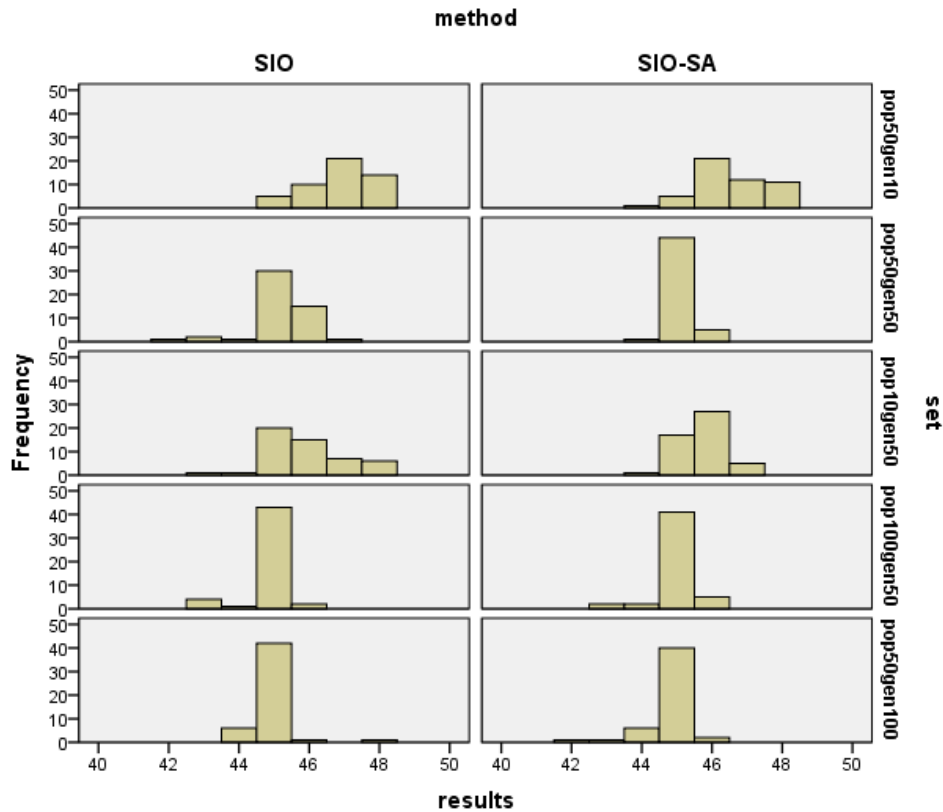
Πίνακας 7.12: Αποτελέσματα ΒΕΗ και ΒΕΗ-ΠΑ για το κριτήριο G_f

	Καλύτερο Προφίλ	Μέση Απόδοση	Τυπική Απόκλιση
Νωρίτερη Έναρξη (ES)	71	-	-
Αργότερη Έναρξη (LS)	88	-	-
ΒΕΗ (10-50)	43	45,88	1,1250
ΒΕΗ (50-10)	45	46,88	0,9304
ΒΕΗ (50-50)	42	45,18	0,8412
ΒΕΗ (50-100)	44	44,96	0,5643
ΒΕΗ (100-50)	43	44,86	0,6003
ΒΕΗ-ΠΑ (10-50)	44	45,72	0,6645
ΒΕΗ-ΠΑ (50-10)	44	46,54	1,0042
ΒΕΗ-ΠΑ (50-50)	44	45,08	0,3371
ΒΕΗ-ΠΑ (50-100)	42	44,82	0,6226
ΒΕΗ-ΠΑ (100-50)	43	44,98	0,5474

Η αντίστοιχη σύγκριση για το κριτήριο G_f φαίνεται στην Εικόνα 7.4, όπου είναι εμφανής κιόλας η διαφοροποίηση των δύο κριτηρίων. Το κριτήριο G_f έχει στόχο την ελαχιστοποίηση της μέσης χρήσης, ενώ το κριτήριο RLI στοχεύει στην ελαχιστοποίηση της απόκλισης των ημερήσιων πόρων από τη μέση χρήση. Το προφίλ της καλύτερης λύσης σύμφωνα με το κριτήριο G_f παρουσιάζει μέγιστη ημερήσια χρήση 42 μονάδες πόρων, ενώ το αντίστοιχο προφίλ σύμφωνα με το κριτήριο RLI επιτυγχάνει βέλτιστη (ελάχιστη) τιμή ίση με 20453.

**Εικόνα 7.4:** Σύγκριση του καλύτερου προφίλ πόρων για το κριτήριο G_f με τα προφίλ της Ενωρίτερης και Αργότερης Έναρξης

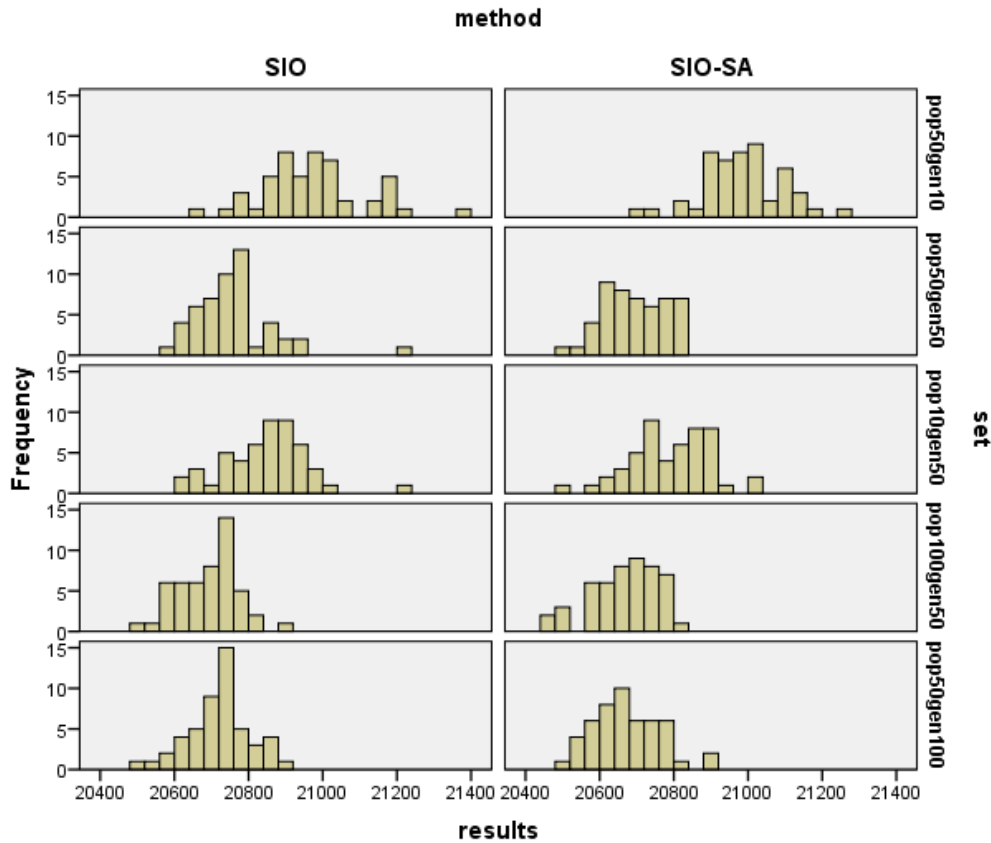
7.2.5 Ανάλυση Παραμέτρων της Προτεινόμενης Μεθόδου στο Πρόβλημα Κατανομής Πόρων



Γράφημα 7-2: Συγκριτικές κατανομές αποτελεσμάτων απλής και υβριδικής μεθόδου για το κριτήριο G_f

Στην επίλυση του πραγματικού έργου κατασκευής Υποσταθμών Υψηλών Τάσεων έγινε ανάλυση παραμέτρων τόσο της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (BEH) όσο και του υβριδικού σχήματος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό-Προσομοιωμένης Ανόπτησης (BEH-ΠΑ), όπως αναφέρεται και παραπάνω. Οι κατανομές των αποτελεσμάτων 50 ανεξάρτητων πειραμάτων για το κριτήριο της ελαχιστοποίησης της μέγιστης χρήσης (G_f - Γράφημα 7-2) και για το κριτήριο της ελαχιστοποίησης της μέσης χρήσης (RLI - Γράφημα 7-3) φαίνονται στις επιμέρους εικόνες της παρούσας υποενότητας.

Όπως ήταν αναμενόμενο, η αύξηση των τιμών των παραμέτρων είχε ως αποτέλεσμα την βελτίωση της απόδοσης και των δύο μεθόδων. Ειδικότερα, όσο αυξάνονται οι αναζητητές (agents) και οι επαναλήψεις (generations), η κατανομή των αποτελεσμάτων αποκτά αρνητική λοξότητα, δηλαδή τα ανεξάρτητα τρεξίματα δίνουν κυρίως μικρότερες τιμές κοντά στο ελάχιστο που έχει επιτευχθεί. Επιπλέον, η αύξηση των δύο προαναφερόμενων παραμέτρων έχει ως αποτέλεσμα λεπτόκυρτες κατανομές, που πρακτικά σημαίνει ότι τα αποτελέσματα των ανεξάρτητων τρεξιμάτων συσσωρεύονται γύρω από τη διάμεσο ή την μέση τιμή τους.



Γράφημα 7-3: Συγκριτικές κατανομές αποτελεσμάτων απλής και υβριδικής μεθόδου για το κριτήριο RLI

7.3 Βελτιστοποίηση Γραμμών Παραγωγής

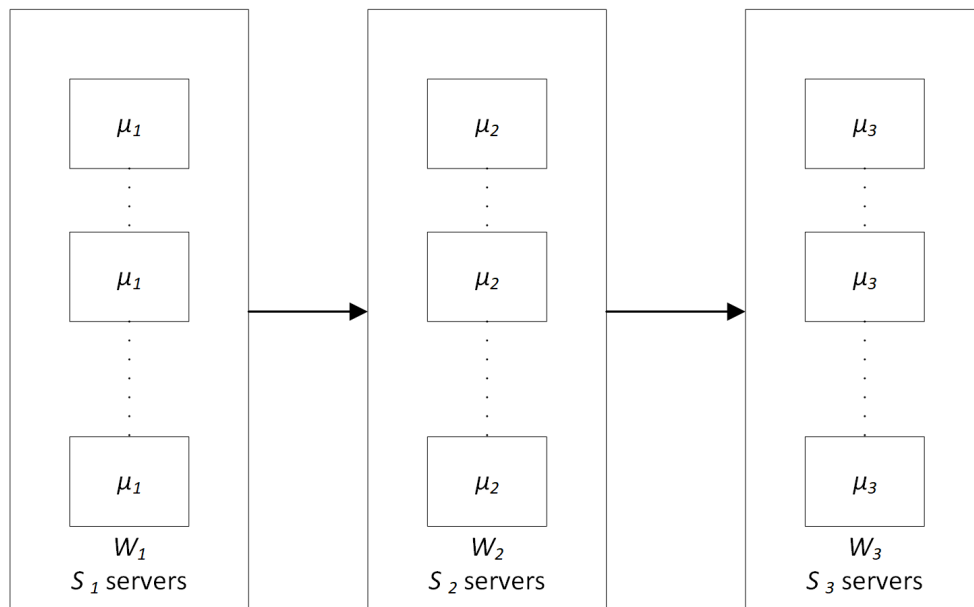
Για την μοντελοποίηση των σειριακών γραμμών παραγωγής (serial production lines) υπάρχουν δύο στρατηγικές: η προσομοίωση (simulation) και η ανάπτυξη ακριβών ή προσεγγιστικών αναλυτικών μεθόδων (accurate or approximate analytical methods) (Boulas, Dounias and Papadopoulos, 2017). Ο στόχος και των δύο στρατηγικών είναι ο υπολογισμός μέτρων απόδοσης (performance metrics) που χρησιμοποιούνται συχνά. Τέτοια μέτρα απόδοσης είναι η ποιότητα (quality), η ταχύτητα (speed), η αξιοπιστία (dependability), η ευελιξία (flexibility) και το κόστος (cost). Οπότε, το αναμενόμενο πλήθος ολοκληρωμένων προϊόντων που παράχθηκαν σε μια μονάδα χρόνου είναι μια βασική μετρική του συστήματος παραγωγής και ονομάζεται διεκπαιρευτική ικανότητα (throughput) ή μέσος ρυθμός παραγωγής (mean production rate ή mean output rate) (Papadopoulos et al., 2009).

Ακριβής αξιολόγηση τέτοιων συστημάτων μπορεί να γίνει μοντελοποιώντας τα ως δίκτυα αναμονής (queuing networks) τα οποία λύνονται με Μαρκοβιανή Ανάλυση υποθέτοντας ότι οι χρόνοι επεξεργασίας ακολουθούν διάφορες κατανομές (Erlang, εκθετική κλπ.), οι οποίες είναι ειδικές περιπτώσεις της κατανομής τύπου φάσης (phase-type distribution).

Ο χώρος καταστάσεων (state space) αυξάνεται εκθετικά, όσο αυξάνεται το πλήθος των σταθμών και η χωρητικότητα των ενταμιευτών. Αυτό έχει ως αποτέλεσμα η Μαρκοβιανή Ανάλυση να καλείται να χειριστεί τεράστια συστήματα εξισώσεων, τα οποία απαιτούν και τεράστια υπολογιστική ισχύ. Για παράδειγμα, για μια γραμμή παραγωγής 14 σταθμών απαιτείται η επίλυση σχεδόν 48 εκατομμυρίων εξισώσεων. Γι' αυτό τα τελευταία χρόνια έχουν επιστρατευθεί Νοήμονες Μέθοδοι (Tsakonas, Papadopoulos and Dounias, 2001; Boulas, Dounias and Papadopoulos, 2017) για την επίλυση μεγαλύτερων συστημάτων.

Χρήσιμη για τους μηχανικούς που σχεδιάζουν γραμμές παραγωγής είναι η ύπαρξη ενός μαθηματικού τύπου (mathematical formula) για να υπολογίζουν την απόδοσή τους, καθώς προσφέρει έναν γρήγορο τρόπο υπολογισμού της διεκπαιρευτικής ικανότητας των γραμμών παραγωγής (throughput) χωρίς να απαιτεί προγραμματιστικές γνώσεις και υπολογιστική ισχύ. Το πρόβλημα είναι ότι τέτοιοι τύποι υπάρχουν μόνο για μικρά συστήματα, όπως μικρές γραμμές παραγωγής όπου οι σταθμοί παραγωγής αποτελούνται από έναν διακομιστή (server).

Η πρωτοτυπία της εργασίας (Boulas, Tzanetos and Dounias, 2018) ήταν το γεγονός ότι για πρώτη φορά παράχθηκαν μαθηματικά μοντέλα υπολογισμού της διεκπαιρευτικής ικανότητας (throughput) μιας γραμμικής γραμμής παραγωγής με πανομοιότυπους παράλληλους διακομιστές (servers) σε κάθε σταθμό παραγωγής (workstation). Η προσεγγιστική σχέση (approximate formula) δημιουργείται από τον Γενετικό Προγραμματισμό (Genetic Programming) (Koza, 1992) και βελτιώνεται από τη Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (Sonar Inspired Optimization).

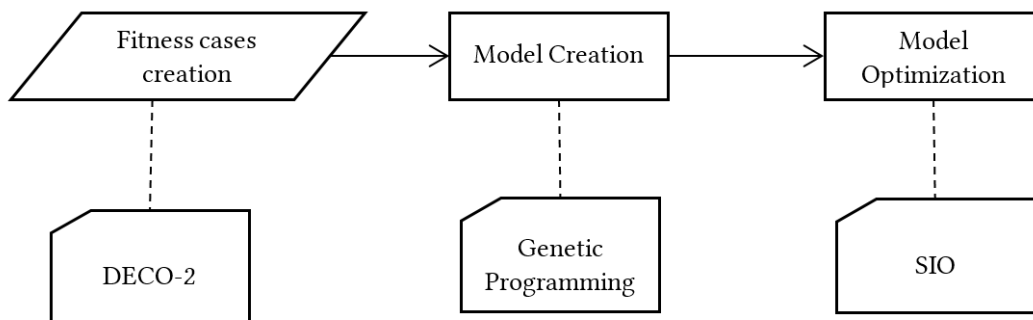


Εικόνα 7.5: Παράδειγμα σειριακής γραμμής παραγωγής (serial production line)

7.3.1 Μέθοδος Επίλυσης

Το πρόβλημα περιγράφεται από την σταδιακή αύξηση κάποιων παραμέτρων λειτουργίας της γραμμής: το πλήθος των σταδίων (stages) K , με $K \in \{2,3\}$, τον μέσο ρυθμό παραγωγής (mean production rate) μ_i , όπου $i \in \{1, \dots, K\}$ και $\mu \in \{0.9, 0.91, \dots, 1.1\}$ και τον αριθμό των πανομοιότυπων μηχανών ανά στάδιο S_i , όπου $S_i \in \{1, 2, \dots, 5\}$ και μια περίπτωση όπου $S_i \in \{1, 2, \dots, 20\}$. Ο αριθμός των διακομιστών (servers) μπορεί να διαφέρει από επίπεδο σε επίπεδο, αλλά όλοι ακολουθούν την εκθετική κατανομή με μέσο ρυθμό εξυπηρέτησης (mean service rate) ίσο με $1/\mu$. Ένα παράδειγμα μιας σειριακής γραμμής παραγωγής φαίνεται στην Εικόνα 7.5.

Το σετ εκπαίδευσης (training set) δημιουργείται με την χρήση του αλγόριθμου DECO-2 (Papadopoulos et al., 2009). Αφού παραχθεί το μοντέλο (formula) από τον Γενετικό Προγραμματισμό, εισάγεται στην Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό με σκοπό να βελτιώσει τις τιμές των συντελεστών (σταθερές). Η διαδικασία φαίνεται σχεδιαγραμματικά στην Εικόνα 7.6.



Εικόνα 7.6: Η μέθοδος επίλυσης του προβλήματος

Ο αλγόριθμος (SIO) θεωρεί κάθε αναζητητή ως το διάνυσμα λύσης:

$$X_i = \{c^1, c^2, c^3, \dots, c^n\}$$

όπου c^d είναι η εκάστοτε σταθερά του μοντέλου που παράγει ο Γενετικός Προγραμματισμός, με N συμβολίζεται το πλήθος των αναζητητών, με n συμβολίζεται το πλήθος των σταθερών του μοντέλου και ισχύουν: $d \in 1, 2, \dots, n$ και $i \in 1, 2, \dots, N$.

Όπως αναφέρθηκε και παραπάνω, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό έχει αποδειχθεί καλό εργαλείο βελτιστοποίησης, αλλά η βελτιστοποίηση προϋποθέτει να υπάρχουν μεταβλητές απόφασης και ένα εύρος τιμών για την κάθε μια, όπου αναζητάται ο καλύτερος συνδυασμός τιμών που επιλύει ένα αντίστοιχο πρόβλημα. Για τον σκοπό αυτό δημιουργήθηκαν τα άνω και κάτω όρια για κάθε διάσταση, βάσει των τιμών c^d της λύσης του Γενετικού Προγραμματισμού:

$$upper_bound_d = GPsol_d + GPsol_d \cdot 0.1 \quad (7.3.1.1)$$

$$lower_bound_d = GPsol_d - GPsol_d \cdot 0.1 \quad (7.3.1.2)$$

Σύμφωνα με την βιβλιογραφία (Li et al., 2016a), ένας χώρος αναζήτησης (search space) της τάξεως του $\pm 10\%$ δημιουργήθηκε για να βελτιώσει την απόδοση της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό, αντί να δοθεί αυθαίρετα ένα εύρος.

Τρεις περιπτώσεις μελετήθηκαν και παράχθηκαν μαθηματικά μοντέλα υπολογισμού της διεκπαιρευτικής ικανότητας (throughput) των γραμμών παραγωγής:

- **Περίπτωση 1:** γραμμή παραγωγής με δύο σταθμούς (workstations). Ο μέσος ρυθμός παραγωγής κάθε μηχανής κυμαίνεται από 0.9 έως 1.1 και ο αριθμός των πανομοιότυπων μηχανών σε κάθε στάδιο από 1 έως 5.
- **Περίπτωση 2:** γραμμή παραγωγής με τρεις σταθμούς. Όλα τα στάδια έχουν τον ίδιο αριθμό μηχανών και κάθε μηχανή έχει τον ίδιο μέσο ρυθμό εξυπηρέτησης. Σε αυτή την περίπτωση, ο μέσος ρυθμός παραγωγής κυμαίνεται από 0.9 έως 1.1 και ο αριθμός των πανομοιότυπων μηχανών ανά στάδιο κυμαίνεται από 1 έως 20.
- **Περίπτωση 3:** γραμμή παραγωγής με τρεις σταθμούς, όπου ο μέσος ρυθμός παραγωγής κάθε μηχανής κυμαίνεται από 0.9 έως 1.1 και ο αριθμός των πανομοιότυπων μηχανών σε κάθε στάδιο από 1 έως 5, όπως και στην Περίπτωση 1.

7.3.2 Περίπτωση 1

Το μοντέλο της Περίπτωσης 1 φαίνεται στην Εξίσωση (7.3.2.1). Σημειωτέον ότι η συγκεκριμένη περίπτωση έχει επιλυθεί με ακρίβεια από τους Diamantidis, Papadopoulos and Heavey (2007) και είναι το δομικό στοιχείο (structure element) αποσύνθεσης του αλγόριθμου DECO-2, από το οποίο πήρε και το όνομά του. Δίνει ακριβή αποτελέσματα, δηλαδή, στην περίπτωση της γραμμής παραγωγής με δύο στάδια.

$$X_{case1} = \left(\frac{w_0 \cdot s_2}{\left(w_1 \cdot \mu_2 + \frac{\mu_1^2 \cdot \left(\frac{w_2 \cdot s_2}{w_3 \cdot s_1} \right)^2 \cdot w_4}{w_5 \cdot \mu_2} \right)} + \frac{\mu_1 \cdot \left(\frac{w_6 \cdot s_2}{w_7 \cdot s_1} \right)^2 \cdot w_8}{w_9 \cdot \mu_2} + \frac{w_{10} \cdot s_2}{w_{11} \cdot \mu_2} + w_{12} \right) \quad (7.3.2.1)$$

7.3.3 Περίπτωση 2

Αυτή η περίπτωση γραμμής παραγωγής είναι η απλούστερη μεταξύ των τριών. Έχει δύο ανεξάρτητες μεταβλητές (μ και s) και περιγράφει ένα πλήρως ισορροπημένο

σύστημα. Η συγκεκριμένη περίπτωση συναντάται συχνά στην βιβλιογραφία. Η σχέση (7.3.3.1) υπολογίζει την διεκπαιρευτική ικανότητα (throughput) αυτής της περίπτωσης:

$$X_{case2} = \left(\frac{w_0 \cdot \mu \cdot \left(w_1 \cdot s + \frac{w_2}{w_3} \right)}{\left(\frac{w_4}{(w_5 \cdot s + w_6)} + w_7 \right)} \cdot w_8 + w_9 \right) \quad (7.3.3.1)$$

7.3.4 Περίπτωση 3

Η πιο περίπλοκη εκ των τριών περιπτώσεων που παρουσιάζονται. Έξι ανεξάρτητες μεταβλητές συνδυάστηκαν μεταξύ τους για να παράξουν πάνω από ένα εκατομμύριο περιπτώσεις καταλληλότητας (fitness cases). Για υπολογιστικούς λόγους, μειώθηκε το δείγμα εκπαίδευσης (training set) στο 0.4% των 700,000 περιπτώσεων καταλληλότητας.

$$X_{case3} = \exp(A - B) \cdot w_{15} + w_{16} \quad (7.3.4.1)$$

όπου:

$$A = \ln \left(\left(\ln(w_0 \cdot s_3) \right)^2 \right) - \frac{\frac{(\ln(w_1 \cdot s_3))^2}{(w_2 \cdot s_1)^2}}{\left(\frac{\frac{w_3 \cdot s_3}{w_4 \cdot s_2}}{\left(\left(w_5 - \ln \left(\left(\ln(w_6 \cdot s_3) \right)^2 \right) \right) + \ln(w_7 \cdot s_3) \right)} + w_8 \cdot \mu_3 \right)}$$

$$B = \left(\frac{\frac{w_9 \cdot s_3}{w_{10} \cdot s_2}}{\left((w_{11} - w_{12} \cdot \mu_3) + \ln(\ln(w_{13} \cdot s_3)) \right)} + w_{14} \cdot \mu_3 \right)$$

7.3.5 Αποτελέσματα

Στον Πίνακα 7.13 παρουσιάζονται τα στατιστικά αποτελέσματα του Γενετικού Προγραμματισμού. Παρατηρεί κανείς ότι επιτυγχάνει πολύ χαμηλές τιμές Μέσου Απόλυτου Σφάλματος (Mean Absolute Error – MAE), τόσο στο δείγμα εκπαίδευσης (training set) όσο και στο δείγμα ελέγχου (test set). Επιπλέον, τα παραγόμενα μοντέλα έχουν πολύ καλή ακρίβεια (R^2) και στα δύο δείγματα.

Οι τρεις γραμμές του Πίνακα 7.13 που έχουν σημειωθεί με πορτοκαλί χρώμα περιέχουν τις τιμές Μέσου Απόλυτου Σφάλματος (MAE), του Συντελεστή Συσχέτισης Pearson (R^2) και του Μέσου Σχετικού Σφάλματος (Average Relative Error) στο δείγμα ελέγχου (test set) των δεδομένων. Αυτές οι τιμές συγκρίνονται με τις αντίστοιχες που έδωσαν οι τελικές λύσεις του βελτιωμένου μοντέλου, αφού εφαρμόστηκε και ο αλγόριθμος Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό, όπως φαίνονται στον Πίνακα 7.14, τα σφάλματα έχουν μειωθεί, ιδιαίτερα στην δεύτερη Περίπτωση του απλού μοντέλου, όπου η ακρίβεια του μοντέλου έγινε σχεδόν απόλυτη ($R^2 = 1.000000$). Σημειωτέον ότι με τον συντελεστή συσχέτισης Pearson μετράται η συσχέτιση του προσεγγιστικού μαθηματικού μοντέλου με τις περιπτώσεις καταλληλότητας (fitness cases) του DECO-2.

Πίνακας 7.13: Αποτελέσματα του Γενετικού Προγραμματισμού

	Περίπτωση 1	Περίπτωση 2	Περίπτωση 3
Πλήθος περιπτώσεων δείγματος εκπαίδευσης	5512	373	28000
Πλήθος περιπτώσεων δείγματος ελέγχου	2724	42	357626
Συνολικό πλήθος περιπτώσεων	11025	415	1157625
Πλήθος ανεξάρτητων μεταβλητών	4	2	6
Μήκος δενδρικής μορφής λύσης (Model Length)	31	21	48
Βάθος δενδρικής μορφής λύσης (Model Depth)	10	9	16
MAE δείγματος εκπαίδευσης	0.0251	0.0028	0.0963
MAE δείγματος ελέγχου	0.0244	0.0024	0.0965
R^2 δείγματος εκπαίδευσης	0.999	0.9999	0.979
R^2 δείγματος ελέγχου	0.999	0.9999	0.979
Μέσο σχετικό σφάλμα (εκπαίδευση)	1.4604%	0.1110%	7.3405%
Μέσο σχετικό σφάλμα (έλεγχος)	1.4555%	0.0624%	7.3500%

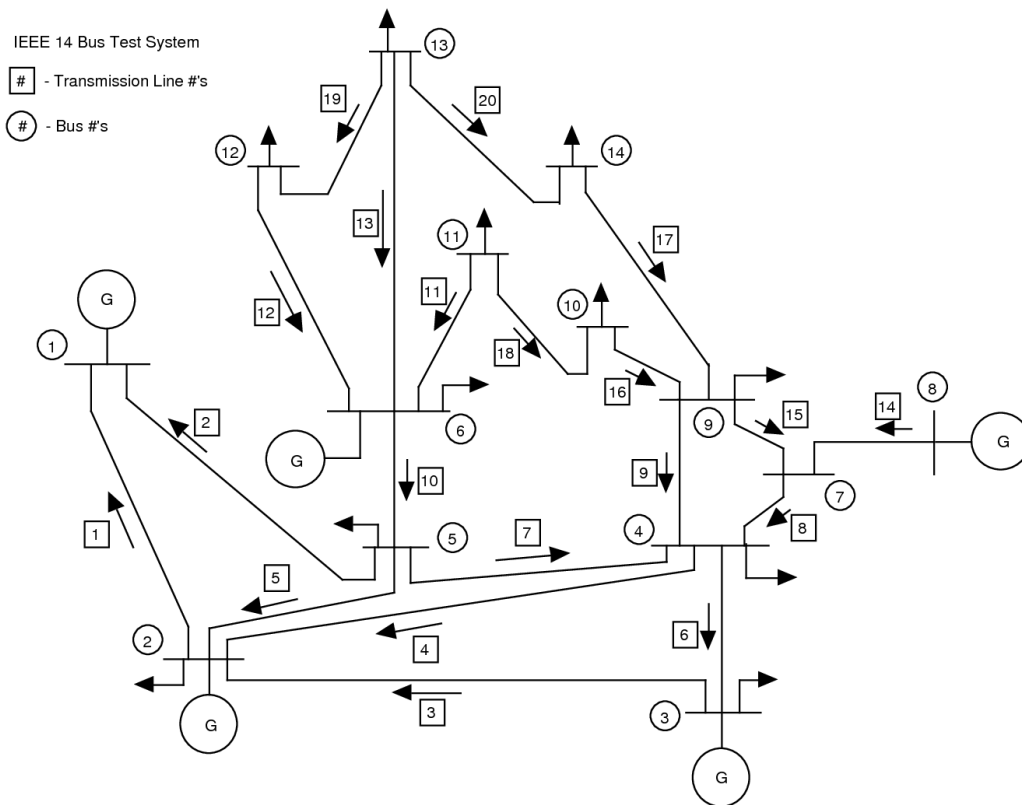
Γενικά, η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό αποδεικνύεται χρήσιμο εργαλείο τοπικής αναζήτησης, αλλά και γενικότερα ένα σημαντικό στοιχείο (component) υβριδικών σχημάτων στην περαιτέρω βελτίωση λύσεων, καθώς εξερευνά σε μεγαλύτερο βαθμό τον χώρο γύρω από κάθε ήδη υπάρχουσα λύση.

Πίνακας 7.14: Αποτελέσματα Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό

	Περίπτωση 1	Περίπτωση 2	Περίπτωση 3
MAE δείγματος ελέγχου	0.0244	0.0020	0.0842
R^2 δείγματος ελέγχου	0.999556	1.000000	0.991275
Μέσο σχετικό σφάλμα (test)	0.0146	0.0006	0.0598

7.4 Βέλτιστη Οικονομική Κατανομή Ενέργειας

Διάφορα προβλήματα βελτιστοποίησης έχουν διατυπωθεί στον χώρο της Ενέργειας, όπως η Βέλτιστη Ροή Ενέργειας (Optimal Power Flow) και η Οικονομική Κατανομή Φορτίου (Economic Load Dispatch ή ELD). Τα προβλήματα αυτού του είδους συνήθως διερευνούν τη βέλτιστη στρατηγική που πρέπει να ακολουθήσει ο λειτουργός ενός δικτύου για την καλύτερη απόδοση του δικτύου ή την ελαχιστοποίηση του κόστους που μπορεί να προέρχεται από την παραγωγή ενέργειας.



Εικόνα 7.7: Δοκιμαστικό σύστημα 14 ζυγών του Ινστιτούτου Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE)

Για παράδειγμα, η Εικόνα 7.7 απεικονίζει το δοκιμαστικό σύστημα (test system) 14 ζυγών (buses) του Ινστιτούτου Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών (IEEE) (Al-Roomi, 2015). Η ένδειξη G στους μεγάλους κύκλους απεικονίζει τις γεννήτριες του συστήματος, ενώ οι μικροί κύκλοι αντιστοιχούν στους ζυγούς (buses), στους οποίους μπορεί να είναι συνδεδεμένες μία ή περισσότερες γραμμές μετάδοσης ενέργειας (transmission lines). Η γεννήτρια που είναι συνδεδεμένη στο ζυγό εγχέει ρεύμα στο σύστημα, ενώ τα βέλη δείχνουν το φορτίο (load) που απορροφούν ρεύμα από τον συγκεκριμένο ζυγό. Οι ζυγοί του παραδείγματος που είναι συνδεδεμένοι με γεννήτριες ονομάζονται ζυγοί με ελεγχόμενη τάση ή αλλιώς ζυγοί παραγωγής (ζυγοί PV). Οι ζυγοί 4, 5, 7, 9, 10, 11,

12, 13 και 14 είναι Ζυγοί φορτίου (ζυγοί PQ), καθώς δεν υπάρχει παραγωγή αλλά μόνο φορτία.

Επειδή οι απώλειες μεταφοράς (transmission losses) δεν μπορούν να προσδιοριστούν εκ των προτέρων, η παραγωγή που απαιτείται για την τροφοδοσία συγκεκριμένου φορτίου στο δίκτυο δεν είναι δυνατόν να προκαθοριστεί στο σύνολό της. Γι' αυτό ένας εκ των ζυγών που είναι συνδεδεμένος με γεννήτρια ορίζεται ως ζυγός αναφοράς (ή ζυγός ταλάντωσης) και συνήθως είναι αυτός με την μεγαλύτερη παραγωγή πραγματικής ισχύος (Μελανίτη, 2013).

Το πρόβλημα της Οικονομικής Κατανομής Ενέργειας περιγράφει την εύρεση της βέλτιστης παραγόμενης ισχύος κάθε γεννήτριας με στόχο την ελαχιστοποίηση του κόστους καυσίμου (fuel cost). Το εν λόγω πρόβλημα έχει διάφορες παραλλαγές, ανάλογα τους αντικειμενικούς στόχους που ορίζονται, όπως είναι η ελαχιστοποίηση των εκπομπών ρύπων (emissions) και η ελαχιστοποίηση των απωλειών ενέργειας.

7.4.1 Αντικειμενικοί Στόχοι του Προβλήματος

Συμβολίζοντας με N_G τον συνολικό αριθμό των γεννητριών του συστήματος και P_{G_i} την παραγόμενη ισχύ (active power generation) της i -οστής θερμικής γεννήτριας, ισχύουν οι παρακάτω αντικειμενικές συναρτήσεις, οι οποίες εκφράζουν διαφορετικούς αντικειμενικούς στόχους βέλτιστης διαχείρισης ενέργειας.

Ελαχιστοποίηση του Κόστους Καυσίμου: Ο υπολογισμός του κόστους καυσίμων δίνεται από τη σχέση (7.4.1.1) (Das and Suganthan, 2010):

$$FC(P_{G_i}) = \sum_{i=1}^{N_G} (a_i + b_i P_{G_i} + c_i P_{G_i}^2) \quad (7.4.1.1)$$

όπου a_i (\$/h), b_i (\$/hMW) και c_i (\$/hMW²) είναι οι συντελεστές κόστους καυσίμου (fuel cost coefficients) της i -οστής θερμικής γεννήτριας και $FC(P_{G_i})$ είναι το συνολικό κόστος καυσίμου για γεννήτρες που δουλεύουν με ένα είδος καυσίμου και υπολογίζεται σε δολλάρια ανά ώρα (\$/h).

Ελαχιστοποίηση των Εκπομπών Ρύπων: Στα εργοστάσια παραγωγής ενέργειας κατά την καύση των καυσίμων που χρησιμοποιούνται για την παραγωγή ενέργειας εκλύονται διάφοροι ρύποι, όπως οξείδια του θείου (SO_x), οξείδια του Αζώτου (NO_x) κλπ. Η σχέση (7.4.1.2) χρησιμοποιείται στη βιβλιογραφία (Rahman and Mohamad-Saleh, 2018), για τον υπολογισμό των εκπομπών ρύπων:

$$EC(P_{G_i}) = \sum_{i=1}^{N_G} (a_i + \beta_i P_{G_i} + \gamma_i P_{G_i}^2 + \xi_i e^{\lambda_i P_{G_i}}) \quad (7.4.1.2)$$

όπου, $\gamma_i(\text{ton/h})$, $\beta_i(\text{ton/hMW})$, $\alpha_i(\text{ton/hMW}^2)$, $\xi_i(\text{ton/h})$ και $\lambda_i(1/\text{MW})$ είναι οι συντελεστές εκπομπών (emission coefficients) της i -οστής θερμικής γεννήτριας και με $EC(P_{G_i})$ ισυμβολίζεται το άθροισμα όλων των ειδών ρύπων, μετρούμενο σε τόνους ανά ώρα (ton/h).

Ελαχιστοποίηση των Απωλειών Ενέργειας: Όπως είναι λογικό, κάθε θερμική μονάδα έχει απώλειες ενέργειας κατά τη λειτουργία της. Αυτές οι απώλειες ενέργειας εκφράζονται μέσω της σχέσης (7.4.1.3), όπως φαίνεται παρακάτω:

$$PL(P_{G_i}) = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_{G_i} B_{ij} P_{G_j} + \sum_{i=1}^{N_G} B_{oi} P_{G_i} + B_{oo} \quad (7.4.1.3)$$

όπου, B_{ij} είναι το ij -οστό στοιχείο του πίνακα συντελεστών απωλειών ενέργειας (loss coefficient matrix). B_{oi} είναι το i -οστό στοιχείο του διανύσματος συντελεστών απωλειών ενέργειας (loss coefficient vector). B_{oo} είναι η σταθερά απωλειών ενέργειας (loss coefficient constant) και με $PL(P_{G_i})$ συμβολίζεται το σύνολο των απωλειών ενέργειας (network losses), το οποίο μετράται σε Μεγαβάτ (MW).

7.4.2 Περιορισμοί του Προβλήματος

7.4.2.1 Ισοζύγιο ισχύος (power balance)

Ο συγκεκριμένος περιορισμός βασίζεται στην αρχή της ισορροπίας μεταξύ της συνολικής παραγόμενης ενέργειας και του αθροίσματος της συνολικής ζήτησης (demand) του συστήματος και των απωλειών ενέργειας που αναφέρονται παραπάνω. Ισχύει:

$$\sum_{i=1}^{N_G} P_{G_i} = P_D + P_L \quad (7.4.2.1)$$

όπου, P_D είναι η συνολική ζήτηση του συστήματος σε Μεγαβάτ (MW), P_L είναι οι συνολικές απώλειες μετάδοσης ενέργειας (power transmission losses) του δικτύου σε Μεγαβάτ (MW) and N_G είναι ο συνολικός αριθμός των γεννητριών του συστήματος.

7.4.2.2 Περιορισμός παραγόμενης ισχύος

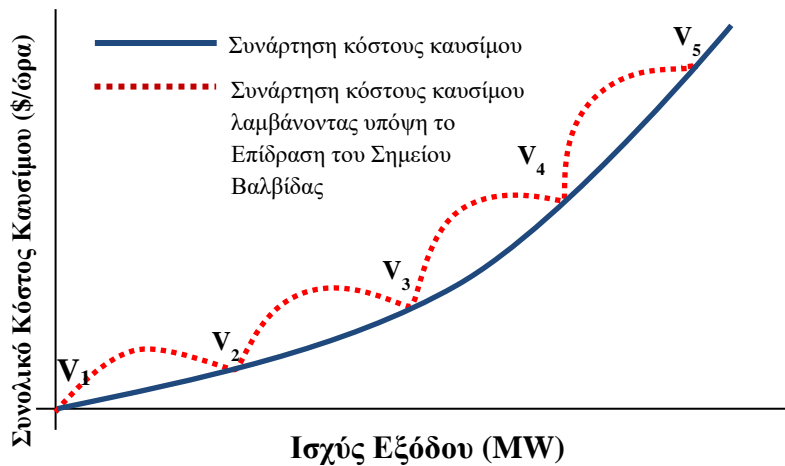
Η παραγόμενη ισχύς κάθε μονάδας δεν μπορεί να παραβιάζει τα όρια ισχύος της εκάστοτε μονάδας:

$$P_{G_i}^{\min} \leq P_{G_i} \leq P_{G_i}^{\max} \quad (7.4.2.2)$$

όπου, P_{G_i} είναι η παραγόμενη ισχύς της i -οστής θερμικής γεννήτριας, ενώ με $P_{G_i}^{\min}$ και $P_{G_i}^{\max}$ συμβολίζονται το κάτω και άνω όριο παραγόμενης ισχύος, αντίστοιχα.

7.4.3 Επίδραση του Σημείου Βαλβίδας (Valve-point effect)

Στις μονάδες παραγωγής ενέργειας χρησιμοποιούνται βαλβίδες ελέγχου (control valves) για τον έλεγχο της ροής του ατμού με σκοπό τη ρύθμιση της παραγόμενης ισχύος. Κάθε φορά που μια τέτοια βαλβίδα ανοίγει, μια ξαφνική αύξηση της απώλειας ενέργειας παρατηρείται με αποτέλεσμα να δημιουργούνται διακυμάνσεις στη συνάρτηση υπολογισμού του κόστους καυσίμου, όπως φαίνονται και στην Εικόνα 7.8. Αυτό το φαινόμενο αποκαλείται Επίδραση του Σημείου Βαλβίδας (Valve-Point Effect ή VPE). Στην Εικόνα 7.8 απεικονίζεται η σύγκριση της συνάρτησης κόστους καυσίμου με και χωρίς την Επίδραση του Σημείου Βαλβίδας (Chakraborty et al., 2011).



Εικόνα 7.8: Η επίδραση του φαινομένου της βαλβίδας στη συνάρτηση κόστους

Σύμφωνα με τους L. S. Coelho and V. C. Mariani (2006), η συνάρτηση του κόστους καυσίμου μεταβάλλεται με βάση την καμπύλη διακυμάνσεων που φαίνεται παραπάνω, ώστε η μοντελοποίηση της συνάρτησης να είναι πιο ακριβής. Αυτή η καμπύλη παρουσιάζει μη-γραμμικότητα υψηλότερης τάξεως (higher order nonlinearity) και ασυνέχεια (discontinuity) λόγω της Επίδρασης του Σημείου Βαλβίδας και μπορεί να μοντελοποιηθεί με τη χρήση της ημιτονοειδούς συνάρτησης. Κατ' επέκταση, η πολυωνυμική συνάρτηση κόστους που περιγράφεται στην σχέση (7.4.1.2), εφόσον ληφθεί υπόψη την Επίδραση του Σημείου Βαλβίδας μετατρέπεται ως ακολούθως:

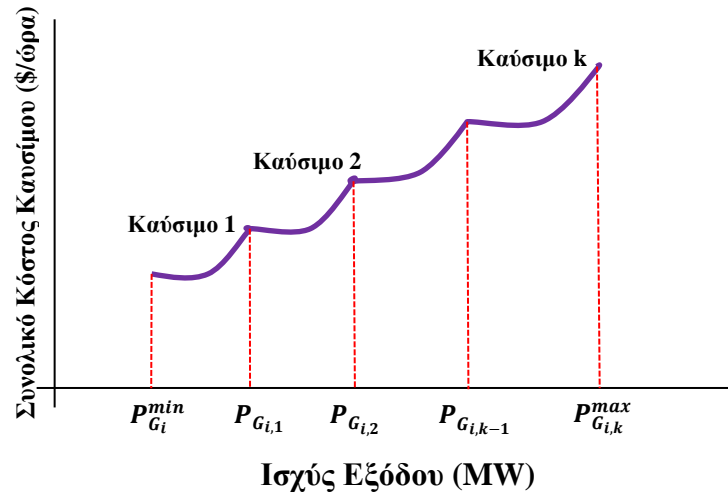
$$C_{VPE}(P_{G_i}) = \sum_{i=1}^{N_G} (a_i P_{G_i}^2 + b_i P_{G_i} + c_i P_{G_i}^2) + \left| d_i \sin(e_i (P_{G_i}^{min} - P_{G_i})) \right| \quad (7.4.3.1)$$

όπου, a_i , b_i , c_i , d_i και e_i είναι οι συντελεστές κόστους καυσίμου (fuel cost coefficients) της i -οστής θερμικής γεννήτριας λαμβάνοντας υπόψη και την Επίδραση του Σημείου Βαλβίδας (Valve-point Effect - VPE). Με $C_{VPE}(P_{G_i})$ συμβολίζεται το συνολικό κόστος καυσίμου για γεννήτριες που δουλεύουν με ένα είδος καυσίμου,

λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας (Valve-point Effect) και υπολογίζεται σε δολλάρια ανά ώρα (\$/h).

7.4.4 Λειτουργία πολλαπλών καυσίμων (multi-fuel operation)

Στην πραγματικότητα στα εργοστάσια παραγωγής ενέργειας δεν χρησιμοποιείται ένα είδος καυσίμου, αλλά αρκετά διαφορετικά είδη. Οπότε, η συνάρτηση κόστους μετατρέπεται σε τμηματική (piecewise) πολυωνυμική εξίσωση, όπου λαμβάνονται υπόψη τα διάφορα καύσιμα, όπως φαίνεται στην Εικόνα 7.9.



Εικόνα 7.9: Η συνάρτηση κόστους του προβλήματος στην περίπτωση πολλαπλών καυσίμων

Κατ' επέκταση, η σχέση (7.4.3.1) μετατρέπεται ως ακολούθως (A. Bhattacharya and P. K. Chattopadhyay, 2010):

$$C_{VPE_MFO}(P_{G_i}) = \sum_{i=1}^{N_G} (a_{i,k}P_{G_i}^2 + b_{i,k}P_{G_i} + c_{i,k}P_{G_i}^2) + |d_{i,k} \sin(e_{i,k}(P_{G_i}^{min} - P_{G_i}))| \quad (7.4.4.1)$$

δεδομένου $P_{G_{i,k}}^{min} \leq P_{G_i} \leq P_{G_{i,k}}^{max}$, για την επιλογή του καυσίμου $k = 1, 2, \dots, N_F$. Στην παραπάνω σχέση (7.4.4.1), με $a_{i,k}$, $b_{i,k}$, $c_{i,k}$, $d_{i,k}$ και $e_{i,k}$ συμβολίζονται οι συντελεστές (coefficients) της συνάρτησης κόστους της i γεννήτριας για το είδος καυσίμου k . P_{G_i} είναι η παραγόμενη ισχύς της i -οστής θερμικής γεννήτριας, ενώ $P_{G_{i,k}}^{min}$ και $P_{G_{i,k}}^{max}$ είναι τα αντίστοιχα άνω και κάτω όρια που περιγράφονται στη σχέση (7.4.2.2), που όμως αντιστοιχούν στο εκάστοτε καύσιμο k . N_F είναι το πλήθος των διαθέσιμων καυσίμων και με $C_{VPE_MFO}(P_{G_i})$ συμβολίζεται το συνολικό κόστος καυσίμου για τις γεννήτριες του συστήματος, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας (Valve-point Effect) και την λειτουργία πολλαπλών καυσίμων (Multi-fuel operation - MFO), και υπολογίζεται σε δολλάρια ανά ώρα (\$/h).

7.4.5 Οικονομική Κατανομή Ενέργειας σε ένα τεχνητό σύστημα 10 γεννητριών

Για την διερεύνηση της απόδοσης της προτεινόμενης μεθόδου επιλέχθηκε ένα σύστημα 10 γεννητριών (10-unit test system) από τη βιβλιογραφία (Naderi et al., 2017), του οποίου η ζήτηση είναι 2000 MW. Στους Πίνακες 7.15 έως 7.17 περιλαμβάνονται τα δεδομένα του προβλήματος.

Πίνακας 7.15: Δεδομένα συστήματος 10 γεννητριών

Γεννήτρια	1	2	3	4	5	6	7	8	9	10
$P_{G_i}^{min}$	10	20	47	20	50	70	60	70	135	150
$P_{G_i}^{max}$	55	80	120	130	160	240	300	340	470	470

Οι συντελεστές του κόστους καυσίμου (fuel cost coefficients) και εκπομπών (emission coefficients) για κάθε γεννήτρια, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας, που χρησιμοποιούνται στις σχέσεις (7.4.1.1), (7.4.1.2) και (7.4.3.1), φαίνονται παρακάτω στον Πίνακα 7.16:

Πίνακας 7.16: Συντελεστές του κόστους καυσίμου (fuel cost coefficients) και εκπομπών (emission coefficients) για κάθε γεννήτρια, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας

Γεννήτρια	a_i	b_i	c_i	d_i	e_i	α_i	β_i	γ_i	ξ_i	λ_i
1	1000.403	40.5407	0.12951	33	0.0174	360.0012	-3.9864	0.04702	0.25475	0.01234
2	950.606	39.5804	0.10908	25	0.0178	350.0056	-3.9524	0.04652	0.25475	0.01234
3	900.705	36.5104	0.12511	32	0.0162	330.0056	-3.9023	0.04652	0.25163	0.01215
4	800.705	39.5104	0.12111	30	0.0168	330.0056	-3.9023	0.04652	0.25163	0.01215
5	756.799	38.5390	0.15247	30	0.0148	13.8593	0.3277	0.00420	0.24970	0.01200
6	451.325	46.1592	0.10587	20	0.0163	13.8593	0.3277	0.00420	0.24970	0.01200
7	1243.531	38.3055	0.03546	20	0.0152	40.2669	-0.5455	0.00680	0.24800	0.01290
8	1049.998	40.3965	0.02803	30	0.0128	40.2669	-0.5455	0.00680	0.24990	0.01203
9	1658.569	36.3278	0.02111	60	0.0136	42.8955	-0.5112	0.00460	0.25470	0.01234
10	1356.659	38.2704	0.01799	40	0.0141	42.8955	-0.5112	0.00460	0.25470	0.01234

Παρακάτω δίνεται ο πίνακας συντελεστών απωλειών ενέργειας (loss coefficient matrix) B_{ij} , το διάνυσμα συντελεστών απωλειών ενέργειας (loss coefficient vector) B_{oi} , καθώς και η σταθερά απωλειών ενέργειας (loss coefficient constant) B_{oo} του υπό μελέτη συστήματος:

$$B_{ij} = \begin{bmatrix} 0.000049 & 0.000014 & 0.000015 & 0.000015 & 0.000016 & 0.000017 & 0.000017 & 0.000018 & 0.000019 & 0.000020 \\ 0.000014 & 0.000045 & 0.000016 & 0.000016 & 0.000017 & 0.000015 & 0.000015 & 0.000016 & 0.000018 & 0.000018 \\ 0.000015 & 0.000016 & 0.000039 & 0.000010 & 0.000012 & 0.000012 & 0.000014 & 0.000014 & 0.000016 & 0.000016 \\ 0.000015 & 0.000016 & 0.000010 & 0.000040 & 0.000014 & 0.000010 & 0.000011 & 0.000012 & 0.000014 & 0.000015 \\ 0.000016 & 0.000017 & 0.000012 & 0.000014 & 0.000035 & 0.000011 & 0.000013 & 0.000013 & 0.000015 & 0.000016 \\ 0.000017 & 0.000015 & 0.000012 & 0.000010 & 0.000011 & 0.000036 & 0.000012 & 0.000012 & 0.000014 & 0.000015 \\ 0.000017 & 0.000015 & 0.000014 & 0.000011 & 0.000013 & 0.000012 & 0.000038 & 0.000016 & 0.000016 & 0.000018 \\ 0.000018 & 0.000016 & 0.000014 & 0.000012 & 0.000013 & 0.000012 & 0.000016 & 0.000040 & 0.000015 & 0.000016 \\ 0.000019 & 0.000018 & 0.000016 & 0.000014 & 0.000015 & 0.000014 & 0.000016 & 0.000015 & 0.000042 & 0.000019 \\ 0.000020 & 0.000018 & 0.000016 & 0.000015 & 0.000016 & 0.000015 & 0.000018 & 0.000016 & 0.000019 & 0.000044 \end{bmatrix}$$

$$B_{oi} = 0, B_{oo} = 0$$

Οι συντελεστές του κόστους καυσίμου (fuel cost coefficients) ανά τύπο καυσίμου, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας, που χρησιμοποιούνται στη σχέση (7.4.4.1), φαίνονται παρακάτω στον Πίνακα 7.17:

Πίνακας 7.17: Συντελεστές του κόστους καυσίμου (fuel cost coefficients) ανά τύπο καυσίμου, λαμβάνοντας υπόψη την Επίδραση του Σημείου Βαλβίδας

Γενήτρια	Παραγόμενη ισχύς - Καύσιμα							Τύπος καυσίμου	Συντελεστές κόστους				
	$P_{G_i}^{min}$	F_1	P_1	F_2	P_2	F_3	$P_{G_i}^{max}$		a_i	b_i	c_i	d_i	e_i
1	100		196		-		250	1	26.97	-0.39750	0.002176	0.026970	-3.9750
		1		2		-		2	21.13	-0.30590	0.001861	0.021130	-3.0590
2	50		114		157		230	1	118.40	-1.26900	0.004194	0.118400	-12.6900
		2		3		1		2	1.865	-0.03988	0.001138	0.001865	-0.3988
								3	13.65	-0.19800	0.001620	0.013650	-1.9800
3	200		332		388		500	1	39.79	-0.31160	0.001457	0.039790	-3.1160
		1		3		2		2	-59.14	0.48640	0.00001176	-0.059140	4.8640
								3	-2.875	0.03389	0.0008035	-0.002876	0.3389
4	99		138		200		265	1	1.983	-0.03114	0.001049	0.001983	-0.3114
		1		2		3		2	52.85	-0.63480	0.002758	0.052850	-6.3480
								3	266.80	-2.33800	0.005935	0.266800	-23.3800
5	190		338		407		490	1	13.92	-0.08733	0.001066	0.013920	-0.8733
		1		2		3		2	99.76	-0.52060	0.001597	0.099760	-5.2060
								3	-53.99	0.44620	0.0001498	-0.053990	4.4620
6	85		138		200		265	1	52.85	-0.63480	0.002758	0.052850	-6.3480
		2		1		3		2	1.983	-0.03114	0.001049	0.001983	-0.3114
								3	266.80	-2.33800	0.005935	0.266800	-23.3800
7	200		331		391		500	1	18.93	-0.13250	0.001107	0.018930	-1.3250
		1		2		3		2	43.77	-0.22670	0.001165	0.043770	-2.2670
								3	-43.35	0.35590	0.0002454	-0.043350	3.5590
8	99		138		200		265	1	1.983	-0.03114	0.001049	0.001983	-0.3114
		1		2		3		2	52.85	-0.63480	0.002758	0.052850	-6.3480
								3	266.80	-2.33800	0.005935	0.266800	-23.3800
9	130		213		370		440	1	88.53	-0.56750	0.001554	0.088530	-5.6750
		3		1		3		2	15.30	-0.04514	0.007033	0.014230	-0.1817
								3	14.23	-0.01817	0.0006121	0.014230	-0.1817
10	200		362		407		490	1	13.97	-0.09938	0.001102	0.013970	-0.9938
		1		3		2		2	-61.13	0.50840	0.00004164	-0.061130	5.0840
								3	46.71	-0.20240	0.001137	0.046710	-2.0240

Στη βιβλιογραφία, οι αντικειμενικοί στόχοι του προβλήματος της Οικονομικής Κατανομής Φορτίου λύνονται είτε ξεχωριστά είτε συνδυαστικά, με τη βοήθεια μεθόδων όπως η μέθοδος αθροιστικής στάθμισης (weighted sum method) και τα διαγράμματα Pareto. Στον Πίνακα 7.18 φαίνονται συγκεντρωτικά οι αντικειμενικοί στόχοι και οι ιδιαιτερότητες κάθε επιμέρους προβλήματος, όπως αυτό περιγράφεται στην βιβλιογραφία (Naderi et al., 2017).

Τα προβλήματα I, II και III προσεγγίζουν την Οικονομική Κατανομή Φορτίου βάσει μιας εκ των τριών αντικειμενικών συναρτήσεων που περιγράφονται στην ενότητα 7.4.1, ενώ στο πρόβλημα IV οι επιμέρους αντικειμενικές συναρτήσεις συγχωνεύονται σε μια συντελώντας ένα πρόβλημα πολλών στόχων (multi-objective).

Στόχος του προβλήματος I είναι η ελαχιστοποίηση του κόστους καυσίμου σύμφωνα με τη σχέση (7.4.1.1). Το συγκεκριμένο πρόβλημα χωρίζεται σε τρεις επιμέρους περιπτώσεις (cases): στην Περίπτωση I.I δεν λαμβάνεται κανένα φαινόμενο υπόψη, ενώ στις Περιπτώσεις I.II και I.III λαμβάνονται υπόψη το Φαινόμενο της Βαλβίδας (ΦτΒ) και η Λειτουργία Πολλαπλών Καυσίμων (ΛΠΚ). Στην Περίπτωση I.III λαμβάνονται υπόψη και οι Απώλειες Δικτύου (ΑΔ), οι οποίες σχετίζονται με τις απώλειες του συστήματος. Λαμβάνοντας υπόψη τα ΦτΒ και ΛΠΚ, ο χώρος λύσεων του αρχικού προβλήματος γίνεται μη-ομαλός (non-smooth) και μη-κυρτός (non-convex) μετατρέποντας το πρόβλημα βελτιστοποίησης σε NP-δύσκολης κλάσης από NP.

Πίνακας 7.18: Περιορισμοί και εκδοχές του προβλήματος Οικονομικής Κατανομής Ενέργειας για το σύστημα 10 γεννητριών

Προβλήματα - Περιπτώσεις	Αντικειμενικοί Στόχοι	ΦτΒ	ΛΠΚ	ΑΔ
I.I	Ελαχιστοποίηση Κόστους Καυσίμου	-	-	-
I.II	Ελαχιστοποίηση Κόστους Καυσίμου	✓	✓	-
I.III	Ελαχιστοποίηση Κόστους Καυσίμου	✓	✓	✓
II	Ελαχιστοποίηση Εκπομπών Ρύπων	-	-	✓
III	Ελαχιστοποίηση Απωλειών Ενέργειας	-	-	-
IV	Πολυ-κριτήριο Πρόβλημα	-	-	-

Ο στόχος του προβλήματος II είναι η ελαχιστοποίηση των εκπομπών ρύπων, ενώ του προβλήματος III είναι η ελαχιστοποίηση των απωλειών ενέργειας. Στο πρόβλημα IV επιλύονται όλοι οι παραπάνω στόχοι., δηλαδή η ελαχιστοποίηση του κόστους καυσίμου, η ελαχιστοποίηση των εκπομπών ρύπων και η ελαχιστοποίηση των απωλειών ενέργειας, συντελώντας έτσι το πρόβλημα Οικονομικής Κατανομής Φορτίου και Εκπομπών (Economic Emissions Load Dispatch problem ή EELD) (Apostolopoulos and Vlachos, 2011). Το συγκεκριμένο πρόβλημα χαρακτηρίζεται ως μη-ομαλό (non-smooth), μη-κυρτό (non-convex) και πολλαπλών στόχων (multi-objective). Ωστόσο, το Φαινόμενο της Βαλβίδας (ΦτΒ) και η Λειτουργία Πολλαπλών Καυσίμων (ΛΠΚ) δεν λαμβάνονται υπόψη σε αυτή τη μορφή του προβλήματος στη βιβλιογραφία (Naderi et al., 2017) και για λόγους σύγκρισης αποτελεσμάτων, αυτή η τακτική ακολουθείται και στην παρούσα διατριβή.

Για την επίλυση των παραπάνω προβλημάτων με τη χρήση της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό, κατασκευάστηκαν έξι (6) διαφορετικές αντικειμενικές συναρτήσεις, οι οποίες αντιστοιχούν σε κάθε πρόβλημα και περίπτωση που φαίνεται στον Πίνακα 7.18 και δίνονται στις παρακάτω σχέσεις:

$$obj_{I,I} = FC(P_{G_i}) + 10^3 \cdot PBP \quad (7.4.5.1)$$

$$obj_{I,II} = C_{VPE_MFO}(P_{G_i}) + 10^1 \cdot PBP \quad (7.4.5.2)$$

$$obj_{I,III} = C_{VPE_MFO}(P_{G_i}) + 10^3 \cdot PBP + PL(P_{G_i}) \quad (7.4.5.3)$$

$$obj_{II} = EC(P_{G_i}) + 10^4 \cdot PBP + 10^2 \cdot PL(P_{G_i}) \quad (7.4.5.4)$$

$$obj_{III} = PL(P_{G_i}) + PBP \quad (7.4.5.5)$$

$$obj_{IV} = FC(P_{G_i}) + EC(P_{G_i}) + 10^5 \cdot PBP + 10^2 \cdot PL(P_{G_i}) \quad (7.4.5.6)$$

όπου το PBP είναι μια συνάρτηση ποινής (penalty function), η οποία χειροτερεύει την ποιότητα των λύσεων που παραβιάζουν τον περιορισμό του ισοζυγίου ισχύος (power balance constraint) και υπολογίζεται ως:

$$PBP = \left| P_D - \sum_{i=1}^{N_G} P_{G_i} \right| \quad (7.4.5.7)$$

Ο πολλαπλασιασμός με κάποια δύναμη του δέκα (10) έγινε για να έχουν όλοι οι στόχοι την ίδια τάξη μεγέθους και να συνεισφέρουν εξίσου στην τελική λύση.

Πίνακας 7.19: Κατανομή Ενέργειας σύμφωνα με τα αποτελέσματα της μεθόδου για κάθε πρόβλημα

Πρόβλημα Περίπτωση	I			II	III	IV
	I.I	I.II	I.III			
Ζήτηση του δικτύου (MW)	2000	2700	2700	2000	2000	2000
P_{G_1}	54.1063	217.2060	222.8991	54.8694	46.1135	51.5205
P_{G_2}	78.2936	212.6991	211.7118	73.4562	61.2648	79.3155
P_{G_3}	111.7484	287.0443	273.7992	90.7497	118.6768	87.6005
P_{G_4}	76.3568	239.1194	239.6446	100.1967	128.2961	83.0033
P_{G_5}	71.8717	270.5148	277.7134	137.7294	159.4203	103.8117
P_{G_6}	71.4365	240.9868	242.8708	221.6755	237.5198	141.1924
P_{G_7}	294.0659	292.5837	295.6449	255.1432	293.6623	286.3655
P_{G_8}	329.2705	237.8868	239.0937	290.3642	318.8012	306.7620
P_{G_9}	449.5824	425.0959	422.5314	391.4659	407.1916	440.4434
$P_{G_{10}}$	463.2645	276.8645	274.1013	384.3498	229.0538	419.9852
Συνολική παραγωγή P_G (MW)	2000.00	2700.00	2829.20	2000.00	2000.00	2000.00
Εκπομπές Ρύπων (lb/h)	-	-	-	3746.3581	-	3950.12
Απώλειες Ενέργειας (MW)	-	-	129.1903	75.2671	72.5526	78.2415
Συνολικό Κόστος (\$/h)	106130.50	624.2600	624.2703	-	-	107220.20

Για κάθε ένα από τα προβλήματα που αναφέρονται παραπάνω έγιναν 40 ανεξάρτητα πειράματα με σκοπό τη μέτρηση της στατιστικής απόδοσης της προτεινόμενης μεθόδου. Τα αποτελέσματα συγκρίνονται με τα αντίστοιχα δύο υβριδικών σχημάτων που αποτελούνται από Μεθόδους Εμπνευσμένες από τη Φύση. Συγκεκριμένα, αυτά είναι το Υβριδικό σχήμα Βελτιστοποίησης Σμήνους Σωματιδίων και Διαφορικής Εξέλιξης βασισμένο σε Ασάφια (Fuzzy Based Hybrid Particle Swarm Optimization-Differential Evolution – ΥΒΣΣ-ΔΕΒΑ) και το Υβριδικό σχήμα Βελτιστοποίησης Σμήνους Σωματιδίων και Διαφορικής Εξέλιξης (Hybrid Particle Swarm Optimization-Differential Evolution – ΥΒΣΣ-ΔΕ). Επίσης, ως μέθοδος σύγκρισης χρησιμοποιείται και ο καθιερωμένος (established) αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization – ΒΣΣ). Οι τρεις προαναφερόμενες μέθοδοι πέτυχαν τα καλύτερα αποτελέσματα στη βιβλιογραφία (Naderi et al., 2017) επιλύοντας το συγκεκριμένο πρόβλημα.

Πίνακας 7.20: Συγκριτικά αποτελέσματα στο πρόβλημα της Οικονομικής Κατανομής Φορτίου

Αλγόριθμος	I			II	III
	I.I	I.II	I.III		
BEH	106130.50	624.2600	624.2703	3746.3581	72.5526
ΥΒΣΣ-ΔΕΒΑ	106170.09	623.8224	700.0118	3920.4819	68.7708
ΥΒΣΣ-ΔΕ	-	624.1034	-	3921.6618	69.1269
ΒΣΣ	-	627.6902	-	3929.4431	76.8925
TNΔHop	-	626.24	-	-	-
ΕΠ	-	626.26	-	-	-
ΜΙΔ	-	625.18	-	-	-
ΜΚΤΓΑ II	-	-	-	4130.20	-
ΕΑΔPar 2	-	-	-	4109.10	-

Επιπρόσθετα, μερικές κλασικές μέθοδοι όπως τα Τεχνητά Νευρωνικά Δίκτυα Hopfield (Hopfield Neural Networks – TNΔHop) με τεχνικές προσαρμογής μεροληψίας (bias adjustment), ο Εξελικτικός Προγραμματισμός (Evolutionary Programming – ΕΠ), η Μέθοδος Ιεραρχικής Δομής (Hierarchical Structure Method – ΜΙΔ), ο Μη-Κυριαρχικός Ταξινομικός Γενετικός Αλγόριθμος II (Non-dominated Sorting Genetic Algorithm II – ΜΚΤΓΑ II) και ο Εξελικτικός Αλγόριθμος Δύναμης Pareto 2 (Strength Pareto Evolutionary Algorithm 2 – ΕΑΔPar 2) χρησιμοποιούνται ως μέθοδοι αναφοράς (benchmark). Οι παράμετροι όλων των προαναφερόμενων μεθόδων παρουσιάζονται λεπτομερώς στην εργασία των (Naderi et al., 2017). Τα συγκριτικά αποτελέσματα παρουσιάζονται στον Πίνακα 7.20.

Όλα τα πειράματα έγιναν σε προγραμματιστικό περιβάλλον Matlab σε υπολογιστές με μνήμη 4GB, επεξεργαστή 3.6GHz Intel Core i7 και λειτουργικό Windows 10 Pro. Οι παράμετροι της Βελτιστοποίησης Εμπνευσμένης από τον

Ηχοεντοπισμό (ΒΕΗ), καθώς και των μεθόδων που χρησιμοποιήθηκαν ως σύγκριση φαίνονται στον Πίνακα 7.21.

Πίνακας 7.21: Τιμές των παραμέτρων της ΒΕΗ και των συγκρινόμενων μεθόδων

Παράμετρος	ΒΕΗ	ΥΒΣΣ-ΔΕΒΑ	ΥΒΣΣ-ΔΕ	ΒΣΣ
Πληθυσμός	50	300	300	300
Γενιές	1000	1000	1000	1000
μ	0.05	-	-	-

7.5 Συμπεράσματα

Τα Κεφάλαια 6 και 7 αποσκοπούν στην εφαρμογή της προτεινόμενης μεθοδολογίας σε προβλήματα βελτιστοποίησης με υψηλή πολυπλοκότητα. Στο Κεφάλαιο 6 αποδείχθηκε η αποδοτικότητα του προτεινόμενου αλγορίθμου σε προβλήματα γνωστής καλύτερης λύσης και σε προβλήματα κλάσης P. Το παρόν κεφάλαιο παρουσιάζει τις εφαρμογές της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (ΒΕΗ) σε πραγματικά προβλήματα κλάσεων NP, NP-πληρότητας και NP-δύσκολα.

Εφαρμογές από το φάσμα των προβλημάτων που καλείται να επιλύσει ένας μηχανικός μελετήθηκαν και επιλύθηκαν επιτυχώς από την προτεινόμενη μέθοδο βελτιστοποίησης. Συγκεκριμένα, μελετήθηκε ένα πρόβλημα Χρηματοοικονομικής Μηχανικής (Financial Engineering), ένα πρόβλημα Μηχανικής της Διοίκησης (Management Engineering) όπου επιλύθηκαν πολλές περιπτώσεις τεχνητών δεδομένων και μία περίπτωση ενός πραγματικού έργου, ένα πρόβλημα Επιχειρησιακής Έρευνας (Operational Research) και ένα πρόβλημα Ενέργειας.

Από τον χώρο της Χρηματοοικονομικής Μηχανικής μελετήθηκε το πρόβλημα της Βελτιστοποίησης Χαρτοφυλακίου (Portfolio Optimization) και αναπτύχθηκε μια υβριδική μέθοδος, η οποία περιέχει τον προτεινόμενο αλγόριθμο της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό. Τα αποτελέσματα της υβριδικής μεθόδου παρείχαν λύσεις υψηλότερης ποιότητας σε σχέση με παλαιότερες εργασίες.

Στο πρόβλημα της Βέλτιστης Διαχείρισης Πόρων (Optimal Resource Management), το οποίο ανήκει στο πεδίο της Μηχανικής της Διοίκησης, μελετήθηκε και αναπτύχθηκε μια προσαρμοσμένη εκδοχή της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό, καθώς κι ένα υβριδικό σχήμα αυτής. Η προτεινόμενη μέθοδος παρείχε καλύτερες λύσεις στις περισσότερες περιπτώσεις τεχνητών έργων σε σχέση με τις συγκρινόμενες μεθόδους της βιβλιογραφίας. Επιπλέον, τόσο η προσαρμοσμένη εκδοχή όσο και το υβριδικό σχήμα που την αποτελεί κατάφεραν να επιλύσουν ένα πραγματικό έργο κατασκευής Υποσταθμών Υψηλών Τάσεων. Επιπλέον, υπό ανάπτυξη βρίσκεται η δημιουργία ενός Γραφικού Περιβάλλοντος Διεπαφής Χρήστη

(Graphic User Interface) με σκοπό την χρήση της μεθόδου που προτείνεται στην παρούσα διατριβή στην βέλτιστη κατανομή πόρων.

Στην περίπτωση της Βελτιστοποίησης Γραμμών Παραγωγής, η προτεινόμενη μέθοδος της παρούσας διατριβής χρησιμοποιήθηκε για την βελτίωση των συντελεστών του μοντέλου που παράγαγε ο Γενετικός Προγραμματισμός.

Τέλος, μελετήθηκε και επιλύθηκε ένα πρόβλημα Βέλτιστης Κατανομής Ενέργειας, το οποίο περιέχει πολλαπλούς στόχους, όπως η ελαχιστοποίηση κόστους καυσίμων, η ελαχιστοποίηση εκπομπών ρύπων και η ελαχιστοποίηση απωλειών ενέργειας. Τόσο στους επιμέρους στόχους όσο και στο πολύ-κριτηριακό πρόβλημα, η προτεινόμενη μέθοδος παρείχε λύσεις ίσης ή υψηλότερης ποιότητας με αυτές των συγκρινόμενων μεθόδων.

8. Ανακεφαλαίωση - Γενικά Συμπεράσματα

8.1 Σύνοψη Έρευνας

Την τελευταία εικοσαετία έχει αυξηθεί ραγδαία ο ρυθμός εμφάνισης μιας κατηγορίας τεχνικών που συγκαταλέγονται στους Εξελικτικούς Αλγόριθμους, επονομαζόμενες ως Υπολογιστικές Μέθοδοι Εμπνευσμένες από τη Φύση (Nature Inspired Computational Methods), οι οποίες είναι εμπνευσμένες από στρατηγικές που ακολουθούν οργανισμοί στη φύση για την επίλυση βασικών προβλημάτων τους, όπως η εύρεση τροφής και η επιλογή βιότοπων. Στόχος τους είναι η προσομοίωση της στρατηγικής από την οποία έχουν εμπνευστεί με τέτοιο τρόπο που να δημιουργηθεί μια αλγοριθμική (βηματική δηλαδή) διαδικασία που να επιλύει προβλήματα βελτιστοποίησης.

Ένα καίριο ζήτημα που προκύπτει είναι το γεγονός ότι η πληθώρα των μεθόδων που εμφανίζονται τα τελευταία χρόνια συνήθως είναι αντιγραφή κάποιων καθιερωμένων (established) μεθόδων και κατ' επέκταση δεν προσφέρουν βελτίωση στον τρόπο προσέγγισης των προβλημάτων ή γρηγορότερες και αυτοματοποιημένες μεθόδους, απλά κακές αναπαραγωγές παλαιότερων μεθόδων κουβαλώντας μαζί και τις ατέλειές (drawbacks) τους.

Στην παρούσα διατριβή συλλέχθηκαν και μελετήθηκαν τα μοντέλα των Εμπνευσμένων από τη Φύση μεθόδων. Επιπλέον, αναπτύχθηκε μια νέα μέθοδος εμπνευσμένη από ένα φαινόμενο της φύσης. Για να διερευνηθεί η αποτελεσματικότητά της, μελετήθηκαν και επιλύθηκαν διάφορα προβλήματα βελτιστοποίησης από το φάσμα των προβλημάτων που καλείται να λύσει ένας Μηχανικός Οικονομίας και Διοίκησης, αλλά και γενικότερα από το χώρο των προβλημάτων μηχανικού.

Η προτεινόμενη μέθοδος ελαχιστοποιεί την παρέμβαση του χρήστη μέσω μηχανισμών αυτορρύθμισης και έχει μελετηθεί με τέτοιο τρόπο που να μπορεί να προσεγγίσει διαφορετικά είδη προβλημάτων, χωρίς να απαιτεί τεράστια παραμετροποίηση ή εκ νέου προγραμματισμό. Επιπλέον, προτείνονται υβριδικά μοντέλα που περιέχουν την προτεινόμενη μέθοδο με σκοπό να βελτιώσουν την απόδοσή της και να αντιμετωπιστούν η αδυναμίες της.

8.2 Κύρια σημεία πρωτοτυπίας της διατριβής

Τα σημαντικότερα σημεία συνεισφοράς της παρούσας διατριβής παρουσιάζονται παρακάτω:

1. Έγινε για πρώτη φορά μια πλήρης χαρτογράφηση του χώρου των μεθόδων που είναι εμπνευσμένες από τη φύση (Κεφάλαιο 2). Επιπλέον, προτάθηκε ένας τρόπος κατηγοριοποίησης για αυτές τις μεθόδους, η οποία βασίστηκε σε κοινά χαρακτηριστικά των αλγορίθμων, αλλά και του πεδίου από το οποίο αντλούν την έμπνευσή τους.

2. Με σκοπό να διερευνηθεί τι είδους προβλήματα μπορούν να λύσουν οι Αλγόριθμοι που είναι Εμπνευσμένοι από τη Φύση, μελετήθηκε η χρησιμότητα τους τόσο ανά υπο-κατηγορία, όσο και συνολικά (Κεφάλαιο 2). Επιπλέον, έγινε μια ανάλυση του είδους των προβλημάτων που χρησιμοποιούνται σε συνάρτηση με τα ιδιαίτερα χαρακτηριστικά (π.χ. υποκατηγορία βάσει πεδίου έμπνευσης) που έχουν οι μέθοδοι.
3. Δομήθηκε ένα πλαίσιο εργασίας (framework) στην ανάπτυξη, την βελτίωση και στην αξιολόγηση των Νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση (Κεφάλαιο 3), το οποίο εστιάζει κατά κύριο λόγο στα χαρακτηριστικά εκείνα που καθιστούν έναν τέτοιο αλγόριθμο αποδοτικό. Επιπλέον, παρουσιάζεται η γενική διαγραμματική απεικόνιση και ο γενικός ψευδοκώδικας των Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση (Κεφάλαιο 2).
4. Παρουσιάζεται μια νέα μέθοδος βελτιστοποίησης (Κεφάλαιο 4), η οποία ανήκει στην ευρύτερη κατηγορία των μεθόδων που είναι Εμπνευσμένες από τη Φύση (Nature Inspired Methods). Η μέθοδος αυτή δομήθηκε με στόχο την ελαχιστοποίηση παραμετροποίησης από τον χρήστη, αλλά και με σκοπό την αντιμετώπιση του προβλήματος του εγκλωβισμού σε τοπικά βέλτιστα (local optima trapping). Αυτός ο στόχος επιτεύχθηκε μέσα από την εφαρμογή μηχανισμών αυτο-ρύθμισης (auto-tuning) που εισήχθησαν στη διαδικασία της μεθόδου.
5. Οι κύριοι μηχανισμοί της μεθόδου που προτείνεται στην παρούσα διατριβή, αποτελούν σημαντική συνεισφορά στο πεδίο της Επιστήμης Υπολογιστών (Computer Science) και της Υπολογιστικής Νοημοσύνης (Computational Intelligence), καθώς μπορούν να χρησιμοποιηθούν σε άλλες νοήμονες μεθόδους, ώστε να αντωμετωπιστεί το πρόβλημα της πρόωρης σύγκλισης (premature convergence) και να επιτευχθεί η κατάλληλη ισορροπία μεταξύ εξερεύνησης (exploration) και εκμετάλλευσης (exploitation).
6. Επιπλέον, μελετήθηκαν διάφορα προβλήματα Λήψης Απόφασης από τον ευρύτερο χώρο των προβλημάτων Μηχανικού, ώστε να προταθούν τροποποιημένες εκδοχές της προτεινόμενης μεθόδου για διάφορα είδη προβλημάτων (Κεφάλαια 6 και 7).
7. Παράλληλα, μελετήθηκαν και αναπτύχθηκαν υβριδικά σχήματα στα οποία συμμετέχει η προτεινόμενη μέθοδος, με σκοπό την επίλυση προβλημάτων πολλαπλών στόχων ή την βελτίωση της συμπεριφοράς της προτεινόμενης μεθόδου (Κεφάλαια 6 και 7).

8.3 Μελλοντική Έρευνα

Διάφορες κατευθύνσεις προσφέρονται για περαιτέρω έρευνα με βάση τα ευρήματα και την μελέτη που έγινε στην παρούσα διατριβή. Αρχικά, η χαρτογράφηση

του χώρου των Νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση δίνει τη δυνατότητα για δύο διαφορετικές ερευνητικές κατευθύνσεις:

- α) Σε συνδυασμό με τα συμπεράσματα που προκύπτουν από τη μελέτη πάνω στις εφαρμογές των Νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση, μπορούν να δοκιμασθούν αλγόριθμοι που ταιριάζουν σε συγκεκριμένα είδη προβλημάτων, να συγκριθεί η απόδοσή τους και να διερευνηθούν πιθανά υβριδικά σχήματα. Αυτή η κατεύθυνση προσφέρει πολλαπλά θέματα έρευνας ανά χώρο εφαρμογής.
- β) Βάσει και του πλαισίου εργασίας που προτείνεται στην παρούσα διατριβή, χρήσιμη θα ήταν η ανάπτυξη ενός εργαλείου, βασισμένου στη λογική του άλλων υφιστάμενων συλλογών νοημόνων αλγορίθμων (WEKA, RapidMiner, HeuristicLab κ.ά.), που θα περιέχει τους Αλγόριθμους που είναι Εμπνευσμένοι από τη Φύση και θα δίνει τη δυνατότητα στον χρήστη να τους χρησιμοποιήσει στην επίλυση προβλημάτων.

Εξίσου ενδιαφέρουσα θα ήταν και η ανάλυση και ανάπτυξη μεθόδων μέτρησης της συμπεριφοράς (σύγκλιση (convergence), εξερεύνηση (exploration), εκμετάλλευση (exploitation), υπολογιστικός χρόνος) των Νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση με σκοπό να αναπτυχθούν μέθοδοι που θα δίνουν τη δυνατότητα στους αλγορίθμους να αυτορυθμίζονται.

Επιπρόσθετα, σημαντικά βήματα μπορούν να γίνουν στην εφαρμογή των νοημόνων Αλγορίθμων που είναι Εμπνευσμένοι από τη Φύση σε περιοχές προβλημάτων που δεν αναλύονται στην παρούσα διατριβή, όπως τα προβλήματα κατηγοριοποίησης και συσταδοποίησης ή προβλήματα πρόβλεψης.

Περαιτέρω έρευνα μπορεί να γίνει και επί της προτεινόμενης μεθόδου. Για την αντιμετώπιση του εκτεταμένου Υπολογιστικού Χρόνου που απαιτεί η Βελτιστοποίηση Εμπνευσμένη από τον Ηχοεντοπισμό (BEH), χρήσιμη θα ήταν η συμβολή του Παράλληλου Υπολογισμού (Parallel Computing), όπου θα εκτελούνταν παράλληλα χρονοβόρες διεργασίες, όπως η Σάρωση Πλήρους Κύκλου.

Επιπλέον, διαφορετικά υβριδικά σχήματα μπορούν να προταθούν με σκοπό την βελτίωση της συμπεριφοράς της BEH. Ειδικότερα, υπάρχει η δυνατότητα να μελετηθούν οι αδυναμίες που παρουσιάζει η BEH όταν καλείται να επιλύσει προβλήματα ενός συγκεκριμένου πεδίου εφαρμογής και να προταθούν υβριδικά σχήματα στα οποία θα καλύπτονται αυτές οι αδυναμίες από άλλες μεθόδους Υπολογιστικής Νοημοσύνης.

Στο κομμάτι της επιτυχούς αυτορύθμισης της προτεινόμενης μεθόδου, μπορούν να μελετηθούν προβλήματα στα οποία οι τιμές που μπορεί να λάβει η αντικειμενική συνάρτηση απαιτούν μετασχηματισμό των μηχανισμών της BEH και να προταθούν βελτιωμένες εκδοχές της εν λόγω μεθόδου. Στις βελτιώσεις αυτές δύναται να βελτιστοποιηθούν διάφορες τεχνικές λεπτομέρειες (technicalities) της προτεινόμενης μεθόδου. Μάλιστα, μια ειδική περίπτωση είναι η Απλουστευμένη Βελτιστοποίηση

Εμπνευσμένη από τον Ηχοεντοπισμό (Simplified SIO), η οποία θα απλοποιούσε τη διαδικασία, ώστε να μην απαιτείται μετασχηματισμός των παραμέτρων της μεθόδου οι οποίες ακολουθούν πιστά το φυσικό ανάλογο.

Τέλος, υπό μελέτη βρίσκεται η επιτυχής διακριτοποίηση της μεθόδου που προτείνεται στην παρούσα διατριβή, με σκοπό να επιλύει προβλήματα με μεταβλητές αόφασης που ορίζονται σε διακριτό χώρο λύσεων, όπως τα προβλήματα δρομολόγησης, χρονοδρομολόγησης κ.ά.

9. Βιβλιογραφία

- A. A. A. Esmin and G. Lambert-Torres, 2006. Loss Power Minimization Using Particle Swarm Optimization. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. The 2006 IEEE International Joint Conference on Neural Network Proceedings. pp.1988–1992.
- A. Bhattacharya and P. K. Chattopadhyay, 2010. Biogeography-Based Optimization for Different Economic Load Dispatch Problems. *IEEE Transactions on Power Systems*, 25(2), pp.1064–1077.
- A. Sharma, 2010. A new optimizing algorithm using reincarnation concept. In: *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*. 2010 11th International Symposium on Computational Intelligence and Informatics (CINTI). pp.281–288.
- Abbass, H.A., 2001. MBO: Marriage in honey bees optimization-A haplometrosis polygynous swarming approach. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546)*. IEEE. pp.207–214.
- Abdechiri, M., Meybodi, M.R. and Bahrami, H., 2013. Gases Brownian Motion Optimization: an Algorithm for Optimization (GBMO). *Applied Soft Computing*, 13(5), pp.2932–2946.
- Abedinia, O., Amjady, N. and Ghasemi, A., 2016. A new metaheuristic algorithm based on shark smell optimization. *Complexity*, 21(5), pp.97–116.
- Abedinpourshotorban, H., Mariyam Shamsuddin, S., Beheshti, Z. and Jawawi, D.N.A., 2016. Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm and Evolutionary Computation*, 26, pp.8–22.
- Ackley, D., 1987. *A Connectionist Machine for Genetic Hillclimbing*. The Springer International Series in Engineering and Computer Science. [online] Springer US. Available at: <<https://www.springer.com/gp/book/9780898382365>> [Accessed 7 Nov. 2019].
- Aghay Kaboli, S.Hr., Selvaraj, J. and Rahim, N.A., 2017. Rain-fall optimization algorithm: A population based algorithm for solving constrained optimization problems. *Journal of Computational Science*, 19, pp.31–42.
- Ahmadi, F., Salehi, H. and Karimi, K., 2012. Eurygaster Algorithm: A New Approach to Optimization. *International Journal of Computer Applications*, 57(2), pp.9–13.
- Ahrari, A. and Atai, A.A., 2010. Grenade Explosion Method—A novel tool for optimization of multimodal functions. *Optimisation Methods & Applications in Decision-Making Processes*, 10(4), pp.1132–1140.
- Akbari, R., Mohammadi, A. and Ziarati, K., 2010. A novel bee swarm optimization algorithm for numerical function optimization. *Communications in Nonlinear Science and Numerical Simulation*, 15(10), pp.3142–3155.
- Alatas, B., 2017. Sports inspired computational intelligence algorithms for global optimization. *Artificial Intelligence Review*. [online] Available at: <<https://doi.org/10.1007/s10462-017-9587-x>>.
- Ali, J., Saeed, M., Chaudhry, N.A., Luqman, M. and Tabassum, M.F., 2015. Artificial showering algorithm: a new meta-heuristic for unconstrained optimization. *Sci. Int*, 27(6), pp.4939–4942.

Aljarah, I., Faris, H. and Mirjalili, S., 2018. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22(1), pp.1–15.

Almonacid, B. and Soto, R., 2019. Andean Condor Algorithm for cell formation problems. *Natural Computing*, 18(2), pp.351–381.

Al-Obaidi, A.T.S., Abdullah, H.S. and Ahmed, zied O., 2018. Meerkat Clan Algorithm: A New Swarm Intelligence Algorithm. *Indonesian Journal of Electrical Engineering and Computer Science*, 10(1), pp.354–360.

Al-Roomi, A.R., 2015. *Power Flow Test Systems Repository*. [online] Electric Power Systems Analysis & Nature-Inspired Optimization Algorithms. Available at: <<https://al-roomi.org/power-flow>>.

Alsattar, H.A., Zaidan, A.A. and Zaidan, B.B., 2019. Novel meta-heuristic bald eagle search optimisation algorithm. *Artificial Intelligence Review*. [online] Available at: <<https://doi.org/10.1007/s10462-019-09732-5>>.

An, J., Kang, Q., Wang, L. and Wu, Q., 2013. Mussels Wandering Optimization: An Ecologically Inspired Algorithm for Global Optimization. *Cognitive Computation*, 5(2), pp.188–199.

Anagnostopoulos, K.P. and Koulinas, G.K., 2010. A simulated annealing hyperheuristic for construction resource levelling. *Construction Management and Economics*, 28(2), pp.163–175.

Anandaraman, C., Madurai Sankar, A.V. and Natarajan, R., 2012. A New Evolutionary Algorithm Based on Bacterial Evolution and Its Application for Scheduling A Flexible Manufacturing System. *Jurnal Teknik Industri; Vol 14, No 1 (2012): JUNE 2012DO* - 10.9744/jti.14.1.1-12. [online] Available at: <<http://jurnalindustri.petra.ac.id/index.php/ind/article/view/18401>>.

Apostolopoulos, T. and Vlachos, A., 2011. Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem. *International Journal of Combinatorics*, [online] 2011. Available at: <<https://www.hindawi.com/journals/ijcom/2011/523806/cta/>> [Accessed 15 Aug. 2019].

Arora, J.S., 2016. *Introduction to Optimum Design - 4th Edition*. 4th ed. [online] Elsevier. Available at: <<https://www.elsevier.com/books/introduction-to-optimum-design/arora/978-0-12-800806-5>> [Accessed 7 Aug. 2019].

Arora, S. and Singh, S., 2019. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing*, 23(3), pp.715–734.

Asil Gharebaghi, S. and Ardalan Asl, M., 2017. New meta-heuristic optimization algorithm using neuronal communication. *International Journal of Optimization in Civil Engineering*, 7(3), pp.413–431.

Askarzadeh, A., 2014. Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Communications in Nonlinear Science and Numerical Simulation*, 19(4), pp.1213–1228.

Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, 169, pp.1–12.

Atashpaz-Gargari, E. and Lucas, C., 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: *2007 IEEE congress on evolutionary computation*. IEEE. pp.4661–4667.

Azizyan, G., Miarnaemi, F., Rashki, M. and Shabakhty, N., 2019. Flying Squirrel Optimizer (FSO): A novel SI-based optimization algorithm for engineering problems. *Iranian Journal of Optimization*. [online] Available at: <http://ijo.iaurasht.ac.ir/article_663726.html>.

B. Xue, M. Zhang and W. N. Browne, 2013. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Transactions on Cybernetics*, 43(6), pp.1656–1671.

Bachmann, P., 1894. *Die analytische zahlentheorie*. Teubner.

Back, T., Fogel, D.B. and Michalewicz, Z. eds., 1997. *Handbook of Evolutionary Computation*. 1st ed. Bristol, UK, UK: IOP Publishing Ltd.

Bandelloni, M., Tucci, M. and Rinaldi, R., 1994. Optimal resource leveling using non-serial dyanamic programming. *Project Management and Scheduling*, 78(2), pp.162–177.

Bansal, J.C., Sharma, H., Jadon, S.S. and Clerc, M., 2014. Spider Monkey Optimization algorithm for numerical optimization. *Memetic Computing*, 6(1), pp.31–47.

Beattie, P.D. and Bishop, J.M., 1998. Self-Localisation in the ‘Senario’ Autonomous Wheelchair. *Journal of Intelligent and Robotic Systems*, 22(3), pp.255–267.

Beiranvand, H. and Rokrok, E., 2015. General Relativity Search Algorithm: A Global Optimization Approach. *International Journal of Computational Intelligence and Applications*, 14(03), p.1550017.

Belegundu, A.D. and Arora, J.S., 1985. A study of mathematical programming methods for structural optimization. Part I: Theory. *International Journal for Numerical Methods in Engineering*, 21(9), pp.1583–1599.

Bendato, I., Cassettari, L., Giribone, P.G. and Fioribello, S., 2016. Attraction Force Optimization (AFO): A Deterministic Nature-Inspired Heuristic for Solving Optimization Problems in Stochastic Simulation. *Applied Mathematical Sciences*, 10(20), pp.989–1011.

Beni, G. and Wang, J., 1993. Swarm Intelligence in Cellular Robotic Systems. In: P. Dario, G. Sandini and P. Aebischer, eds. *Robots and Biological Systems: Towards a New Bionics?* Springer Berlin Heidelberg. pp.703–712.

Birbil, Ş.İ. and Fang, S.-C., 2003. An electromagnetism-like mechanism for global optimization. *Journal of global optimization*, 25(3), pp.263–282.

Bishop, J.M. and Torr, P., 1992. The Stochastic Search Network. In: R. Linggard, D.J. Myers and C. Nightingale, eds. *Neural Networks for Vision, Speech and Natural Language*. [online] Dordrecht: Springer Netherlands. pp.370–387. Available at: <https://doi.org/10.1007/978-94-011-2360-0_24>.

Biswas, A., Mishra, K.K., Tiwari, S. and Misra, A.K., 2013. Physics-inspired optimization algorithms: a survey. *Journal of Optimization*, 2013.

Biyanto, T.R., Matradji, Irawan, S., Febrianto, H.Y., Afdanny, N., Rahman, A.H., Gunawan, K.S., Pratama, J.A.D. and Bethiana, T.N., 2017. Killer Whale Algorithm: An Algorithm

Inspired by the Life of Killer Whale. *4th Information Systems International Conference 2017, ISICO 2017, 6-8 November 2017, Bali, Indonesia*, 124, pp.151–157.

Blum, C. and Li, X., 2008. Swarm Intelligence in Optimization. In: C. Blum and D. Merkle, eds. *Swarm Intelligence: Introduction and Applications*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.43–85. Available at: <https://doi.org/10.1007/978-3-540-74089-6_2>.

Boctor, F.F., 1996. Resource-constrained project scheduling by simulated annealing. *International Journal of Production Research*, 34(8), pp.2335–2351.

Bouarara, H.A., Hamou, R.M. and Abdelmalek, A., 2018. Enhanced Artificial Social Cockroaches (EASC) for Modern Information Retrieval. *Information Retrieval and Management: Concepts, Methodologies, Tools, and Applications*, pp.928–960.

Boulas, K., Dounias, G. and Papadopoulos, C., 2017. Approximating Throughput of Small Production Lines Using Genetic Programming. In: E. Grigoroudis and M. Doumpos, eds. *Operational Research in Business and Economics*. Springer International Publishing. pp.185–204.

Boulas, K., Tzanetos, A. and Dounias, G., 2018. Acquisition of approximate throughput formulas for serial production lines with parallel machines using intelligent techniques. In: *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. ACM. p.18.

Boyd, S. and Vandenberghe, L., 2004. *Convex Optimization*. [online] Cambridge: Cambridge University Press. Available at: <<https://www.cambridge.org/core/books/convex-optimization/17D2FAA54F641A2F62C7CCD01DFA97C4>>.

Brabazon, A., Cui, W. and O'Neill, M., 2016. The raven roosting optimisation algorithm. *Soft Computing*, 20(2), pp.525–545.

Bukin, A.D., 1997. *New minimization strategy for nonsmooth functions*.

Can, U. and Alatas, B., 2015. Physics based metaheuristic algorithms for global optimization. *American Journal of Information Science and Computer Engineering*, 1(3), pp.94–106.

Carnevali, P., Coletti, L. and Patarnello, S., 1987. Image processing by simulated annealing. In: M.A. Fischler and O. Firschein, eds. *Readings in Computer Vision*. [online] San Francisco (CA): Morgan Kaufmann. pp.551–561. Available at: <<http://www.sciencedirect.com/science/article/pii/B9780080515816500556>>.

de Castro, L.N. and von Zuben, F.J., 2000. The clonal selection algorithm with engineering applications. *GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, pp.36–37.

Chakraborty, S., Senjyu, T., Yona, A., Saber, A.Y. and Funabashi, T., 2011. Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation. *IET Generation, Transmission & Distribution*, 5(10), pp.1042–1052(10).

Chen, C., Tsai, Y., Liu, I., Lai, C., Yeh, Y., Kuo, S. and Chou, Y., 2015. A Novel Metaheuristic: Jaguar Algorithm with Learning Behavior. In: *2015 IEEE International Conference on Systems, Man, and Cybernetics*. 2015 IEEE International Conference on Systems, Man, and Cybernetics. pp.1595–1600.

- Chen, H., Zhu, Y., Hu, K. and He, X., 2010. Hierarchical Swarm Model: A New Approach to Optimization. *Discrete Dynamics in Nature and Society*. [online] Available at: <<https://www.hindawi.com/journals/ddns/2010/379649/abs/>> [Accessed 10 Jul. 2019].
- Chen, S., 2009a. Locust Swarms - A new multi-optima search technique. In: *2009 IEEE Congress on Evolutionary Computation*. 2009 IEEE Congress on Evolutionary Computation. pp.1745–1752.
- Chen, T., 2009b. A Simulative Bionic Intelligent Optimization Algorithm: Artificial Searching Swarm Algorithm and Its Performance Analysis. In: *2009 International Joint Conference on Computational Sciences and Optimization*. 2009 International Joint Conference on Computational Sciences and Optimization. pp.864–866.
- Chen, Z., 1999. *Computational intelligence for decision support*. CRC Press.
- Cheng, L., Han, L., Zeng, X., Bian, Y. and Yan, H., 2015. Adaptive Cockroach Colony Optimization for Rod-Like Robot Navigation. *Journal of Bionic Engineering*, 12(2), pp.324–337.
- Cheng, L., Zhang, Q., Tao, F., Ni, K. and Cheng, Y., 2016. A novel search algorithm based on waterweeds reproduction principle for job shop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 84(1), pp.405–424.
- Cheng, M.-Y. and Prayogo, D., 2014. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Computers & Structures*, 139, pp.98–112.
- Cheraghalipour, A., Hajiaghaei-Keshteli, M. and Paydar, M.M., 2018. Tree Growth Algorithm (TGA): A novel approach for solving optimization problems. *Engineering Applications of Artificial Intelligence*, 72, pp.393–414.
- Christian, B. and Griffiths, T., 2016. *Algorithms to Live by: The Computer Science of Human Decisions*. HarperCollins Publishers.
- Chu, S.-C., Tsai, P. and Pan, J.-S., 2006. Cat Swarm Optimization. In: Q. Yang and G. Webb, eds. *PRICAI 2006: Trends in Artificial Intelligence*, Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp.854–858.
- Coello Coello, C.A., 2000a. Constraint-handling using an evolutionary multiobjective optimization technique. *Civil Engineering and Environmental Systems*, 17(4), pp.319–346.
- Coello Coello, C.A., 2000b. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 41(2), pp.113–127.
- Colak, M.E. and Varol, A., 2015. A novel intelligent optimization algorithm inspired from circular water waves. *Elektronika ir Elektrotechnika*, 21(5), pp.3–6.
- Comellas, F. and Martinez-Navarro, J., 2009. Bumblebees: a multiagent combinatorial optimization algorithm inspired by social insect behaviour. In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. ACM. pp.811–814.
- Cook, S.A., 1971. The Complexity of Theorem-proving Procedures. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71. [online] 3rd annual ACM symposium on Theory of computing. New York, NY, USA: ACM. pp.151–158. Available at: <<http://doi.acm.org/10.1145/800157.805047>>.

Cortés, P., García, J.M., Muñuzuri, J. and Onieva, L., 2008. Viral systems: A new bio-inspired optimisation approach. *Part Special Issue: Bio-inspired Methods in Combinatorial Optimization*, 35(9), pp.2840–2860.

Cortés-Toro, M.E., Crawford, B., Gómez-Pulido, A.J., Soto, R. and Lanza-Gutiérrez, M.J., 2018. A New Metaheuristic Inspired by the Vapour-Liquid Equilibrium for Continuous Optimization. *Applied Sciences*, 8(11).

Črepinšek, M., Liu, S.-H. and Mernik, M., 2013. Exploration and Exploitation in Evolutionary Algorithms: A Survey. *ACM Comput. Surv.*, 45(3), pp.35:1–35:33.

Cuevas, E., Cienfuegos, M., Zaldívar, D. and Pérez-Cisneros, M., 2013. A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, 40(16), pp.6374–6384.

Cuevas, E., Echavarría, A. and Ramírez-Ortegón, M.A., 2014. An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied Intelligence*, 40(2), pp.256–272.

Cuevas, E., Fausto, F. and González, A., 2020. A Swarm Algorithm Inspired by the Collective Animal Behavior. In: E. Cuevas, F. Fausto and A. González, eds. *New Advancements in Swarm Algorithms: Operators and Applications*. [online] Cham: Springer International Publishing. pp.161–188. Available at: <https://doi.org/10.1007/978-3-030-16339-6_6>.

Cunha Maria da Conceição and Sousa Joaquim, 1999. Water Distribution Network Design Optimization: Simulated Annealing Approach. *Journal of Water Resources Planning and Management*, 125(4), pp.215–221.

D. H. Wolpert and W. G. Macready, 1997. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), pp.67–82.

D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck and B. Baesens, 2007. Classification With Ant Colony Optimization. *IEEE Transactions on Evolutionary Computation*, 11(5), pp.651–665.

Das, S., Biswas, A., Dasgupta, S. and Abraham, A., 2009. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In: *Foundations of Computational Intelligence Volume 3*. Springer. pp.23–55.

Das, S. and Suganthan, P.N., 2010. *Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems*. pp.341–359.

Deb, K., 1991. Optimal design of a welded beam via genetic algorithms. *AIAA Journal*, 29(11), pp.2013–2015.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2), pp.311–338.

Deb, S., Tian, Z., Fong, S., Tang, R., Wong, R. and Dey, N., 2018. Solving permutation flow-shop scheduling problem by rhinoceros search algorithm. *Soft Computing*, 22(18), pp.6025–6034.

- Dhiman, G. and Kaur, A., 2019. STOA: A bio-inspired based optimization algorithm for industrial engineering problems. *Engineering Applications of Artificial Intelligence*, 82, pp.148–174.
- Dhiman, G. and Kumar, V., 2017. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Advances in Engineering Software*, 114, pp.48–70.
- Dhiman, G. and Kumar, V., 2018. Emperor penguin optimizer: A bio-inspired algorithm for engineering problems. *Knowledge-Based Systems*, 159, pp.20–50.
- Dhiman, G. and Kumar, V., 2019. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Systems*, 165, pp.169–196.
- Diamantidis, A.C., Papadopoulos, C.T. and Heavey, C., 2007. Approximate analysis of serial flow lines with multiple parallel-machine stations. *IIE Transactions*, 39(4), pp.361–375.
- Doğan, B. and Ölmez, T., 2015. A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information Sciences*, 293, pp.125–145.
- Dorigo, M., 1992. Optimization, learning and natural algorithms. *PhD Thesis, Politecnico di Milano*.
- Dorigo, M. and Stützle, T., 1999. ACO algorithms for the traveling salesman problem. In: K. Miettinen, ed. *Evolutionary Algorithms in Engineering and Computer Science (K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, eds.) Wiley*. John Wiley & Sons, Inc. pp.163–183.
- Duman, E., Uysal, M. and Alkaya, A.F., 2011. Migrating Birds Optimization: A New Metaheuristic Approach and Its Application to the Quadratic Assignment Problem. In: C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A.I. Esparcia-Alcázar, J.J. Merelo, F. Neri, M. Preuss, H. Richter, J. Togelius and G.N. Yannakakis, eds. *Applications of Evolutionary Computation*. Springer Berlin Heidelberg. pp.254–263.
- E. Pashaei, M. Ozen and N. Aydin, 2015. An application of black hole algorithm and decision tree for medical problem. In: *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*. 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE). pp.1–6.
- Eberhart, R. and Kennedy, J., 1995. A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Ieee. pp.39–43.
- Ebrahimi, A. and Khamehchi, E., 2016. Sperm whale algorithm: An effective metaheuristic algorithm for production optimization problems. *Journal of Natural Gas Science and Engineering*, 29, pp.211–222.
- Eesa, A.S., Brifcani, A.M.A. and Orman, Z., 2013. Cuttlefish Algorithm – A Novel Bio-Inspired Optimization Algorithm. *International Journal of Scientific & Engineering Research*, 4(9), pp.1978–1986.
- Eiben, A.E. and Smith, J.E., 2003. *Introduction to Evolutionary Computing*. Natural Computing Series. [online] Berlin Heidelberg: Springer-Verlag. Available at: <<https://www.springer.com/gp/book/9783642072857>> [Accessed 22 Jul. 2019].

El-Dosuky, M., El-Bassiouny, A., Hamza, T. and Rashad, M., 2012. New Hoopoe Heuristic Optimization. *CoRR*, [online] abs/1211.6410. Available at: <<http://arxiv.org/abs/1211.6410>>.

Elwany, M.H., Korish, I.E., Aly Barakat, M. and Hafez, S.M., 1998. Resource smoothening in repetitive projects. *Selected Papers from the 22nd ICC and IE Conference*, 35(3), pp.415–418.

Erol, O.K. and Eksin, I., 2006. A new optimization method: big bang–big crunch. *Advances in Engineering Software*, 37(2), pp.106–111.

Eskandar, H., Sadollah, A., Bahreininejad, A. and Hamdi, M., 2012. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110, pp.151–166.

Eusuff, M., Lansey, K. and Pasha, F., 2006. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization*, 38(2), pp.129–154.

F. Grimaccia, M. Mussetta and R. E. Zich, 2007. Genetical Swarm Optimization: Self-Adaptive Hybrid Evolutionary Algorithm for Electromagnetics. *IEEE Transactions on Antennas and Propagation*, 55(3), pp.781–785.

F. Zhuang and F. D. Galiana, 1990. Unit commitment by simulated annealing. *IEEE Transactions on Power Systems*, 5(1), pp.311–318.

Fadafen, M.K., Mehrshad, N., Zahiri, S.H. and Razavi, S.M., 2019. A New Algorithm for Optimization Based On Ohm's Law. *Control Engineering Letters*, 2(1), pp.16–22.

Fard, A.F. and Hajiaghaei-Keshteli, M., 2016. Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating. *International Conference on Industrial Engineering, IEEE*, (2016 e). pp.33–34.

Fausto, F., Cuevas, E., Valdivia, A. and González, A., 2017. A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems*, 160, pp.39–55.

Feng, X., Liu, Y., Yu, H. and Luo, F., 2019. Physarum-energy optimization algorithm. *Soft Computing*, 23(3), pp.871–888.

Feng, X., Ma, M. and Yu, H., 2016. Crystal Energy Optimization Algorithm. *Computational Intelligence*, 32(2), pp.284–322.

Fidanova, S. and Durchova, M., 2006. Ant Algorithm for Grid Scheduling Problem. In: I. Lirkov, S. Margenov and J. Waśniewski, eds. *Large-Scale Scientific Computing*. Springer Berlin Heidelberg. pp.405–412.

Filho, C.J.A.B., de Lima Neto, F.B., Lins, A.J.C.C., Nascimento, A.I.S. and Lima, M.P., 2009. Fish School Search. In: R. Chiong, ed. *Nature-Inspired Algorithms for Optimisation*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.261–277. Available at: <https://doi.org/10.1007/978-3-642-00267-0_9>.

Fister, I., Strnad, D. and Yang, X.-S., 2015. Adaptation and hybridization in nature-inspired algorithms. In: *Adaptation and hybridization in computational intelligence*. Springer. pp.3–50.

Fister Jr, I., Mlakar, U., Brest, J. and Fister, I., 2016. A new population-based nature-inspired algorithm every month: is the current era coming to the end. In: *Proceedings of the 3rd*

Student Computer Science Research Conference. 3rd Student Computer Science Research Conference. University of Primorska Press.pp.33–37.

Fister Jr., I., Yang, X.-S., Fister, I., Brest, J. and Fister, D., 2013. A Brief Review of Nature-Inspired Algorithms for Optimization. *arXiv:1307.4186 [cs]*. [online] Available at: <<http://arxiv.org/abs/1307.4186>> [Accessed 7 Apr. 2019].

Flores, J.J., López, R. and Barrera, J., 2011. Gravitational Interactions Optimization. In: C.A.C. Coello, ed. *Learning and Intelligent Optimization*, Lecture Notes in Computer Science. Springer Berlin Heidelberg.pp.226–237.

Formato, R.A., 2007. Central force optimization. *Prog Electromagn Res*, 77, pp.425–491.

Fountas, C. and Vlachos, A., 2005. Ant Colonies Optimization (ACO) for the solution of the Vehicle Routing Problem (VRP). *Journal of Information and Optimization Sciences*, 26(1), pp.135–142.

Gaitanidis, A., Vassiliadis, V., Kyriklidis, C. and Dounias, G., 2016. Hybrid Evolutionary Algorithms in Resource Leveling Optimization: Application in a Large Real Construction Project of a 50000 DWT Ship. In: *Proceedings of the 9th Hellenic Conference on Artificial Intelligence*, SETN '16. [online] New York, NY, USA: ACM.pp.25:1–25:8. Available at: <<http://doi.acm.org/10.1145/2903220.2903227>>.

Gandomi, A.H. and Alavi, A.H., 2012. Krill herd: a new bio-inspired optimization algorithm. *Communications in nonlinear science and numerical simulation*, 17(12), pp.4831–4845.

Garcia, F.J.M. and Pérez, J.A.M., 2008. Jumping frogs optimization: a new swarm method for discrete optimization. *Documentos de Trabajo del DEIOC*, 3.

Garcia, M.A.P., Montiel, O., Castillo, O., Sepúlveda, R. and Melin, P., 2009. Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. *Applied Soft Computing*, 9(3), pp.1102–1110.

Ghaemi, M. and Feizi-Derakhshi, M.-R., 2014. Forest Optimization Algorithm. *Expert Systems with Applications*, 41(15), pp.6676–6687.

Ghasemi, M., Ghavidel, S., Aghaei, J., Akbari, E. and Li, L., 2018. CFA optimizer: A new and powerful algorithm inspired by Franklin's and Coulomb's laws theory for solving the economic load dispatch problems. *International Transactions on Electrical Energy Systems*, 28(5), p.e2536.

Gheraibia, Y. and Moussaoui, A., 2013. Penguins Search Optimization Algorithm (PeSOA). In: M. Ali, T. Bosse, K.V. Hindriks, M. Hoogendoorn, C.M. Jonker and J. Treur, eds. *Recent Trends in Applied Artificial Intelligence*, Lecture Notes in Computer Science. Springer Berlin Heidelberg.pp.222–231.

Ghojogh, B. and Sharifian, S., 2018. Pontogammarus Maeoticus Swarm Optimization: A Metaheuristic Optimization Algorithm. *CoRR*, [online] abs/1807.01844. Available at: <<http://arxiv.org/abs/1807.01844>>.

Ghojogh, B., Sharifian, S. and Mohammadzade, H., 2018. Tree-Based Optimization: A Meta-Algorithm for Metaheuristic Optimization. *CoRR*, [online] abs/1809.09284. Available at: <<http://arxiv.org/abs/1809.09284>>.

Giannakouris, G., Vassiliadis, V. and Dounias, G., 2010. Experimental study on a hybrid nature-inspired algorithm for financial portfolio optimization. In: *Hellenic Conference on Artificial Intelligence*. Springer. pp.101–111.

Gittins, J.C., 1979. Bandit Processes and Dynamic Allocation Indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2), pp.148–164.

Gittins, J.C. and Jones, D.M., 1979. A dynamic allocation index for the discounted multiarmed bandit problem. *Biometrika*, 66(3), pp.561–565.

Gutjahr, W.J. and Rauner, M.S., 2007. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Logistics of Health Care Management*, 34(3), pp.642–666.

H. Mo and L. Xu, 2013. Magnetotactic bacteria optimization algorithm for multimodal optimization. In: *2013 IEEE Symposium on Swarm Intelligence (SIS)*. 2013 IEEE Symposium on Swarm Intelligence (SIS). pp.240–247.

H. R. E. H. Bouchekara, 2019. Electrostatic discharge algorithm: a novel nature-inspired optimisation algorithm and its application to worst-case tolerance analysis of an EMC filter. *IET Science, Measurement & Technology*, 13(4), pp.491–499.

H. Wu, J. Geng, R. Jin, J. Qiu, W. Liu, J. Chen and S. Liu, 2009. An Improved Comprehensive Learning Particle Swarm Optimization and Its Application to the Semiautomatic Design of Antennas. *IEEE Transactions on Antennas and Propagation*, 57(10), pp.3018–3028.

Haddad, O.B., Afshar, A. and Mariño, M.A., 2006. Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization. *Water Resources Management*, 20(5), pp.661–680.

Hajiaghaei-Keshteli, M. and Aminnayeri, M., 2013. Keshtel Algorithm (KA): a new optimization algorithm inspired by Keshtels' feeding. In: *Proceeding in IEEE conference on industrial engineering and management systems*. IEEE conference on industrial engineering and management systems. Rabat, Morocco: IEEE. pp.2249–2253.

Haktanirlar Ulutas, B. and Kulturel-Konak, S., 2011. A review of clonal selection algorithm and its applications. *Artificial Intelligence Review*, 36(2), pp.117–138.

Haldar, V. and Chakraborty, N., 2017. A novel evolutionary technique based on electrolocation principle of elephant nose fish and shark: fish electrolocation optimization. *Soft Computing*, 21(14), pp.3827–3848.

Harifi, S., Khalilian, M., Mohammadzadeh, J. and Ebrahimnejad, S., 2019. Emperor Penguins Colony: a new metaheuristic algorithm for optimization. *Evolutionary Intelligence*, 12(2), pp.211–226.

Hartmanis, J. and Stearns, R.E., 1965. On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society*, 117, pp.285–306.

Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W. and Mirjalili, S., 2019. Henry gas solubility optimization: A novel physics-based algorithm. *Future Generation Computer Systems*, 101, pp.646–667.

Hatamlou, A., 2013. Black hole: A new heuristic optimization approach for data clustering. *Information sciences*, 222, pp.175–184.

Hatamlou, A., 2014. Heart: a novel optimization algorithm for cluster analysis. *Progress in Artificial Intelligence*, 2(2), pp.167–173.

Hayes-Roth, B., 1995. An architecture for adaptive intelligent systems. *Artificial Intelligence*, 72(1), pp.329–365.

Hayyolalam, V. and Pourhaji Kazem, A.A., 2020. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 87, p.103249.

He, Q. and Wang, L., 2007. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 20(1), pp.89–99.

Hedayatzadeh, R., Salmassi, F.A., Keshtgari, M., Akbari, R. and Ziarati, K., 2010. Termite colony optimization: A novel approach for optimizing continuous problems. In: *2010 18th Iranian Conference on Electrical Engineering*. 2010 18th Iranian Conference on Electrical Engineering. pp.553–558.

Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97, pp.849–872.

Hellwig, M. and Beyer, H.-G., 2019. Benchmarking evolutionary algorithms for single objective real-valued constrained optimization – A critical review. *Swarm and Evolutionary Computation*, 44, pp.927–944.

Hernández, H. and Blum, C., 2011. Implementing a Model of Japanese Tree Frogs' Calling Behavior in Sensor Networks: A Study of Possible Improvements. In: *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation, GECCO '11*. [online] New York, NY, USA: ACM. pp.615–622. Available at: <<http://doi.acm.org/10.1145/2001858.2002057>>.

Hernández, H. and Blum, C., 2012. Distributed graph coloring: an approach based on the calling behavior of Japanese tree frogs. *Swarm Intelligence*, 6(2), pp.117–150.

Himmelblau, D.M., 1972. *Applied nonlinear programming*. McGraw-Hill Companies.

Hodashinsky, I. and Samsonov, S., 2017. Design of fuzzy rule based classifier using the monkey algorithm. *Business Informatics*, (1 (39)), pp.61–67.

Holland, J.H., 1975. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Oxford, England: U Michigan Press.

Holland, J.H., 1992a. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Holland, J.H., 1992b. Genetic algorithms. *Scientific american*, 267(1), pp.66–73.

Hosseini, E., 2017. Laying chicken algorithm: A new meta-heuristic approach to solve continuous programming problems. *Journal of Applied & Computational Mathematics*, 6, pp.10–4172.

Hsiao, Y.-T., Chuang, C.-L., Jiang, J.-A. and Chien, C.-C., 2005. A novel optimization algorithm: space gravitational optimization. In: *2005 IEEE international conference on systems, man and cybernetics*. 2005 IEEE International Conference on Systems, Man and Cybernetics. IEEE.pp.2323–2328.

Huang, F., Wang, L. and He, Q., 2007. An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, 186(1), pp.340–356.

Huang, G., 2016. Artificial infectious disease optimization: A SEIQR epidemic dynamic model-based function optimization algorithm. *Swarm and Evolutionary Computation*, 27, pp.31–67.

Huang, J., Wang, X. and Chen, R., 2010. Genetic Algorithms for Optimization of Resource Allocation in Large Scale Construction Project Management. *Journal of Computers*, 5(12), p.1917.

Hudaib, A.A. and Fakhouri, H.N., 2017. Supernova Optimizer: A Novel Natural Inspired Meta-Heuristic. *Modern Applied Science*, 12(1), p.p32.

Hurley, S. and Whitaker, R.M., 2002. An Agent Based Approach to Site Selection for Wireless Networks. In: *Proceedings of the 2002 ACM Symposium on Applied Computing, SAC '02*. [online] New York, NY, USA: ACM.pp.574–577. Available at: <<http://doi.acm.org/10.1145/508791.508902>>.

Hussain, K., Salleh, M.N.M., Cheng, S. and Shi, Y., 2019. On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Computing and Applications*, 31(11), pp.7665–7683.

Husseinzadeh Kashan, A., 2015. A new metaheuristic for optimization: Optics inspired optimization (OIO). *Computers & Operations Research*, 55, pp.99–125.

Hyunchul Kim and Byungchul Ahn, 2001. A new evolutionary algorithm based on sheep flocks heredity model. In: *2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233)*. 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (IEEE Cat. No.01CH37233). pp.514–517 vol.2.

Ibrahim, M.K. and Ali, R.S., 2016. Novel Optimization Algorithm Inspired by Camel Traveling Behavior. *Iraqi Journal for Electrical And Electronic Engineering*, 12(2), pp.167–177.

Igel, C. and Toussaint, M., 2005. A No-Free-Lunch theorem for non-uniform distributions of target functions. *Journal of Mathematical Modelling and Algorithms*, 3(4), pp.313–322.

J. M. Abdullah and T. Ahmed, 2019. Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process. *IEEE Access*, 7, pp.43473–43486.

J. M. Bishop, 1989. Stochastic searching networks. In: *1989 First IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313)*. 1989 First IEE International Conference on Artificial Neural Networks, (Conf. Publ. No. 313). pp.329–331.

J. Nenortaitė and R. Simutis, 2005. Adapting particle swarm optimization to stock markets. In: *5th International Conference on Intelligent Systems Design and Applications (ISDA '05)*. 5th International Conference on Intelligent Systems Design and Applications (ISDA '05). pp.520–525.

- J. Pierezan and L. Dos Santos Coelho, 2018. Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018 IEEE Congress on Evolutionary Computation (CEC). pp.1–8.
- Jaddi, N.S., Alvankarian, J. and Abdullah, S., 2017. Kidney-inspired algorithm for optimization problems. *Communications in Nonlinear Science and Numerical Simulation*, 42, pp.358–369.
- Jaderyan, M. and Khotanlou, H., 2016. Virulence Optimization Algorithm. *Applied Soft Computing*, 43, pp.596–618.
- Jahani, E. and Chizari, M., 2018. Tackling global optimization problems with a novel algorithm – Mouth Brooding Fish algorithm. *Applied Soft Computing*, 62, pp.987–1002.
- Jain, M., Maurya, S., Rani, A. and Singh, V., 2018. Owl search algorithm: A novel nature-inspired heuristic paradigm for global optimization. *Journal of Intelligent & Fuzzy Systems*, 34(3), pp.1573–1582.
- Jain, M., Singh, V. and Rani, A., 2019. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm and Evolutionary Computation*, 44, pp.148–175.
- Jamil, M. and Yang, X.-S., 2013. A Literature Survey of Benchmark Functions For Global Optimization Problems. *arXiv e-prints*, p.arXiv:1308.4008.
- Javidy, B., Hatamlou, A. and Mirjalili, S., 2015. Ions motion algorithm for solving optimization problems. *Applied Soft Computing*, 32, pp.72–79.
- Jiang, Q., Wang, L., Hei, X., Fei, R., Yang, D., Zou, F., Li, H., Cao, Z. and Lin, Y., 2014. Optimal approximation of stable linear systems with a novel and efficient optimization algorithm. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2014 IEEE Congress on Evolutionary Computation (CEC). pp.840–844.
- Jiang, X. and Li, S., 2017. BAS: Beetle Antennae Search Algorithm for Optimization Problems. *CoRR*, [online] abs/1710.10724. Available at: <<http://arxiv.org/abs/1710.10724>>.
- K. Kiran, P. D. Shenoy, K. R. Venugopal and L. M. Patnaik, 2014. Fault tolerant BeeHive routing in mobile ad-hoc multi-radio network. In: *2014 IEEE REGION 10 SYMPOSIUM*. 2014 IEEE REGION 10 SYMPOSIUM. pp.116–120.
- Kallioras, N.Ath., Lagaros, N.D. and Avtzis, D.N., 2018. Pity beetle algorithm – A new metaheuristic inspired by the behavior of bark beetles. *Advances in Engineering Software*, 121, pp.147–166.
- Karaboga, D. and Basturk, B., 2007. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), pp.459–471.
- Karci, A. and Alatas, B., 2006. Thinking Capability of Saplings Growing Up Algorithm. In: E. Corchado, H. Yin, V. Botti and C. Fyfe, eds. *Intelligent Data Engineering and Automated Learning – IDEAL 2006*. Springer Berlin Heidelberg. pp.386–393.
- Kaveh, A. and Bakhshpoori, T., 2016. Water Evaporation Optimization: A novel physically inspired optimization algorithm. *Computers & Structures*, 167, pp.69–85.

Kaveh, A. and Dadras, A., 2017. A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in Engineering Software*, 110, pp.69–84.

Kaveh, A. and Farhoudi, N., 2013. A new optimization method: Dolphin echolocation. *Advances in Engineering Software*, 59, pp.53–70.

Kaveh, A. and Ilchi Ghazaan, M., 2017. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. *Acta Mechanica*, 228(1), pp.307–322.

Kaveh, A. and Khayatazad, M., 2012. A new meta-heuristic method: Ray Optimization. *Computers & Structures*, 112–113, pp.283–294.

Kaveh, A. and Kooshkebaghi, M., 2019. Artificial Coronary Circulation System; A new bio-inspired metaheuristic algorithm. *Scientia Iranica*. [online] Available at: <http://scientiairanica.sharif.edu/article_21366.html>.

Kaveh, A. and Mahdavi, V.R., 2014. Colliding bodies optimization: A novel meta-heuristic method. *Computers & Structures*, 139, pp.18–27.

Kaveh, A. and Mahjoubi, S., 2018. Lion pride optimization algorithm: A meta-heuristic method for global optimization problems. *Scientia Iranica*, 25(Special Issue Dedicated to Professor Goodarz Ahmadi), pp.3113–3132.

Kaveh, A. and Talatahari, S., 2010. A novel heuristic optimization method: charged system search. *Acta Mechanica*, 213(3), pp.267–289.

Kaveh, A. and Zolghadr, A., 2017. Cyclical parthenogenesis algorithm: a new meta-heuristic algorithm. *Asian Journal of Civil Engineering (Building and Housing)*, 18(5), pp.673–701.

Kazem, A., Sharifi, E., Hussain, F.K., Saberi, M. and Hussain, O.K., 2013. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing*, 13(2), pp.947–958.

Kazikova, A., Pluhacek, M., Senkerik, R. and Viktorin, A., 2019. Proposal of a New Swarm Optimization Method Inspired in Bison Behavior. In: R. Matoušek, ed. *Recent Advances in Soft Computing*. Springer International Publishing. pp.146–156.

Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization (PSO). In: *Proc. IEEE International Conference on Neural Networks, Perth, Australia*. pp.1942–1948.

Khan, A.T., Li, S., Stanimirovic, P.S. and Zhang, Y., 2018. Model-Free Optimization Using Eagle Perching Optimizer. *CoRR*, [online] abs/1807.02754. Available at: <<http://arxiv.org/abs/1807.02754>>.

Kiran, M.S., 2015. TSA: Tree-seed algorithm for continuous optimization. *Expert Systems with Applications*, 42(19), pp.6686–6698.

Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, 220(4598), pp.671–680.

Kılınc, N., Mahouti, P. and Güneş, F., 2013. Space Gravity Optimization Applied to the Feasible Design Target Space Required for a Wide-band Front-end Amplifier. In: *PIERS Proceedings*.

Klein, C.E., Mariani, V.C. and Coelho, L. dos S., 2018. Cheetah Based Optimization Algorithm: A Novel Swarm Intelligence Paradigm. In: *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*. [online] ESANN 2018: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: UCL Upcoming Conferences for Computer Science & Electronics, pp.685–690. Available at: <<http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-18.pdf>>.

Klein, C.E. and dos Santos Coelho, L., 2018. Meerkats-inspired Algorithm for Global Optimization Problems. In: *26th European Symposium on Artificial Neural Networks, ESANN 2018, Bruges, Belgium, April 25-27, 2018*. [online] ESANN 2018: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium. Available at: <<https://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2018-35.pdf>>.

Knuth, D.E., 1976. Big Omicron and big Omega and big Theta. *ACM SIGACT News*, 8(2), pp.18–24.

Kodratoff, Y. and Michalski, R.S. eds., 1990. *Machine Learning: An Artificial Intelligence Approach Volume III*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Koza, J.R., 1992. *Genetic programming: on the programming of computers by means of natural selection*. MIT press.

Krishnanand, K.N. and Ghose, D., 2006. Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multiagent and Grid Systems*, 2(3), pp.209–222.

Kühn, J., 2006. *Optimal Risk-Return Trade-Offs of Commercial Banks: and the Suitability of Profitability Measures for Loan Portfolios*. Lecture Notes in Economics and Mathematical Systems. [online] Berlin Heidelberg: Springer-Verlag. Available at: <<https://www.springer.com/gp/book/9783540348191>> [Accessed 7 Aug. 2019].

Kumar, M. and Kulkarni, A.J., 2019. Socio-inspired Optimization Metaheuristics: A Review. In: A.J. Kulkarni, P.K. Singh, S.C. Satapathy, A. Husseinzadeh Kashan and K. Tai, eds. *Socio-cultural Inspired Metaheuristics*. [online] Singapore: Springer Singapore, pp.241–265. Available at: <https://doi.org/10.1007/978-981-13-6569-0_12>.

Kuo, I.-H., Horng, S.-J., Kao, T.-W., Lin, T.-L., Lee, C.-L. and Pan, Y., 2009. An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization. *Expert Systems with Applications*, 36(3, Part 2), pp.6108–6117.

Kyriklidis, C. and Dounias, G., 2014. Application of Evolutionary Algorithms in Project Management. In: L. Iliadis, I. Maglogiannis and H. Papadopoulos, eds. *Artificial Intelligence Applications and Innovations*. Springer Berlin Heidelberg, pp.335–343.

Kyriklidis, C. and Dounias, G., 2016. Evolutionary computation for resource leveling optimization in project management. *Integrated Computer-Aided Engineering*, 23(2), pp.173–184.

Kyriklidis, C., Vassiliadis, V., Kirytopoulos, K. and Dounias, G., 2014. Hybrid nature-inspired intelligence for the resource leveling problem. *Operational Research*, 14(3), pp.387–407.

L. S. Coelho and V. C. Mariani, 2006. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power Systems*, 21(2), pp.989–996.

van Laarhoven, P.J.M., Aarts, E.H.L. and Lenstra, J.K., 1992. Job Shop Scheduling by Simulated Annealing. *Operations Research*, 40(1), pp.113–125.

Labbi, Y., Attous, D.B., Gabbar, H.A., Mahdad, B. and Zidan, A., 2016. A new rooted tree optimization algorithm for economic dispatch with valve-point effect. *International Journal of Electrical Power & Energy Systems*, 79, pp.298–311.

Lam, A.Y.S. and Li, V.O.K., 2010. Chemical-Reaction-Inspired Metaheuristic for Optimization. *IEEE Transactions on Evolutionary Computation*, 14(3), pp.381–399.

Lamy, J.-B., 2019. Artificial Feeding Birds (AFB): A New Metaheuristic Inspired by the Behavior of Pigeons. In: S.K. Shandilya, S. Shandilya and A.K. Nagar, eds. *Advances in Nature-Inspired Computing and Applications*. [online] Cham: Springer International Publishing. pp.43–60. Available at: <https://doi.org/10.1007/978-3-319-96451-5_3>.

Landau, E., 2000. *Handbuch der Lehre von der Verteilung der Primzahlen*. Рипол Классик.

Lee, K.S. and Geem, Z.W., 2005. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, 194(36), pp.3902–3933.

Lessing, L., Dumitrescu, I. and Stützle, T., 2004. A Comparison Between ACO Algorithms for the Set Covering Problem. In: M. Dorigo, M. Birattari, C. Blum, L.M. Gambardella, F. Mondada and T. Stützle, eds. *Ant Colony Optimization and Swarm Intelligence*. Springer Berlin Heidelberg. pp.1–12.

Leu, S.-S. and Hung, T.-H., 2002. An optimal construction resource leveling scheduling simulation model. *Canadian Journal of Civil Engineering*, 29(2), pp.267–275.

Leu, S.-S., Yang, C.-H. and Huang, J.-C., 2000. Resource leveling in construction by genetic algorithm-based optimization and its decision support system application. *Automation in Construction*, 10(1), pp.27–41.

Levin, L.A., 1973. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3), pp.115–116.

Li, K., Gao, X.-W., Zhou, H.-B. and Han, Y., 2015. Fault diagnosis for down-hole conditions of sucker rod pumping systems based on the FBH-SC method. *Petroleum Science*, 12(1), pp.135–147.

Li, L., Qian, Y., Du, K. and Yang, Y., 2016a. Analysis of approximately balanced production lines. *International Journal of Production Research*, 54(3), pp.647–664.

Li, M.D., Zhao, H., Weng, X.W. and Han, T., 2016b. A novel nature-inspired algorithm for optimization: Virus colony search. *Advances in Engineering Software*, 92, pp.65–88.

Li, X., Zhang, J. and Yin, M., 2014. Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Computing and Applications*, 24(7–8), pp.1867–1877.

- Li, X.L., Shao, Z.J. and Qian, J.X., 2002. An optimizing method based on autonomous animates: Fish-swarm algorithm. *System Engineering Theory and Practice*, 22(11), pp.32–38.
- Liao, C.-J., Chao-Tang Tseng and Luarn, P., 2007. A discrete version of particle swarm optimization for flowshop scheduling problems. *Computers & Operations Research*, 34(10), pp.3099–3111.
- Liu, S.-H., Mernik, M. and Bryant, B.R., 2009. To explore or to exploit: An entropy-driven approach for evolutionary algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 13(3–4), pp.185–206.
- Lu, X. and Zhou, Y., 2008. A Novel Global Convergence Algorithm: Bee Collecting Pollen Algorithm. In: D.-S. Huang, D.C. Wunsch, D.S. Levine and K.-H. Jo, eds. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*. Springer Berlin Heidelberg, pp.518–525.
- Lurton, X., 2002. *An introduction to underwater acoustics: principles and applications*. Springer Science & Business Media.
- M. Alauddin, 2016. Mosquito flying optimization (MFO). In: *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*. 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). pp.79–84.
- M. Bidar, H. R. Kanan, M. Mouhoub and S. Sadaoui, 2018. Mushroom Reproduction Optimization (MRO): A Novel Nature-Inspired Evolutionary Algorithm. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018 IEEE Congress on Evolutionary Computation (CEC). pp.1–10.
- M. Bidar and H. Rashidy Kanan, 2013. Jumper firefly algorithm. In: *ICCKE 2013*. ICCKE 2013. pp.267–271.
- M. C. Catalbas and A. Gulten, 2018. Circular structures of puffer fish: A new metaheuristic optimization algorithm. In: *2018 Third International Conference on Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC)*. 2018 Third International Conference on Electrical and Biomedical Engineering, Clean Energy and Green Computing (EBECEGC). IEEE, pp.1–5.
- M. Dehghani, Z. Montazeri, A. Dehghani and A. Seifi, 2017. Spring search algorithm: A new meta-heuristic optimization algorithm inspired by Hooke's law. In: *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI). pp.0210–0214.
- M. M. al-Rifaie and A. Aber, 2012. Identifying metastasis in bone scans with Stochastic Diffusion Search. In: *2012 International Symposium on Information Technologies in Medicine and Education*. 2012 International Symposium on Information Technologies in Medicine and Education. pp.519–523.
- M. Shen, Z. Zhan, W. Chen, Y. Gong, J. Zhang and Y. Li, 2014. Bi-Velocity Discrete Particle Swarm Optimization and Its Application to Multicast Routing Problem in Communication Networks. *IEEE Transactions on Industrial Electronics*, 61(12), pp.7141–7151.
- Maaranen, H., Miettinen, K. and Mäkelä, M.M., 2004. Quasi-random initial population for genetic algorithms. *Computers & Mathematics with Applications*, 47(12), pp.1885–1895.

Mahdavi, M., Fesanghary, M. and Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188(2), pp.1567–1579.

Mahmood, M. and Al-Khateeb, B., 2019. The Blue Monkey: A New Nature Inspired Metaheuristic Optimization Algorithm | Mahmood | Periodicals of Engineering and Natural Sciences. *Periodicals of Engineering and Natural Sciences*, 7(3), pp.1054–1066.

Marinakis, Y. and Marinaki, M., 2011. Bumble Bees Mating Optimization Algorithm for the Vehicle Routing Problem. In: B.K. Panigrahi, Y. Shi and M.-H. Lim, eds. *Handbook of Swarm Intelligence: Concepts, Principles and Applications*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.347–369. Available at: <https://doi.org/10.1007/978-3-642-17390-5_15>.

Markowitz, H., 1952. Portfolio selection. *The journal of finance*, 7(1), pp.77–91.

Masadeh, R., 2019. Sea Lion Optimization Algorithm. *International Journal of Advanced Computer Science and Applications*, 10(5), pp.388–395.

Masadeh, R., Sharieh, A. and Mahafzah, B., 2019. Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing. *Int J Adv Sci Technol*, 13(3), pp.121–140.

McCarthy, J., Minsky, M.L. and Rochester, N., 1955. A PROPOSAL FOR THE DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE.

Mehrabian, A.R. and Lucas, C., 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, 1(4), pp.355–366.

Mejía-de-Dios, J.-A. and Mezura-Montes, E., 2019. A New Evolutionary Optimization Method Based on Center of Mass. In: K. Deep, M. Jain and S. Salhi, eds. *Decision Science in Action: Theory and Applications of Modern Decision Analytic Optimisation*. [online] Singapore: Springer Singapore. pp.65–74. Available at: <https://doi.org/10.1007/978-981-13-0860-4_6>.

Meng, X., Liu, Y., Gao, X. and Zhang, H., 2014. A new bio-inspired algorithm: chicken swarm optimization. In: *International conference in swarm intelligence*. Springer. pp.86–94.

Meng, X.-B., Gao, X.Z., Lu, L., Liu, Y. and Zhang, H., 2016. A new bio-inspired optimisation algorithm: Bird Swarm Algorithm. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(4), pp.673–687.

Merrikh-Bayat, F., 2015. The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature. *Applied Soft Computing*, 33, pp.292–303.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E., 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), pp.1087–1092.

Mezura-Montes, E. and Coello, C.A.C., 2008. An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. *International Journal of General Systems*, 37(4), pp.443–473.

Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. eds., 1983. *Machine Learning: An Artificial Intelligence Approach*. Artificial Intelligence. [online] Berlin Heidelberg: Springer-Verlag. Available at: <<https://www.springer.com/gp/book/9783662124079>> [Accessed 19 Apr. 2019].

Michalski, S.R., Carbonell, G.J. and Mitchell, M.T. eds., 1986. *Machine Learning an Artificial Intelligence Approach Volume II*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Minhas, F. ul A.A. and Arif, M., 2011. MOX: A novel global optimization algorithm inspired from Oviposition site selection and egg hatching inhibition in mosquitoes. *Applied Soft Computing*, 11(8), pp.4614–4625.

Mirjalili, S., 2015a. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 89, pp.228–249.

Mirjalili, S., 2015b. The Ant Lion Optimizer. *Advances in Engineering Software*, 83, pp.80–98.

Mirjalili, S., 2016a. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Computing and Applications*, 27(4), pp.1053–1073.

Mirjalili, S., 2016b. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, pp.120–133.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M., 2017. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114, pp.163–191.

Mirjalili, S. and Lewis, A., 2016. The whale optimization algorithm. *Advances in engineering software*, 95, pp.51–67.

Mirjalili, S., Mirjalili, S.M. and Hatamlou, A., 2016. Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 27(2), pp.495–513.

Mirjalili, S., Mirjalili, S.M. and Lewis, A., 2014. Grey Wolf Optimizer. *Advances in Engineering Software*, 69, pp.46–61.

Mirjalili, S.Z., Mirjalili, S., Saremi, S., Faris, H. and Aljarah, I., 2018. Grasshopper optimization algorithm for multi-objective optimization problems. *Applied Intelligence*, 48(4), pp.805–820.

Mitchell, T.M., 1997. *Machine Learning, McGraw-Hill Companies*. Inc.

Moein, S. and Logeswaran, R., 2014. KGMO: A swarm optimization algorithm based on the kinetic energy of gas molecules. *Information Sciences*, 275, pp.127–144.

Moez, H., Kaveh, A. and Taghizadieh, N., 2016. Natural Forest Regeneration Algorithm: A New Meta-Heuristic. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 40(4), pp.311–326.

Moghdani, R. and Salimifard, K., 2018. Volleyball Premier League Algorithm. *Applied Soft Computing*, 64, pp.161–185.

Motevali, M.M., Shanghoosabad, A.M., Aram, R.Z. and Keshavarz, H., 2018. WHO: A New Evolutionary Algorithm Bio-Inspired by Wildebeests with a Case Study on Bank Customer Segmentation. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(05), p.1959017.

Mozaffari, A., Fathi, A. and Behzadipour, S., 2012. The great salmon run: a novel bio-inspired algorithm for artificial system design and optimisation. *International Journal of Bio-Inspired Computation*, 4(5), p.286.

Mucherino, A. and Seref, O., 2007. Monkey search: a novel metaheuristic search for global optimization. In: *AIP conference proceedings*. AIP.pp.162–173.

Muller, S.D., Marchetto, J., Airaghi, S. and Kournoutsakos, P., 2002. Optimization based on bacterial chemotaxis. *IEEE Transactions on Evolutionary Computation*, 6(1), pp.16–29.

N. Pang, Y. Shi and Y. You, 2008. Resource Leveling Optimization of Network Schedule Based on Particle Swarm Optimization with Constriction Factor. In: *2008 International Conference on Advanced Computer Theory and Engineering*. 2008 International Conference on Advanced Computer Theory and Engineering. pp.652–656.

Naderi, E., Azizivahed, A., Narimani, H., Fathi, M. and Narimani, M.R., 2017. A comprehensive study of practical economic dispatch problems by a new hybrid evolutionary algorithm. *Applied Soft Computing*, 61, pp.1186–1206.

Nasir, A.N.K., Tokhi, M.O., Sayidmarie, O. and Ismail, R.R., 2013. A novel adaptive spiral dynamic algorithm for global optimization. In: *2013 13th UK Workshop on Computational Intelligence (UKCI)*. IEEE.pp.334–341.

Nematollahi, A.F., Rahiminejad, A. and Vahidi, B., 2017. A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization. *Applied Soft Computing*, 59, pp.596–621.

Neme, A. and Hernández, S., 2009. Algorithms Inspired in Social Phenomena. In: R. Chiong, ed. *Nature-Inspired Algorithms for Optimisation*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg.pp.369–387. Available at: <https://doi.org/10.1007/978-3-642-00267-0_13>.

Neshat, M., Sepidnam, G. and Sargolzaei, M., 2013. Swallow swarm optimization algorithm: a new method to optimization. *Neural Computing and Applications*, 23(2), pp.429–454.

Neumann, K. and Zimmermann, J., 1999. Resource levelling for projects with schedule-dependent time windows. *European Journal of Operational Research*, 117(3), pp.591–605.

Neumann, K. and Zimmermann, J., 2000. Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints. *European Journal of Operational Research*, 127(2), pp.425–443.

Nguyen, H.T. and Bhanu, B., 2012. Zombie Survival Optimization: A swarm intelligence algorithm inspired by zombie foraging. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE.pp.987–990.

Nilsson, M. and Snoad, N., 2002. Optimal mutation rates in dynamic environments. *Bulletin of Mathematical Biology*, 64(6), p.1033.

O. Olorunda and A. P. Engelbrecht, 2008. Measuring exploration/exploitation in particle swarms using swarm diversity. In: *2008 IEEE Congress on Evolutionary Computation (IEEE*

World Congress on Computational Intelligence). 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). pp.1128–1134.

Odili, J.B. and Kahar, M.N.M., 2015. African buffalo optimization (ABO): a new meta-heuristic algorithm. *Journal of Advanced & Applied Sciences*, 3(03), pp.101–106.

Oftadeh, R., Mahjoob, M.J. and Shariatpanahi, M., 2010. A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search. *Computers & Mathematics with Applications*, 60(7), pp.2087–2098.

Ollion, C. and Doncieux, S., 2011. Why and How to Measure Exploration in Behavioral Space. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, GECCO '11. [online] New York, NY, USA: ACM.pp.267–274. Available at: <<http://doi.acm.org/10.1145/2001576.2001613>>.

Osuna-Enciso, V., Cuevas, E., Oliva, D., Sossa, H. and Cisneros, M.A.P., 2016. A bio-inspired evolutionary algorithm: allostatic optimisation. *IJBIC*, 8(3), pp.154–169.

Pan, W.-T., 2011. Fruit fly optimization algorithm. *Tsang Hai Book Publishing Co., Taipei*, pp.10–12.

Papadopoulos, C.T., O'Kelly, M.E.J., Vidalis, M.J. and Spinellis, D., 2009. *Analysis and Design of Discrete Part Production Lines*. Springer Optimization and Its Applications. [online] New York: Springer-Verlag. Available at: <<https://www.springer.com/gp/book/9780387894935>> [Accessed 7 Aug. 2019].

Passino, K.M., 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), pp.52–67.

Patel, V.K. and Savsani, V.J., 2015. Heat transfer search (HTS): a novel optimization algorithm. *Information Sciences*, 324, pp.217–246.

Pattnaik, S.S., Bakwad, K.M., Sohi, B.S., Ratho, R.K. and Devi, S., 2013. Swine Influenza Models Based Optimization (SIMBO). *Applied Soft Computing*, 13(1), pp.628–653.

Pedrycz, W., 1997. *Computational intelligence: an introduction*. CRC press.

Pham, D.T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S. and Zaidi, M., 2006. - The Bees Algorithm — A Novel Tool for Complex Optimisation Problems. In: D.T. Pham, E.E. Eldukhri and A.J. Soroka, eds. *Intelligent Production Machines and Systems*. [online] Oxford: Elsevier Science Ltd.pp.454–459. Available at: <<http://www.sciencedirect.com/science/article/pii/B978008045157250081X>>.

Połap, D. and Wozniak, M., 2017. Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry*, 9(10).

Premaratne, U., Samarabandu, J. and Sidhu, T., 2009. A new biologically inspired optimization algorithm. In: *2009 international conference on industrial and information systems (ICIIS)*. 2009 International Conference on Industrial and Information Systems (ICIIS). IEEE.pp.279–284.

Qi, X., Zhu, Y., Chen, H., Zhang, D. and Niu, B., 2013. An Idea Based on Plant Root Growth for Numerical Optimization. In: D.-S. Huang, K.-H. Jo, Y.-Q. Zhou and K. Han, eds. *Intelligent Computing Theories and Technology*. Springer Berlin Heidelberg.pp.571–578.

Qi, X., Zhu, Y. and Zhang, H., 2017. A new meta-heuristic butterfly-inspired algorithm. *Journal of Computational Science*, 23, pp.226–239.

Qiao, P. and Duan, H., 2014. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning. *International Journal of Intelligent Computing and Cybernetics*, 7(1), pp.24–37.

R. C. Eberhart and Xiaohui Hu, 1999. Human tremor analysis using particle swarm optimization. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). pp.1927-1930 Vol. 3.

R. D. Maia, L. N. de Castro and W. M. Caminhas, 2012. Bee colonies as model for multimodal continuous optimization: The OptBees algorithm. In: *2012 IEEE Congress on Evolutionary Computation*. 2012 IEEE Congress on Evolutionary Computation. pp.1–8.

R. Gowri and R. Rathipriya, 2018. Non-Swarm Plant Intelligence Algorithm: BladderWorts Suction (BWS) Algorithm. In: *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*. 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET). pp.1–7.

R. Omidvar, H. Parvin and F. Rad, 2015. SSPCO Optimization Algorithm (See-See Partridge Chicks Optimization). In: *2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI)*. 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI). pp.101–106.

R. S. Parpinelli and H. S. Lopes, 2011. An eco-inspired evolutionary algorithm applied to numerical optimization. In: *2011 Third World Congress on Nature and Biologically Inspired Computing*. 2011 Third World Congress on Nature and Biologically Inspired Computing. pp.466–471.

Rabanal, P., Rodríguez, I. and Rubio, F., 2007. Using River Formation Dynamics to Design Heuristic Algorithms. In: S.G. Akl, C.S. Calude, M.J. Dinneen, G. Rozenberg and H.T. Wareham, eds. *Unconventional Computation*, Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp.163–177.

Ragsdell, K.M. and Phillips, D.T., 1976. Optimal Design of a Class of Welded Structures Using Geometric Programming. *Journal of Engineering for Industry*, 98(3), pp.1021–1025.

Rahman, I. and Mohamad-Saleh, J., 2018. Hybrid bio-Inspired computational intelligence techniques for solving power system optimization problems: A comprehensive survey. *Applied Soft Computing*, 69, pp.72–130.

Rahnamayan, S., Tizhoosh, H.R. and Salama, M.M.A., 2007. A novel population initialization method for accelerating evolutionary algorithms. *Computers & Mathematics with Applications*, 53(10), pp.1605–1614.

Rajakumar, B.R., 2012. The Lion's Algorithm: A New Nature-Inspired Search Algorithm. *2nd International Conference on Communication, Computing & Security [ICCCS-2012]*, 6, pp.126–135.

Rashedi, E., Nezamabadi-Pour, H. and Saryazdi, S., 2009. GSA: a gravitational search algorithm. *Information sciences*, 179(13), pp.2232–2248.

- Rastrigin, L.A., 1974. Extremal control systems. *Theoretical foundations of engineering cybernetics series*, 3.
- Ravikumar Pandi, V. and Panigrahi, B.K., 2011. Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm. *Expert Systems with Applications*, 38(7), pp.8509–8514.
- Rbough, I. and Imrani, A.A.E., 2014. Hurricane-based Optimization Algorithm. *AASRI Procedia*, 6, pp.26–33.
- Rieck, J., Zimmermann, J. and Gather, T., 2012. Mixed-integer linear programming for resource leveling problems. *European Journal of Operational Research*, 221(1), pp.27–37.
- Rojas, R., 2013. *Neural Networks: A Systematic Introduction*. Springer Science & Business Media.
- Rosenblatt, F., 1959. *Two theorems of statistical separability in the perceptron. Mechanisation of thought processes: Proceedings of a symposium held at the National Physical Laboratory, ed. by DV Blake and Albert M. Uttley, 419-56*. London: Her Majesty's Stationery Office.
- Rudolph, G., 1990. *Globale Optimierung mit parallelen Evolutionsstrategien*. PhD Thesis. Diplomarbeit, Universität Dortmund, Fachbereich Informatik.
- S. Arora and S. Singh, 2015. Butterfly algorithm with Lévy Flights for global optimization. In: *2015 International Conference on Signal Processing, Computing and Control (ISPCC)*. 2015 International Conference on Signal Processing, Computing and Control (ISPCC). pp.220–224.
- S. Deb, S. Fong and Z. Tian, 2015. Elephant Search Algorithm for optimization problems. In: *2015 Tenth International Conference on Digital Information Management (ICDIM)*. 2015 Tenth International Conference on Digital Information Management (ICDIM). pp.249–255.
- S. He, Q. H. Wu and J. R. Saunders, 2006. A Novel Group Search Optimizer Inspired by Animal Behavioural Ecology. In: *2006 IEEE International Conference on Evolutionary Computation*. 2006 IEEE International Conference on Evolutionary Computation. pp.1272–1278.
- S. He, Q. H. Wu and J. R. Saunders, 2009. Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. *IEEE Transactions on Evolutionary Computation*, 13(5), pp.973–990.
- S. Mohseni, R. Gholami, N. Zarei and A. R. Zadeh, 2014. Competition over Resources: A New Optimization Algorithm Based on Animals Behavioral Ecology. In: *2014 International Conference on Intelligent Networking and Collaborative Systems*. 2014 International Conference on Intelligent Networking and Collaborative Systems. pp.311–315.
- S. Sahni and T. Gonzales, 1974. P-complete problems and approximate solutions. In: *15th Annual Symposium on Switching and Automata Theory (swat 1974)*. 15th Annual Symposium on Switching and Automata Theory (swat 1974). pp.28–32.
- S. Zangbari Koochi, N. A. W. Abdul Hamid, M. Othman and G. Ibragimov, 2019. Raccoon Optimization Algorithm. *IEEE Access*, 7, pp.5383–5399.

S. Zhao, Y. Liu, H. Zhao and R. Zhou, 2006. GA-Based Resource Leveling Optimization for Construction Project. In: *2006 International Conference on Machine Learning and Cybernetics*. 2006 International Conference on Machine Learning and Cybernetics. pp.2363–2367.

Sacco, W.F. and Oliveira, C.R.E.D., 2005. A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic. In: *Proceedings of 6th WCSMO*. 6th World Congresses of Structural and Multidisciplinary Optimization. Rio de Janeiro, 30 May - 03 June 2005, Brazil.

Sadollah, A., Bahreininejad, A., Eskandar, H. and Hamdi, M., 2012. Mine blast algorithm for optimization of truss structures with discrete variables. *Computers & Structures*, 102–103, pp.49–63.

Salcedo-Sanz, S., Del Ser, J., Landa-Torres, I., Gil-López, S. and Portilla-Figueras, J.A., 2014. The Coral Reefs Optimization Algorithm: A Novel Metaheuristic for Efficiently Solving Optimization Problems. *The Scientific World Journal*, [online] 2014. Available at: <<https://www.hindawi.com/journals/tswj/2014/739768/cta/>> [Accessed 14 Jul. 2019].

Salhi, A. and Fraga, E.S., 2011. Nature-Inspired Optimisation Approaches and the New Plant Propagation Algorithm. [online] Proceedings of the The International Conference on Numerical Analysis and Optimization (ICeMATH 2011). Yogyakarta, Indonesia.p.K271-K278. Available at: <<http://repository.essex.ac.uk/12110/>> [Accessed 6 Jun. 2019].

Salimi, H., 2015. Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*, 75, pp.1–18.

Salmani, M.H. and Eshghi, K., 2017. A Metaheuristic Algorithm Based on Chemotherapy Science: CSA. *Journal of Optimization*. [online] Available at: <<https://www.hindawi.com/journals/jopti/2017/3082024/cta/>> [Accessed 3 Jun. 2019].

Samareh Moosavi, S.H. and Khatibi Bardsiri, V., 2017. Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. *Engineering Applications of Artificial Intelligence*, 60, pp.1–15.

Savitch, W.J., 1970. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2), pp.177–192.

Selim, S.Z. and Alsultan, K., 1991. A simulated annealing algorithm for the clustering problem. *Pattern Recognition*, 24(10), pp.1003–1008.

Serani, A. and Diez, M., 2018. Dolphin Pod Optimization. In: G. Nicosia, P. Pardalos, G. Giuffrida and R. Umeton, eds. *Machine Learning, Optimization, and Big Data*. Springer International Publishing, pp.50–62.

Shadravan, S., Naji, H.R. and Bardsiri, V.K., 2019. The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. *Engineering Applications of Artificial Intelligence*, 80, pp.20–34.

Shah-Hosseini, H., 2009. The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation*, 1(1–2), pp.71–79.

- Shah-Hosseini, H., 2011. Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering*, 6(1–2), pp.132–140.
- Shamsaldin, A.S., Rashid, T.A., Al-Rashid Agha, R.A., Al-Salihi, N.K. and Mohammadi, M., 2019. Donkey and Smuggler Optimization Algorithm: A Collaborative Working Approach to Path Finding. *Journal of Computational Design and Engineering*. [online] Available at: <<http://www.sciencedirect.com/science/article/pii/S2288430018303178>>.
- Shareef, H., Ibrahim, A.A. and Mutlag, A.H., 2015. Lightning search algorithm. *Applied Soft Computing*, 36, pp.315–333.
- Sharpe, W.F., 1994. The Sharpe Ratio. *The Journal of Portfolio Management*, 21(1), p.49.
- Shayanfar, H. and Gharehchopogh, F.S., 2018. Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems. *Applied Soft Computing*, 71, pp.728–746.
- Shirzadi, M.T.M.H. and Bagheri, M.H., 2012. A novel meta-heuristic algorithm for numerical function optimization: Blind, naked mole-rats (BNMR) algorithm. *Scientific Research and Essays*, 7(41), pp.3566–3583.
- Shmygelska, A. and Hoos, H.H., 2005. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(1), p.30.
- Siddique, N. and Adeli, H., 2016. Physics-based search and optimization: Inspirations from nature. *Expert Systems*, 33(6), pp.607–623.
- Simon, D., 2008. Biogeography-based optimization. *IEEE transactions on evolutionary computation*, 12(6), pp.702–713.
- Sipser, M., 2006. 7. Time Complexity. In: *Introduction to the Theory of Computation*, Second. Thomson Course Technology Boston. pp.245–294.
- Song, Y., Liu, L., Ma, H. and Vasilakos, A.V., 2012. Physarum Optimization: A New Heuristic Algorithm to Minimal Exposure Problem. In: *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12*. [online] New York, NY, USA: ACM. pp.419–422. Available at: <<http://doi.acm.org/10.1145/2348543.2348599>>.
- Sörensen, K., 2015. Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*, 22(1), pp.3–18.
- Sprecher, A. and Kolisch, R., 1996. PSPLIB—a project scheduling problem library. *European Journal of Operational Research*, 96, pp.205–216.
- Steer, K.C.B., Wirth, A. and Halgamuge, S.K., 2009. The Rationale Behind Seeking Inspiration from Nature. In: R. Chiong, ed. *Nature-Inspired Algorithms for Optimisation*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.51–76. Available at: <https://doi.org/10.1007/978-3-642-00267-0_2>.
- Storn, R. and Price, K., 1997. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), pp.341–359.

Stützle, T. and Dorigo, M., 1999. ACO algorithms for the quadratic assignment problem. *New Ideas in Optimization*. [online] Available at: <<https://ci.nii.ac.jp/naid/10022007004/en/>>.

Subramanian, C., Sekar, A. and Subramanian, K., 2013. A new engineering optimization method: African wild dog algorithm. *International Journal of Soft Computing*, 8(3), pp.163–170.

Sulaiman, M. and Salhi, A., 2015. A Seed-Based Plant Propagation Algorithm: The Feeding Station Model. *The Scientific World Journal*. [online] Available at: <<https://www.hindawi.com/journals/tswj/2015/904364/abs/>> [Accessed 6 Jun. 2019].

Sulaiman, M.H., Mustaffa, Z., Saari, M.M., Daniyal, H., Musirin, I. and Daud, M.R., 2018. Barnacles Mating Optimizer: An Evolutionary Algorithm for Solving Optimization. In: *2018 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE. pp.99–104.

Sur, C., Sharma, S. and Shukla, A., 2013. Egyptian vulture optimization algorithm—a new nature inspired meta-heuristics for knapsack problem. In: *The 9th International Conference on Computing and Information Technology (IC2IT2013)*. Springer. pp.227–237.

Sur, C. and Shukla, A., 2013. New Bio-inspired Meta-Heuristics - Green Herons Optimization Algorithm - for Optimization of Travelling Salesman Problem and Road Network. In: B.K. Panigrahi, P.N. Suganthan, S. Das and S.S. Dash, eds. *Swarm, Evolutionary, and Memetic Computing*. Springer International Publishing. pp.168–179.

T. C. Havens, C. J. Spain, N. G. Salmon and J. M. Keller, 2008. Roach Infestation Optimization. In: *2008 IEEE Swarm Intelligence Symposium*. 2008 IEEE Swarm Intelligence Symposium. pp.1–7.

Tahani, M. and Babayan, N., 2018. Flow Regime Algorithm (FRA): a physics-based meta-heuristics algorithm. *Knowledge and Information Systems*. [online] Available at: <<https://doi.org/10.1007/s10115-018-1253-3>>.

Taherdangkoo, M., Yazdi, M. and Bagheri, M.H., 2012. Stem Cells Optimization Algorithm. In: D.-S. Huang, Y. Gan, P. Premaratne and K. Han, eds. *Bio-Inspired Computing and Applications*. Springer Berlin Heidelberg. pp.394–403.

Tan, Y. and Zhu, Y., 2010. Fireworks Algorithm for Optimization. In: Y. Tan, Y. Shi and K.C. Tan, eds. *Advances in Swarm Intelligence*. Springer Berlin Heidelberg. pp.355–364.

Tang, D., Dong, S., Jiang, Y., Li, H. and Huang, Y., 2015. ITGO: Invasive tumor growth optimization algorithm. *Applied Soft Computing*, 36, pp.670–698.

Tang, R., Fong, S., Yang, X. and Deb, S., 2012. Wolf search algorithm with ephemeral memory. In: *Seventh International Conference on Digital Information Management (ICDIM 2012)*. Seventh International Conference on Digital Information Management (ICDIM 2012). pp.165–172.

Tang, W.J., Wu, Q.H. and Saunders, J.R., 2007. A bacterial swarming algorithm for global optimization. In: *2007 IEEE Congress on Evolutionary Computation*. 2007 IEEE Congress on Evolutionary Computation. pp.1207–1212.

Tawfeeq, M.A., 2012. Intelligent Algorithm for Optimum Solutions Based on the Principles of Bat Sonar. *arXiv:1211.0730 [cs]*. [online] Available at: <<http://arxiv.org/abs/1211.0730>> [Accessed 17 Apr. 2019].

Teodorovic, D. and Dell’Orco, M., 2005. Bee colony optimization—a cooperative learning approach to complex transportation problems. *Advanced OR and AI methods in transportation*, 51, p.60.

Thomaidis, N.S., Angelidis, T., Vassiliadis, V. and Dounias, G., 2009. Active portfolio management with cardinality constraints: An application of particle swarm optimization. *New Mathematics and Natural Computation*, 5(03), pp.535–555.

Tilahun, S.L. and Ong, H.C., 2013. Prey-Predator Algorithm: A New Metaheuristic Algorithm for Optimization Problems. *International Journal of Information Technology & Decision Making*, 14(06), pp.1331–1352.

Ting, T.O., Man, K.L., Guan, S.-U., Nayel, M. and Wan, K., 2012. Weightless Swarm Algorithm (WSA) for Dynamic Optimization Problems. In: J.J. Park, A. Zomaya, S.-S. Yeo and S. Sahni, eds. *Network and Parallel Computing*, Lecture Notes in Computer Science. Springer Berlin Heidelberg. pp.508–515.

Tripathi, A.K., Sharma, K. and Bala, M., 2019. Military Dog Based Optimizer and its Application to Fake Review. [online] Available at: <<https://arxiv.org/abs/1909.11890>>.

Tsakonas, A., Papadopoulos, C. and Dounias, G., 2001. The throughput rate of short exponential production lines with finite intermediate buffers using genetic programming approximation techniques. 6th International Conference “Measurement and Control in Complex Systems”. Ukraine, 2001.

Tzanetos, A. and Dounias, G., 2017a. A New Metaheuristic Method for Optimization: Sonar Inspired Optimization. In: G. Boracchi, L. Iliadis, C. Jayne and A. Likas, eds. *Engineering Applications of Neural Networks*. Springer International Publishing. pp.417–428.

Tzanetos, A. and Dounias, G., 2017b. Nature Inspired Optimization Algorithms Related to Physical Phenomena and Laws of Science: A Survey. *International Journal on Artificial Intelligence Tools*, 26(06), p.1750022.

Tzanetos, A. and Dounias, G., 2018. Sonar inspired optimization (SIO) in engineering applications. *Evolving Systems*, pp.1–9.

Tzanetos, A. and Dounias, G., 2019. *An application-based taxonomy of Nature Inspired Intelligent algorithms*. [Research Report] Chios, Greece: Management and Decision Engineering Laboratory (MDE-Lab) University of the Aegean, School of Engineering, Dept. of Financial and Management Engineering. Available at: <<http://mde-lab.aegean.gr/images/stories/docs/reportnii2019.pdf>>.

Tzanetos, A. and Dounias, G., 2020. A comprehensive survey on the applications of Swarm Intelligence and Bio-inspired evolutionary strategies (to appear). In: G.A. Tsihrintzis and L.C. Jain, eds. *Machine Learning Paradigms: Advances in Theory and Applications of Deep Learning*, Intelligent Systems Reference Library. [online] Cham: Springer International Publishing. Available at: <https://doi.org/10.1007/978-3-030-13743-4_1> [Accessed 7 Nov. 2019].

Tzanetos, A., Kyriklidis, C., Papamichail, A., Dimoulakis, A. and Dounias, G., 2018. A Nature Inspired metaheuristic for Optimal Leveling of Resources in Project Management. In: *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. ACM. p.17.

Tzanetos, A., Vassiliadis, V. and Dounias, G., 2017. A novel hybrid Nature-Inspired Scheme for solving a financial optimization problem. In: *International Conference on Hybrid Artificial Intelligence Systems*. Springer. pp.161–172.

Tzanetos, A., Vassiliadis, V. and Dounias, G., 2018. Boosting the performance of hybrid Nature-Inspired algorithms: Application from the financial optimization domain. *Logic Journal of the IGPL*.

Uymaz, S.A., Tezel, G. and Yel, E., 2015. Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, pp.153–171.

V. Bhasin, P. Bedi and A. Singhal, 2014. Feature selection for steganalysis based on modified Stochastic Diffusion Search using Fisher score. In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI). pp.2323–2330.

Valdez, F., Melin, P. and Castillo, O., 2014. Modular Neural Networks architecture optimization with a new nature inspired method using a fuzzy combination of Particle Swarm Optimization and Genetic Algorithms. *Information Sciences*, 270, pp.143–153.

Van der Pas, P.W., 1971. The discovery of the Brownian motion. *Scientiarum Historia: Tijdschrift voor de Geschiedenis van de Wetenschappen en de Geneeskunde*, 13(1), pp.27–35.

Varaee, H. and Ghasemi, M.R., 2017. Engineering optimization based on ideal gas molecular movement algorithm. *Engineering with Computers*, 33(1), pp.71–93.

Vasconcelos Segundo, E.H. de, Mariani, V.C. and Santos Coelho, L. dos, 2019. Design of heat exchangers using falcon optimization algorithm. *Applied Thermal Engineering*. [online] Available at: <<http://www.sciencedirect.com/science/article/pii/S1359431119301644>>.

Vassiliadis, V., Thomaidis, N. and Dounias, G., 2009. Active Portfolio Management under a Downside Risk Framework: Comparison of a Hybrid Nature – Inspired Scheme. In: E. Corchado, X. Wu, E. Oja, Á. Herrero and B. Baruque, eds. *Hybrid Artificial Intelligence Systems*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.702–712. Available at: <https://link.springer.com/chapter/10.1007/978-3-642-02319-4_85>.

W. Cai, W. Yang and X. Chen, 2008. A Global Optimization Algorithm Based on Plant Growth Theory: Plant Growth Optimization. In: *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*. 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA). pp.1194–1199.

W. Li and M. Jiang, 2018. Fuzzy-Based Lion Pride optimization for Grayscale Image Segmentation. In: *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*. 2018 IEEE International Conference of Safety Produce Informatization (IICSPI). pp.600–604.

W. Wen, C. Wang, D. Wu and Y. Xie, 2015. An ACO-based Scheduling Strategy on Load Balancing in Cloud Computing Environment. In: *2015 Ninth International Conference on Frontier of Computer Science and Technology*. 2015 Ninth International Conference on Frontier of Computer Science and Technology. pp.364–369.

Wang, B., Jin, X. and Cheng, B., 2012. Lion pride optimizer: An optimization algorithm inspired by lion pride behavior. *Science China Information Sciences*, 55(10), pp.2369–2389.

Wang, G.-G., Deb, S. and Coelho, L. dos S., 2015a. Elephant herding optimization. In: *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*. IEEE, pp.1–5.

Wang, G.-G., Deb, S. and Coelho, L.D.S., 2015b. Earthworm optimization algorithm: a bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation*, 7, pp.1–23.

Wang, G.-G., Deb, S. and Cui, Z., 2015. Monarch butterfly optimization. *Neural Computing and Applications*. [online] Available at: <<https://doi.org/10.1007/s00521-015-1923-y>> [Accessed 11 Apr. 2019].

Wang, P., Zhu, Z. and Huang, S., 2013. Seven-Spot Ladybird Optimization: A Novel and Efficient Metaheuristic Algorithm for Numerical Optimization. *The Scientific World Journal*. [online] Available at: <<https://www.hindawi.com/journals/tswj/2013/378515/cta/>> [Accessed 6 Jun. 2019].

Wang, T., Yang, L. and Liu, Q., 2018. Beetle Swarm Optimization Algorithm: Theory and Application. *arXiv:1808.00206 [cs]*. [online] Available at: <<http://arxiv.org/abs/1808.00206>> [Accessed 17 Apr. 2019].

Wang, X., Chen, Q., Zou, R. and Huang, M., 2008. An ABC Supported QoS Multicast Routing Scheme Based on Beehive Algorithm. In: *Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, QShine '08*. [online] ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp.23:1–23:7. Available at: <<http://dl.acm.org/citation.cfm?id=1535571.1535602>>.

Wang, Y., Lv, J., Zhu, L. and Ma, Y., 2010. Crystal structure prediction via particle-swarm optimization. *Phys. Rev. B*, 82(9), p.094116.

Wedde, H.F., Farooq, M. and Zhang, Y., 2004. BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior. In: M. Dorigo, M. Birattari, C. Blum, L.M. Gambardella, F. Mondada and T. Stützle, eds. *Ant Colony Optimization and Swarm Intelligence*. Springer Berlin Heidelberg, pp.83–94.

Wedyan, A., Whalley, J. and Narayanan, A., 2017. Hydrological cycle algorithm for continuous optimization problems. *Journal of Optimization*, [online] 2017. Available at: <<https://doi.org/10.1155/2017/3828420>>.

Weise, T., Zapf, M., Chiong, R. and Nebro, A.J., 2009. Why Is Optimization Difficult? In: R. Chiong, ed. *Nature-Inspired Algorithms for Optimisation*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg, pp.1–50. Available at: <https://doi.org/10.1007/978-3-642-00267-0_1>.

Werbos, P.J., 1974. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD Thesis. Harvard University.

Widrow, B. and Hoff, M.E., 1960. *Adaptive switching circuits*. Stanford Univ Ca Stanford Electronics Labs.

Wolpert, D.H. and Macready, W.G., 1995. *No Free Lunch Theorems for Search*. [online] Santa Fe Institute. Available at: <<https://EconPapers.repec.org/RePEc:wop:safiwp:95-02-010>>.

Wu, H.-S. and Zhang, F.-M., 2014. Wolf pack algorithm for unconstrained global optimization. *Mathematical Problems in Engineering*, [online] 2014. Available at: <<https://www.hindawi.com/journals/mpe/2014/465082/cta/>>.

Wu, S.-J. and Wu, C.-T., 2013. Computational optimization for S-type biological systems: Cockroach genetic algorithm. *Mathematical Biosciences*, 245(2), pp.299–313.

Wu, S.-J. and Wu, C.-T., 2015. A bio-inspired optimization for inferring interactive networks: Cockroach swarm evolution. *Expert Systems with Applications*, 42(6), pp.3253–3267.

Wu, T., Yao, M. and Yang, J., 2016. Dolphin swarm algorithm. *Frontiers of Information Technology & Electronic Engineering*, 17(8), pp.717–729.

X. Bao, H. Jia and C. Lang, 2019. A Novel Hybrid Harris Hawks Optimization for Color Image Multilevel Thresholding Segmentation. *IEEE Access*, 7, pp.76529–76546.

X. Yang and Suash Deb, 2009. Cuckoo Search via Lévy flights. In: *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC). pp.210–214.

Xie, L., Zeng, J. and Cui, Z., 2009. Using artificial physics to solve global optimization problems. In: *2009 8th IEEE International Conference on Cognitive Informatics*. 2009 8th IEEE International Conference on Cognitive Informatics. pp.502–508.

Xing, B. and Gao, W.-J., 2014. Emerging Physics-based CI Algorithms. In: B. Xing and W.-J. Gao, eds. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Intelligent Systems Reference Library. [online] Cham: Springer International Publishing. pp.375–414. Available at: <https://doi.org/10.1007/978-3-319-03404-1_24> [Accessed 15 Apr. 2019].

Y. Shiqin, J. Jianjun and Y. Guangxing, 2009. A Dolphin Partner Optimization. In: *2009 WRI Global Congress on Intelligent Systems*. 2009 WRI Global Congress on Intelligent Systems. IEEE. pp.124–128.

Yadav, A., 2019. AEFA: Artificial electric field algorithm for global optimization. *Swarm and Evolutionary Computation*.

Yan, G.-W. and Hao, Z.-J., 2013. A novel optimization algorithm based on atmosphere clouds model. *International Journal of Computational Intelligence and Applications*, 12(01), p.1350002.

Yang, X.-S., 2005. Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms. In: J. Mira and J.R. Álvarez, eds. *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach*. Springer Berlin Heidelberg. pp.317–323.

Yang, X.-S., 2009. Firefly Algorithms for Multimodal Optimization. In: O. Watanabe and T. Zeugmann, eds. *Stochastic Algorithms: Foundations and Applications*. Springer Berlin Heidelberg. pp.169–178.

Yang, X.-S., 2010. A New Metaheuristic Bat-Inspired Algorithm. In: J.R. González, D.A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor, eds. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Studies in Computational Intelligence. [online] Berlin, Heidelberg: Springer Berlin Heidelberg. pp.65–74. Available at: <https://doi.org/10.1007/978-3-642-12538-6_6> [Accessed 11 Apr. 2019].

- Yang, X.-S., 2012a. Flower Pollination Algorithm for Global Optimization. In: J. Durand-Lose and N. Jonoska, eds. *Unconventional Computation and Natural Computation*. Springer Berlin Heidelberg, pp.240–249.
- Yang, X.-S., 2012b. Nature-Inspired Metaheuristic Algorithms: Success and New Challenges. *Journal of Computer Engineering and Information Technology*, [online] 01(01). Available at: <<http://arxiv.org/abs/1211.6658>> [Accessed 7 Apr. 2019].
- Yang, X.-S., 2013. Metaheuristic optimization: Nature-inspired algorithms and applications. In: *Artificial Intelligence, Evolutionary Computing and Metaheuristics*. Springer, pp.405–420.
- Yang, X.-S., 2014. Chapter 1 - Introduction to Algorithms. In: X.-S. Yang, ed. *Nature-Inspired Optimization Algorithms*. [online] Oxford: Elsevier, pp.1–21. Available at: <<http://www.sciencedirect.com/science/article/pii/B9780124167438000014>>.
- Yang, X.-S., 2017. Social Algorithms. In: R.A. Meyers, ed. *Encyclopedia of Complexity and Systems Science*. [online] Berlin, Heidelberg: Springer Berlin Heidelberg, pp.1–15. Available at: <https://doi.org/10.1007/978-3-642-27737-5_678-1>.
- Yang, X.-S., 2018. Mathematical Analysis of Nature-Inspired Algorithms. In: X.-S. Yang, ed. *Nature-Inspired Algorithms and Applied Optimization*. [online] Cham: Springer International Publishing, pp.1–25. Available at: <https://doi.org/10.1007/978-3-319-67669-2_1>.
- Yang, X.-S. and He, X.-S., 2019a. Applications of Nature-Inspired Algorithms. In: X.-S. Yang and X.-S. He, eds. *Mathematical Foundations of Nature-Inspired Algorithms*, SpringerBriefs in Optimization. [online] Cham: Springer International Publishing, pp.87–97. Available at: <https://doi.org/10.1007/978-3-030-16936-7_6> [Accessed 20 May 2019].
- Yang, X.-S. and He, X.-S., 2019b. Mathematical Analysis of Algorithms: Part I. In: X.-S. Yang and X.-S. He, eds. *Mathematical Foundations of Nature-Inspired Algorithms*. [online] Cham: Springer International Publishing, pp.59–73. Available at: <https://doi.org/10.1007/978-3-030-16936-7_4>.
- Yang, X.-S., Lees, J.M. and Morley, C.T., 2006. Application of Virtual Ant Algorithms in the Optimization of CFRP Shear Strengthened Precracked Structures. In: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot and J. Dongarra, eds. *Computational Science – ICCS 2006*. Springer Berlin Heidelberg, pp.834–837.
- Yazdani, M. and Jolai, F., 2016. Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of computational design and engineering*, 3(1), pp.24–36.
- Yi, T.-H., Li, H.-N., Gu, M. and Zhang, X.-D., 2014. Sensor Placement Optimization in Structural Health Monitoring Using Niching Monkey Algorithm. *International Journal of Structural Stability and Dynamics*, 14(05), p.1440012.
- Yi, T.-H., Li, H.-N., Song, G. and Zhang, X.-D., 2015. Optimal sensor placement for health monitoring of high-rise structure using adaptive monkey algorithm. *Structural Control and Health Monitoring*, 22(4), pp.667–681.
- Yong, W., Tao, W., Cheng-Zhi, Z. and Hua-Juan, H., 2016. A New Stochastic Optimization Approach — Dolphin Swarm Optimization Algorithm. *International Journal of Computational Intelligence and Applications*, 15(02), p.1650011.

Younis, M.A. and Saad, B., 1996. Optimal resource leveling of multi-resource projects. *Proceedings of the 19th International Conference on Computers and Industrial Engineering*, 31(1), pp.1–4.

Yuce, B., Packianather, M., Mastrocinque, E., Pham, D. and Lambiase, A., 2013. Honey bees inspired optimization method: the bees algorithm. *Insects*, 4(4), pp.646–662.

Z. Bayraktar, M. Komurcu and D. H. Werner, 2010. Wind Driven Optimization (WDO): A novel nature-inspired optimization algorithm and its application to electromagnetics. In: *2010 IEEE Antennas and Propagation Society International Symposium*. 2010 IEEE Antennas and Propagation Society International Symposium. pp.1–4.

Z. Li, P. Wuliang and Z. Zhongliang, 2010. An Ant Colony System for Solving Resource Leveling Problem. In: *2010 International Conference on Intelligent Computation Technology and Automation*. 2010 International Conference on Intelligent Computation Technology and Automation. pp.489–492.

Z. Wei, C. Huang, X. Wang, T. Han and Y. Li, 2019. Nuclear Reaction Optimization: A Novel and Powerful Physics-Based Algorithm for Global Optimization. *IEEE Access*, 7, pp.66084–66109.

Z. Zhao, Z. Cui, J. Zeng and X. Yue, 2011. Artificial Plant Optimization Algorithm for Constrained Optimization Problems. In: *2011 Second International Conference on Innovations in Bio-inspired Computing and Applications*. 2011 Second International Conference on Innovations in Bio-inspired Computing and Applications. pp.120–123.

Zaldívar, D., Morales, B., Rodríguez, A., Valdivia-G, A., Cuevas, E. and Pérez-Cisneros, M., 2018. A novel bio-inspired optimization model based on Yellow Saddle Goatfish behavior. *Biosystems*, 174, pp.1–21.

Zarand, G., Pazmandi, F., Pál, K.F. and Zimányi, G.T., 2002. Using hysteresis for optimization. *Physical review letters*, 89(15), p.150201.

Zeng, B., Gao, L. and Li, X., 2017. Whale Swarm Algorithm for Function Optimization. In: D.-S. Huang, V. Bevilacqua, P. Premaratne and P. Gupta, eds. *Intelligent Computing Theories and Application*. Springer International Publishing. pp.624–639.

Zhang, H., Zhu, Y. and Chen, H., 2014. Root growth model: a novel approach to numerical function optimization and simulation of plant root system. *Soft Computing*, 18(3), pp.521–537.

Zhang, J., Zhang, Y. and Sun, J., 2011. Intrusion Detection Technology Based on Monkey Algorithm. *Computer Engineering*, 37(14), pp.131–133.

Zhang, Q., Wang, R., Yang, J., Lewis, A., Chiclana, F. and Yang, S., 2018. Biology migration algorithm: a new nature-inspired heuristic methodology for global optimization. *Soft Computing*. [online] Available at: <<https://doi.org/10.1007/s00500-018-3381-9>>.

Zhang, X., Chen, X. and He, Z., 2010. An ACO-based algorithm for parameter optimization of support vector machines. *Expert Systems with Applications*, 37(9), pp.6618–6628.

Zhang, Y., Gong, D. and Zhang, J., 2013. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103, pp.172–185.

Zhao, R. and Tang, W., 2008. Monkey algorithm for global numerical optimization. *Journal of Uncertain Systems*, 2(3), pp.165–176.

Zhao, W., Wang, L. and Zhang, Z., 2019a. Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. *Neural Computing and Applications*. [online] Available at: <<https://doi.org/10.1007/s00521-019-04452-x>>.

Zhao, W., Wang, L. and Zhang, Z., 2019b. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*, 163, pp.283–304.

Zhaohui, C. and Haiyan, T., 2011. Cockroach swarm optimization for vehicle routing problems. *Energy Procedia*, 13(Complete), pp.30–35.

Zheng, Y.-J., 2015. Water wave optimization: A new nature-inspired metaheuristic. *Computers & Operations Research*, 55, pp.1–11.

Zong Woo Geem, Joong Hoon Kim and Loganathan, G.V., 2001. A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), pp.60–68.

Zou, Y., 2019. The whirlpool algorithm based on physical phenomenon for solving optimization problems. *Engineering Computations*, 36(2), pp.664–690.

Κωνσταντίνου, Χ., 2018. Εφαρμογή της Βελτιστοποίησης Εμπνευσμένης από τον Ηχοεντοπισμό (*Sonar Inspired Optimization*) στη Διαχείριση Οικονομικού Χαρτοφυλακίου. Διπλωματική Εργασία. Τμήμα Μηχανικών Οικονομίας και Διοίκησης, Πολυτεχνική Σχολή, Πανεπιστήμιο Αιγαίου.

Μελανίτη, Μ.-Α., 2013. Μελέτη ανάπτυξης δικτύου μεταφοράς ηλεκτρικής ενέργειας για την ενσωμάτωση ΑΠΕ. [Διπλωματική Εργασία] Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών, Πανεπιστημίου Πατρών. Available at: <<http://nemertes.lis.upatras.gr/jspui/handle/10889/7864>>.

Παράρτημα Ι

Υποκατηγορίες Αλγορίθμων Εμπνευσμένων από τη Φύση

Πίνακας I-1: Υποκατηγορίες μεθόδων Νοημοσύνης Σμήνους (ΝΣ)

Τροφοληψία		
(Αλγόριθμος) Μπαρμπουνιού με Κίτρινη Ράχη	Yellow Saddle Goatfish	(Zaldívar et al., 2018)
Αλγόριθμος Βελτιστοποίησης Αναζήτησης Πιγκουίνων	Penguins Search Optimization Algorithm	(Gheraibia and Moussaoui, 2013)
Αλγόριθμος Βελτιστοποίησης Κουρνιάσματος Κόρακα	Raven Roosting Optimization Algorithm	(Brabazon, Cui and O'Neill, 2016)
Αλγόριθμος Βελτιστοποίησης Πεταλούδας	Butterfly Optimization Algorithm	(Arora and Singh, 2019)
Αλγόριθμος Βελτιστοποίησης Σμήνους Δελφινιών	Dolphins Swarm Optimization Algorithm	(Yong et al., 2016)
Αλγόριθμος Βελτιστοποίησης Φαλαινών	Whale Optimization Algorithm	(Mirjalili and Lewis, 2016)
Αλγόριθμος Βρωμούσας	Eurygaster Algorithm	(Ahmadi, Salehi and Karimi, 2012)
Αλγόριθμος Νυχτερίδας	Bat Algorithm	(Yang, 2010)
Αλγόριθμος Σμήνους Πτηνών	Bird Swarm Algorithm	(Meng et al., 2016)
Αλγόριθμος Σμήνους Φαλαινών	Whale Swarm Algorithm	(Zeng, Gao and Li, 2017)
Αλγόριθμος του Αφρικανικού Αγριόσκυλου	African Wild Dog Algorithm	(Subramanian, Sekar and Subramanian, 2013)
Αλγόριθμος του Τυφλού, Γυμνού Τυφλοπόντικα	Blind, naked Mole-rat Algorithm	(Shirzadi and Bagheri, 2012)
Αλγόριθμος των Μελισσών	The Bees Algorithm	(Yuce et al., 2013)
Αλγόριθμος Φυλής Σουρικατών	Meerkat Clan Algorithm	(Al-Obaidi, Abdullah and Ahmed, 2018)
Αναζήτηση Κυνηγιού	Hunting Search	(Oftadeh, Mahjoob and Shariatpanahi, 2010)
Αναζήτηση Λύκου	Wolf Search	(Tang et al., 2012)
Βελτιστο-Μέλισσες	OptBees	(R. D. Maia, L. N. de Castro and W. M. Caminhas, 2012)
Βελτιστοποίηση Αποικίας Μελισσών	Bee Colony Optimization	(Teodorovic and Dell'Orco, 2005)
Βελτιστοποίηση Αποικίας Μυρμηγκιών	Ant Colony Optimization	(Dorigo, 1992)
Βελτιστοποίηση Σμήνους Κατσαρίδων	Cockroach Swarm Optimization	(Zhaohui and Haiyan, 2011)

Βελτιστοποίηση Σμήνους Κότας	Chicken Swarm Optimization	(Meng et al., 2014)
Βελτιστοποίηση Σμήνους Μελισσών	Bee Swarm Optimization	(Akbari, Mohammadi and Ziarati, 2010)
Βελτιστοποίηση της Πασχαλίτσας	Seven-spot Ladybird Optimization	(Wang, Zhu and Huang, 2013)
Βελτιστοποίηση Τροφοληψίας Βακτηρίων	Bacterial Foraging Optimization	(Passino, 2002)
Βελτιστοποιητής Γκριζού Λύκου	Grey Wolf Optimizer	(Mirjalili, Mirjalili and Lewis, 2014)
Βελτιστοποιητής Ιστιοφόρου Ξιφία	Sailfish Optimizer	(Shadravan, Naji and Bardsiri, 2019)
Βελτιστοποιητής Ύαινας με Βούλες	Spotted Hyena Optimizer	(Dhiman and Kumar, 2017)
Εξέλιξη Σμήνους Κατσαρίδων	Cockroach Swarm Evolution	(Wu and Wu, 2015)
Κυψέλη Μελισσών	BeeHive	(Wedde, Farooq and Zhang, 2004)
Τεχνητή Αποικία Μελισσών	Artificial Bee Colony	(Karaboga and Basturk, 2007)

Κοινωνική Συμπεριφορά

Αλγόριθμος Αγέλης Λύκων	Wolf Pack Algorithm	(Wu and Zhang, 2014)
Αλγόριθμος Αναζήτησης Ελέφαντα	Elephant Search Algorithm	(S. Deb, S. Fong and Z. Tian, 2015)
Αλγόριθμος Βελτιστοποίησης Γαιοσκώληκα	Earthworm Optimization Algorithm	(Wang, Deb and Coelho, 2015b)
Αλγόριθμος Βελτιστοποίησης Κογιότ	Coyote Optimization Algorithm	(J. Pierezan and L. Dos Santos Coelho, 2018)
Αλγόριθμος Βελτιστοποίησης Λέοντα	Lion Optimization Algorithm	(Yazdani and Jolai, 2016)
Αλγόριθμος Βελτιστοποίησης Σμήνους Χελιδονιών	Swallow Swarm Optimization Algorithm	(Neshat, Sepidnam and Sargolzaei, 2013)
Αλγόριθμος Βελτιστοποίησης του Πιθήκου Αράχνης	Spider Monkey Optimization Algorithm	(Bansal et al., 2014)
Αλγόριθμος Βομβίνων	Bumblebees	(Comellas and Martinez-Navarro, 2009)
Αλγόριθμος Επιδρομής Κατσαρίδων	Roach Infestation Optimization	(T. C. Havens et al., 2008)
Αλγόριθμος Πυγολαμπίδας	Firefly Algorithm	(Yang, 2009)
Αλγόριθμος Σμήνους Δελφινιών	Dolphin Swarm Algorithm	(Wu, Yao and Yang, 2016)

Αλγόριθμος Σμήνους Ψαριών	Fish-swarm Algorithm	(Li, Shao and Qian, 2002)
Αναζήτηση Κούκου	Cuckoo Search	(X. Yang and Suash Deb, 2009)
Αποικία Αυτοκρατορικών Πιγκουίνων	Emperor Penguins Colony	(Harifi et al., 2019)
Βελτιστοποίηση Αποικίας Τερμιτών	Termite Colony Optimization	(Hedayatzadeh et al., 2010)
Βελτιστοποίηση Αφρικανικού Βούβαλου	African Buffalo Optimization	(Odili and Kahar, 2015)
Βελτιστοποίηση Κλωσόπουλων Πέρδικας See-See	See-See Partridge Chicks Optimization	(R. Omidvar, H. Parvin and F. Rad, 2015)
Βελτιστοποίηση Κοινωνικής Αράχνης	Social Spider Optimization	(Cuevas et al., 2013)
Βελτιστοποίηση Κοπαδιού Ελεφάντων	Elephant Herding Optimization	(Wang, Deb and Coelho, 2015a)
Βελτιστοποίηση Μαύρης Χήρας	Black Widow Optimization	(Hayyolalam and Pourhaji Kazem, 2020)
Βελτιστοποίηση Σμήνους Φωσφορίζοντος Σκόκληκα	Glowworm Swarm Optimization	(Krishnanand and Ghose, 2006)
Βελτιστοποίηση Συντρόφου Δελφινιού	Dolphin Partner Optimization	(Y. Shiqin, J. Jianjun and Y. Guangxing, 2009)
Βελτιστοποιητής Αυτοκρατορικού Πιγκουίνου	Emperor Penguin Optimizer	(Dhiman and Kumar, 2018)
Κοπάδι Κριλ	Krill Herd	(Gandomi and Alavi, 2012)
Μπλε Πίθηκος	Blue Monkey	(Mahmood and Al-Khateeb, 2019)
Σμήνος Ακρίδων	Locust Swarms	(Chen, 2009a)
Άλλη Συμπεριφορά Σμήνους		
Αλγόριθμος Αβαρούς Σμήνους	Weightless Swarm Algorithm	(Ting et al., 2012)
Αλγόριθμος Βελτιστοποίησης Ακρίδας	Grasshopper Optimization Algorithm	(Mirjalili et al., 2018)
Αλγόριθμος Βελτιστοποίησης Περρηφάνιας Λέοντα	Lion Pride Optimization Algorithm	(Kaveh and Mahjoubi, 2018)
Αλγόριθμος Βελτιστοποίησης του Γλάρου	Seagull Optimization Algorithm	(Dhiman and Kumar, 2019)
Αλγόριθμος Βιολογικής Μετανάστευσης	Biology Migration Algorithm	(Zhang et al., 2018)
Αλγόριθμος εμπνευσμένος από Σουρικάτες	Meerkat-inspired Algorithm	(Klein and dos Santos Coelho, 2018)

Αλγόριθμος Λιβελούλας	Dragonfly Algorithm	(Mirjalili, 2016a)
Αλγόριθμος Σμήνους Σάλπας	Salp Swarm Algorithm	(Mirjalili et al., 2017)
Αλγόριθμος Συρροής Βακτηρίων	Bacterial Swarming Algorithm	(Tang, Wu and Saunders, 2007)
Αλγόριθμος Τεχνητής Αναζήτησης Σμήνους	Artificial Searching Swarm Algorithm	(Chen, 2009b)
Αλγόριθμος του Βίσωνα	Bison	(Kazikova et al., 2019)
Αλγόριθμος του Μοντέλου Κληρονομικότητας Κοπαδιού Προβάτων	Sheep Flocks Heredity Model Algorithm	(Hyunchul Kim and Byungchul Ahn, 2001)
Αναζήτηση Κοπαδιού Ψαριών	Fish School Search	(Filho et al., 2009)
Αναζήτηση Στοχαστικής Διάχυσης	Stochastic Diffusion Search	(J. M. Bishop, 1989)
Βελτιστοποίηση Αλμάτων Βατράχων	Jumping Frogs Optimization	(Garcia and Pérez, 2008)
Βελτιστοποίηση Αποικίας Κατσαρίδων	Cockroach Colony Optimization	(Cheng et al., 2015)
Βελτιστοποίηση Εμπνευσμένη από τα Περιστερία	Pigeon-inspired Optimization	(Qiao and Duan, 2014)
Βελτιστοποίηση Ζευγαρώματος Βομβίνων	Bumble Bees Mating Optimization	(Marinakis and Marinaki, 2011)
Βελτιστοποίηση Κοπαδιού Γκνου	Wildebeests Herd Optimization	(Motevali et al., 2018)
Βελτιστοποίηση Μετανάστευσης Ζώων	Animal Migration Optimization	(Li, Zhang and Yin, 2014)
Βελτιστοποίηση Μετανάστευσης Πτηνών	Migrating Birds Optimization	(Duman, Uysal and Alkaya, 2011)
Βελτιστοποίηση Νυχτοπεταλούδας και Φλόγας	Moth Flame Optimization	(Mirjalili, 2015a)
Βελτιστοποίηση Σμήνους Σωματιδίων	Particle Swarm Optimization	(Eberhart and Kennedy, 1995)
Βελτιστοποίηση της Πεταλούδας Μονάρχη	Monarch Butterfly Optimization	(Wang, Deb and Cui, 2015)
Βελτιστοποίηση Τραγουδιού της Μεγάπτερης Φάλαινας	Vocalization of Humpback Whale Optimization Algorithm	(Masadeh, Sharieh and Mahafzah, 2019)
Βελτιστοποιητής Εγωιστικού Κοπαδιού	Selfish Herd Optimizer	(Fausto et al., 2017)
Βελτιστοποιητής Εξαρτώμενος από την Καταλληλότητα	Fitness Dependent Optimizer	(J. M. Abdullah and T. Ahmed, 2019)
Βελτιστοποιητής Στρατιωτικού Σκύλου	Military Dog Optimizer	(Tripathi, Sharma and Bala, 2019)
Ιεραρχικό Μοντέλο Σμήνους	Hierarchical Swarm Model	(Chen et al., 2010)

Συλλογική Συμπεριφορά Ζώων

Collective Animal Behavior

(Cuevas, Fausto and González, 2020)

Τεχνητές Κοινωνικές Κατσαρίδες

Artificial Social Cockroaches

(Bouarara, Hamou and Abdelmalek, 2018)

Πίνακας I-2: Υποκατηγορίες μεθόδων Νοημοσύνης Εμπνευσμένης από Οργανισμούς (NEO)

Πανίδα		
Αλγόριθμο Ηχοεντοπισμού Νυχτερίδων	Bat Sonar Algorithm	(Tawfeeq, 2012)
Αλγόριθμος Αναζήτησης Κεραίων Σκαθαριού	Beetle Antennae Search Algorithm	(Jiang and Li, 2017)
Αλγόριθμος Αναζήτησης Κόρακα	Crow Search Algorithm	(Askarzadeh, 2016)
Αλγόριθμος Αναζήτησης Κουκουβάγιας	Owl Search Algorithm	(Jain et al., 2018)
Αλγόριθμος Αναζήτησης Ρινόκερου	Rhinoceros Search Algorithm	(Deb et al., 2018)
Αλγόριθμος Αναζήτησης Σκίουρου	Squirrel Search Algorithm	(Jain, Singh and Rani, 2019)
Αλγόριθμος Ανακατεμένου Άλματος Βατράχων	Shuffled Frog Leaping Algorithm	(Eusuff, Lansey and Pasha, 2006)
Αλγόριθμος Βελτιστοποίησης Αιγυπτιακού Γύπα	Egyptian Vulture Optimization Algorithm	(Sur, Sharma and Shukla, 2013)
Αλγόριθμος Βελτιστοποίησης Βασισμένος στον Γατόπαρδο	Cheetah Based Optimization Algorithm	(Klein, Mariani and Coelho, 2018)
Αλγόριθμος Βελτιστοποίησης Γαϊδάρου και Λαθρέμπορου	Donkey and Smuggler Optimization Algorithm	(Shamsaldin et al., 2019)
Αλγόριθμος Βελτιστοποίησης Γερακιού	Falcon Optimization Algorithm	(Vasconcelos Segundo, Mariani and Santos Coelho, 2019)
Αλγόριθμος Βελτιστοποίησης Επιδημικότητας	Virulence Optimization Algorithm	(Jaderyan and Khotanlou, 2016)
Αλγόριθμος Βελτιστοποίησης Θαλάσσιου Λέοντα	Sea Lion Optimization Algorithm	(Masadeh, 2019)
Αλγόριθμος Βελτιστοποίησης Καπνογλάρου	Sooty Tern Optimization Algorithm	(Dhiman and Kaur, 2019)
Αλγόριθμος Βελτιστοποίησης Κοραλλιογενούς Ύφαλου	Coral Reefs Optimization Algorithm	(Salcedo-Sanz et al., 2014)
Αλγόριθμος Βελτιστοποίησης Μαγνητοτακτικών Βακτηριών	Magnetotactic Bacteria Optimization Algorithm	(H. Mo and L. Xu, 2013)
Αλγόριθμος Βελτιστοποίησης Πολικής Αρκούδας	Polar Bear Optimization Algorithm	(Połap and Wozniak, 2017)
Αλγόριθμος Βελτιστοποίησης Πράσινου Ερωδιού	Green Herons Optimization Algorithm	(Sur and Shukla, 2013)
Αλγόριθμος Βελτιστοποίησης Ρακούν	Raccoon Optimization Algorithm	(S. Zangbari Koohi et al., 2019)
Αλγόριθμος Βελτιστοποίησης Σμήνους Σκαθαριών	Beetle Swarm Optimization Algorithm	(Wang, Yang and Liu, 2018)

Αλγοριθμος Θύτη-Θηράματος	Prey-Predator Algorithm	(Tilahun and Ong, 2013)
Αλγόριθμος Ιαγουάρου	Jaguar Algorithm	(Chen et al., 2015)
Αλγόριθμος Καμήλας	Camel Algorithm	(Ibrahim and Ali, 2016)
Αλγόριθμος Κόκκινου Ελαφιού	Red Deer Algorithm	(Fard and Hajiaghaei-Keshteli, 2016)
Αλγόριθμος Κότας που Γεννάει Αυγά	Laying Chicken Algorithm	(Hosseini, 2017)
Αλγόριθμος Κυκλικής Παρθενογέννησης	Cyclical Parthenogenesis Algorithm	(Kaveh and Zolghadr, 2017)
Αλγόριθμος Πάπιας Keshtel	Keshtel Algorithm	(Hajiaghaei-Keshteli and Aminnayeri, 2013)
Αλγόριθμος Πιθήκου	Monkey Algorithm	(Zhao and Tang, 2008)
Αλγόριθμος Σουπιάς	Cuttlefish Algorithm	(Eesa, Brifcani and Orman, 2013)
Αλγόριθμος Συλλογής Γύρης από Μέλισσες	Bee Collecting Pollen Algorithm	(Lu and Zhou, 2008)
Αλγόριθμος Τεχνητών Φυκιών	Artificial Algae Algorithm	(Uymaz, Tezel and Yel, 2015)
Αλγόριθμος της Πυγολαμπίδας Άλτη	Jumper Firefly Algorithm	(M. Bidar and H. Rashidy Kanan, 2013)
Αλγόριθμος του Κόνδορα των Άνδεων	Andean Condor Algorithm	(Almonacid and Soto, 2019)
Αλγόριθμος του Λιονταριού	The Lion's Algorithm	(Rajakumar, 2012)
Αλγόριθμος του Σκαθαριού Pity	Pity Beetle Algorithm	(Kallioras, Lagaros and Avtzis, 2018)
Αλγόριθμος του Υπερ-βακτηρίου	Superbug Algorithm	(Anandaraman, Madurai Sankar and Natarajan, 2012)
Αλγόριθμος Φάλαινας Δολοφόνου	Killer Whale Algorithm	(Biyanto et al., 2017)
Αλγόριθμος Φάλαινας Φυσητήρα	Sperm Whale Algorithm	(Ebrahimi and Khamehchi, 2016)
Αλγόριθμος Ψαριού που αναπαράγεται από το Στόμα	Mouth Brooding Fish algorithm	(Jahani and Chizari, 2018)
Αναζήτηση Αποικίας Ιών	Virus Colony Search	(Li et al., 2016b)
Αναζήτηση Καραφλού Αετού	Bald Eagle Search	(Alsattar, Zaidan and Zaidan, 2019)
Αναζήτηση Πιθήκων	Monkey Search	(Mucherino and Seref, 2007)
Αναζήτηση Συμβιωτικών Οργανισμών	Symbiotic Organisms Search	(Cheng and Prayogo, 2014)

Ανταγωνισμός στον Πόρο	Competition over Resources	(S. Mohseni et al., 2014)
Βελτιστοποίηση Βασιμένη στο Μοντέλο της Γρίπης των Χοίρων	Swine Influenza Models Based Optimization	(Pattnaik et al., 2013)
Βελτιστοποίηση Βιο-γεωγραφίας	Biogeography-based Optimization	(Simon, 2008)
Βελτιστοποίηση Γερακιών Χάρις	Harris Hawks Optimization	(Heidari et al., 2019)
Βελτιστοποίηση Δεξαμενής Δελφινιών	Dolphin Pod Optimization	(Serani and Diez, 2018)
Βελτιστοποίηση εντοπισμού μέσω ηλεκτρικών πεδίων των Ψαριών	Fish Electrolocation Optimization	(Haldar and Chakraborty, 2017)
Βελτιστοποίηση Ζευγαρώματος Μελισσών	Honey-bees Mating Optimization	(Haddad, Afshar and Mariño, 2006)
Βελτιστοποίηση Περιπλάνησης Μυδιών	Mussels Wandering Optimization	(An et al., 2013)
Βελτιστοποίηση Πετάγματος Κουνουπιού	Mosquito Flying Optimization	(M. Alauddin, 2016)
Βελτιστοποίηση Σμήνους Pontogammarus Maeoticus	Pontogammarus Maeoticus Swarm Optimization	(Ghojogh and Sharifian, 2018)
Βελτιστοποίηση Σμήνους Γάτας	Cat Swarm Optimization	(Chu, Tsai and Pan, 2006)
Βελτιστοποίηση Τεχνητής Πεταλούδας	Artificial Butterfly Optimization	(Qi, Zhu and Zhang, 2017)
Βελτιστοποίηση Φρουτόμυγας	Fruit Fly Optimization	(Pan, 2011)
Βελτιστοποίηση Οσφρησης Καρχαρία	Shark Smell Optimization	(Abedinia, Amjady and Ghasemi, 2016)
Βελτιστοποιητή Σατέν Πτιλονορρυγίδιας	Satin Bowerbird Optimizer	(Samareh Moosavi and Khatibi Bardsiri, 2017)
Βελτιστοποιητής Αναζήτησης Ομάδας	Group Search Optimizer	(S. He, Q. H. Wu and J. R. Saunders, 2009)
Βελτιστοποιητής Διαπεραστικού Αετού	Eagle Piercing Optimizer	(Khan et al., 2018)
Βελτιστοποιητής Ζευγαρώματος Θυσανόποδων	Barnacles Mating Optimizer	(Sulaiman et al., 2018)
Βελτιστοποιητής Ζευγαρώματος Πουλιών	Bird Mating Optimizer	(Askarzadeh, 2014)
Βελτιστοποιητής Ιπτάμενου Σκίουρου	Flying Squirrel Optimizer	(Azizyan et al., 2019)
Βελτιστοποιητής Μηρυλεοντίδας	Ant Lion Optimizer	(Mirjalili, 2015b)
Βελτιστοποιητής Περηφάνιας Λέοντα	Lion Pride Optimizer	(Wang, Jin and Cheng, 2012)
Γάμος Μελισσών	Marriage in Honey Bees	(Abbass, 2001)

Διαδρομή του μεγάλου Σολομού	Great Salmon Run	(Mozaffari, Fathi and Behzadipour, 2012)
Ευρετική Βελτιστοποίηση Τσαλαπετεινού	Hoopoe Heuristic Optimization	(El-Dosuky et al., 2012)
Ηχοεντοπισμός Δελφινιών	Dolphin Echolocation	(Kaveh and Farhoudi, 2013)
Ιαπωνικοί Δενδροβάτραχοι	Japanese tree frogs	(Hernández and Blum, 2011)
Ιογενή Συστήματα	Viral Systems	(Cortés et al., 2008)
Κυκλικές Δομές Τετραοδοντίδων	Circular Structures of Puffer Fish	(M. C. Catalbas and A. Gulten, 2018)
Τεχνητό Τάισμα Πτηνών	Artificial Feeding Birds	(Lamy, 2019)
Χημειοταξία Βακτηρίων	Bacterial Chemotaxis	(Muller et al., 2002)
Ωοτοκία Κουνουπιών	Mosquitos Oviposition	(Minhas and Arif, 2011)
Χλωρίδα		
Αλγόριθμο Ανάπτυξης Ριζών	Root Growth Algorithm	(Zhang, Zhu and Chen, 2014)
Αλγόριθμο Δενδρυλλίων που Αναπτύσσονται	Saplings Growing up Algorithm	(Karci and Alatas, 2006)
Αλγόριθμος Αναπαραγωγής Φυτών που αναπτύσσονται από Σπόρους	Seed-based Plant Propagation Algorithm	(Sulaiman and Salhi, 2015)
Αλγόριθμος Ανάπτυξης Δέντρων	Tree Growth Algorithm	(Cheraghalipour, Hajiaghaei-Keshteli and Paydar, 2018)
Αλγόριθμος Αναρρόφησης Οτρικουλάριων	BladderWorts Suction Algorithm	(R. Gowri and R. Rathipriya, 2018)
Αλγόριθμος Βελτιστοποίησης Δάσους	Forest Optimization Algorithm	(Ghaemi and Feizi-Derakhshi, 2014)
Αλγόριθμος Βελτιστοποίησης Ενέργειας Φύσαρου	Physarum-energy Optimization Algorithm	(Feng et al., 2019)
Αλγόριθμος Βελτιστοποίησης Ριζών Δέντρου	Root Tree Optimization Algorithm	(Labbi et al., 2016)
Αλγόριθμος Βελτιστοποίησης Τεχνητού Φυτού	Artificial Plant Optimization Algorithm	(Z. Zhao et al., 2011)
Αλγόριθμος Γονιμότητας Χωραφιών	Farmland Fertility Algorithm	(Shayanfar and Gharehchopogh, 2018)
Αλγόριθμος εμπνευσμένος από την Οικολογία	Eco-inspired Evolutionary Algorithm	(R. S. Parpinelli and H. S. Lopes, 2011)
Αλγόριθμος Επικονίασης Λουλουδιών	Flower Pollination Algorithm	(Yang, 2012a)

Αλγόριθμος Ορυζώνα	Paddy Field Algorithm	(Premaratne, Samarabandu and Sidhu, 2009)
Αλγόριθμος Παρακλαδιών Ρίζας	The Runner-root Algorithm	(Merrikh-Bayat, 2015)
Αλγόριθμος Πολλαπλασιασμού Φυτών	Plant Propagation Algorithm	(Salhi and Fraga, 2011)
Αλγόριθμος Σπόρου Δέντρου	Tree Seed Algorithm	(Kiran, 2015)
Αλγόριθμος Υδροχαρούς Ζιζανίου	Waterweeds Algorithm	(Cheng et al., 2016)
Βελτιστοποίηση Αναπαραγωγής Μανιταριών	Mushroom Reproduction Optimization	(M. Bidar et al., 2018)
Βελτιστοποίηση Ανάπτυξης Φυτών	Plant Growth Optimization	(W. Cai, W. Yang and X. Chen, 2008)
Βελτιστοποίηση Βασισμένη στα Δέντρα	Tree Based Optimization	(Ghojogh, Sharifian and Mohammadzade, 2018)
Βελτιστοποίηση Μάζας Ρίζας	Root Mass Optimization	(Qi et al., 2013)
Βελτιστοποίηση Φύσαρου	Physarum Optimization	(Song et al., 2012)
Βελτιστοποίηση Χωροκατακτητικού Ζιζανίου	Invasive Weed Optimization	(Mehrabian and Lucas, 2006)
Φυσική Ανάπλαση Δάσους	Natural Forest Regeneration	(Moez, Kaveh and Taghizadieh, 2016)
Λοιπά		
Αλγόριθμος Βελτιστοποίησης Βλαστοκύτταρων	Stem Cells Optimization Algorithm	(Taherdangkoo, Yazdi and Bagheri, 2012)
Αλγόριθμος εμπνευσμένος από το Συκώτι	Kidney-inspired Algorithm	(Jaddi, Alvankarian and Abdullah, 2017)
Αλγόριθμος Επιλογής Κλώνου	Clonal Selection Algorithm	(de Castro and von Zuben, 2000)
Αλγόριθμος Επιστήμης Χημειοθεραπείας	Chemotherapy Science Algorithm	(Salmani and Eshghi, 2017)
Αλγόριθμος της Καρδιάς	Heart	(Hatamlou, 2014)
Αλγόριθμος της Μετενσάρκωσης	Reincarnation Algorithm	(A. Sharma, 2010)
Αλλοστατική Βελτιστοποίηση	Allostatic Optimization	(Osuna-Enciso et al., 2016)
Βελτιστοποίηση βασισμένη σε Τεχνητό Οικοσύστημα	Artificial Ecosystem-based Optimization	(Zhao, Wang and Zhang, 2019a)
Βελτιστοποίηση Επιβίωσης Ζόμπι	Zombie Survival Optimization	(Nguyen and Bhanu, 2012)

Βελτιστοποίηση Επιθετικής Ανάπτυξης Όγκου	Invasive Tumor Growth Optimization	(Tang et al., 2015)
Βελτιστοποίηση Τεχνητής Μεταδοτικής Ασθένειας	Artificial Infectious Disease Optimization	(Huang, 2016)
Γενετικός Αλγόριθμος	Genetic Algorithm	(Holland, 1975)
Επικοινωνία Νευρώνων	Neuronal Communication	(Asil Gharebaghi and Ardalan Asl, 2017)
Τεχνητό Σύστημα Στεφανιαίας Κυκλοφορίας	Artificial Coronary Circulation System	(Kaveh and Kooshkebaghi, 2019)

Πίνακας I-3: Υποκατηγορίες μεθόδων Νοημοσύνης Εμπνευσμένης από Φυσικά Φαινόμενα ή Νόμους των Επιστημών (ΝΕΦΦΝΕ)

Φαινόμενα του Σύμπαντος		
Αλγόριθμος Αναζήτησης βασισμένος στον Γαλαξία	Galaxy-based Search Algorithm	(Shah-Hosseini, 2011)
Αλγόριθμος Βαρύτητας του Διαστήματος	Space Gravitational algorithm	(Hsiao et al., 2005)
Αλγόριθμος Βαρυτικής Αναζήτησης	Gravitational Search Algorithm	(Rashedi, Nezamabadi-Pour and Saryazdi, 2009)
Βελτιστοποίηση Βαρυτικών Διαδράσεων	Gravitational Interactions Optimization	(Flores, López and Barrera, 2011)
Βελτιστοποίηση Διαστημικής Βαρύτητας	Space Gravity Optimization	(Kılınc, Mahouti and Güneş, 2013)
Βελτιστοποιητής Πολυ-σύμπαντος	Multi-Verse Optimizer	(Mirjalili, Mirjalili and Hatamlou, 2016)
Βελτιστοποιητής Υπερκαινοφανούς Αστέρα	Supernova Optimizer	(Hudaib and Fakhouri, 2017)
Μαύρη Τρύπα	Black Hole	(Hatamlou, 2013)
Μεγάλη Έκρηξη – Μεγάλη Σύνθλιψη	Big Bang – Big Crunch	(Erol and Eksin, 2006)
Νόμοι Θετικών Επιστημών		
Αλγόριθμος Αναζήτησης βάσει της Γενικής Θεωρίας της Σχετικότητας	General Relativity Search Algorithm	(Beiranvand and Rokrok, 2015)
Αλγόριθμος Αναζήτησης Ελατηρίου	Spring Search Algorithm	(M. Deghani et al., 2017)
Αλγόριθμος βασισμένος στο Ισοζύγιο μεταξύ αέριας και υγρής μορφής	Vapour-liquid Equilibrium-based Algorithm	(Cortés-Toro et al., 2018)
Αλγόριθμος βασισμένος στους νόμους του Φράνκλιν και του Κουλόμπ	Coulomb's Franklin's Algorithm	(Ghasemi et al., 2018)
Αλγόριθμος Βελτιστοποίησης βασισμένος στον Νόμο του Ωμ	Ohm's Law Optimization Algorithm	(Fadafen et al., 2019)
Αλγόριθμος Δυναμικής Σπειροειδούς κίνησης	Spiral Dynamics Algorithm	(Nasir et al., 2013)
Αλγόριθμος Έκρηξης Νάρκης	Mine Blast algorithm	(Sadollah et al., 2012)
Αλγόριθμος Εξελικτικών Κέντρων	Evolutionary Centers Algorithm	(Mejía-de-Dios and Mezura-Montes, 2019)
Αλγόριθμος Ημίτονου Συνημίτονου	Sine Cosine Algorithm	(Mirjalili, 2016b)

Αλγόριθμος Καθεστώτος Ροής	Flow Regime Algorithm	(Tahani and Babayan, 2018)
Αλγόριθμος Κίνησης Ιόντων	Ions Motion Algorithm	(Javidy, Hatamlou and Mirjalili, 2015)
Αλγόριθμος Μετακίνησης Μορίων Ιδανικών Αερίων	Ideal Gas Molecular Movement Algorithm	(Varaee and Ghasemi, 2017)
Αλγόριθμος Σπειροειδούς Δυναμικής	Spiral Dynamics Algorithm	(Nasir et al., 2013)
Αλγόριθμος Σύγκρουσης Σωματιδίων	Particle Collision Algorithm	(Sacco and Oliveira, 2005)
Αλγόριθμος Συστήματος Παλλόμενων Σωματιδίων	Vibrating Particles System Algorithm	(Kaveh and Ilchi Ghazaan, 2017)
Αλγόριθμος Τεχνητού Ηλεκτρικού Πεδίου	Artificial Electric Field Algorithm	(Yadav, 2019)
Αναζήτηση Αρμονίας	Harmony Search	(Zong Woo Geem, Joong Hoon Kim and Loganathan, 2001)
Αναζήτηση Μεταφοράς Θερμότητας	Heat Transfer Search	(Patel and Savsani, 2015)
Αναζήτηση Στοχαστικών Φράκταλς	Stochastic Fractal Search	(Salimi, 2015)
Αναζήτηση Φορτισμένου Συστήματος	Charged System Search	(Kaveh and Talatahari, 2010)
Βελτιστοποίηση Ακτίνων	Ray Optimization	(Kaveh and Khayatizad, 2012)
Βελτιστοποίηση Αναζήτησης Ατόμου	Atom Search Optimization	(Zhao, Wang and Zhang, 2019b)
Βελτιστοποίηση Ανταλλαγής Θερμότητας	Thermal Exchange Optimization	(Kaveh and Dadras, 2017)
Βελτιστοποίηση βάσει του φαινομένου της Κεντρομόλου Δυνάμεως	Central Force Optimization	(Formato, 2007)
Βελτιστοποίηση βάσει του φαινομένου του Ηλεκτρομαγνητισμού	Electromagnetism Optimization	(Birbil and Fang, 2003)
Βελτιστοποίηση βασισμένη στην Υστέρηση	Hysteretic Optimization	(Zarand et al., 2002)
Βελτιστοποίηση βασισμένη στον Ηχοεντοπισμό	Sonar Inspired Optimization	(Tzanetos and Dounias, 2017a)
Βελτιστοποίηση Διαλυτότητας Αερίων σύμφωνα με τον νόμο του Χένρυ	Henry Gas Solubility Optimization	(Hashim et al., 2019)
Βελτιστοποίηση Έλκτικών Δυνάμεων	Attraction Force Optimization	(Bendato et al., 2016)
Βελτιστοποίηση εμπνευσμένη από τις Χημικές Αντιδράσεις	Chemical-reaction-inspired Optimization	(Lam and Li, 2010)
Βελτιστοποίηση Εμπνευσμένη από την Οπτική	Optics Inspired Optimization	(Husseinzadeh Kashan, 2015)
Βελτιστοποίηση Ηλεκτρομαγνητικού Πεδίου	Electromagnetic Field Optimization	(Abedinpourshotorban et al., 2016)

Βελτιστοποίηση Κίνησης των Αερίων κατά τον Robert Brown	Gases Brownian Motion Optimization	(Abdechiri, Meybodi and Bahrami, 2013)
Βελτιστοποίηση Μορίων Αερίων βάσει των νόμων της Κινητικής	Kinetic Gas Molecule Optimization	(Moein and Logeswaran, 2014)
Βελτιστοποίηση Τεχνητής Φυσικής	Artificial Physics Optimization	(Xie, Zeng and Cui, 2009)
Βελτιστοποίησης βασισμένη στη Σύγκρουση Σωμάτων	Colliding Bodies optimization	(Kaveh and Mahdavi, 2014)
Καταστάσεις της Ύλης	States of Matter	(Cuevas, Echavarría and Ramírez-Ortegón, 2014)
Μέθοδος Έκρηξης Χειροβομβίδας	Grenade Explosion Method	(Ahrari and Atai, 2010)
Φυσικά Φαινόμενα		
Αλγόριθμος Αναζήτησης Δίνης	Vortex Search Algorithm	(Doğan and Ölmez, 2015)
Αλγόριθμος Αναζήτησης Κεραυνών	Lightning Search Algorithm	(Shareef, Ibrahim and Mutlag, 2015)
Αλγόριθμος Βελτιστοποίησης Βροχόπτωσης	Rainfall Optimization Algorithm	(Aghay Kaboli, Selvaraj and Rahim, 2017)
Αλγόριθμος Βελτιστοποίησης Κρυσταλλικής Ενέργειας	Crystal Energy Optimization Algorithm	(Feng, Ma and Yu, 2016)
Αλγόριθμος Ηλεκτροστατικής Εκφόρτισης	Electrostatic Discharge Algorithm	(H. R. E. H. Boucekara, 2019)
Αλγόριθμος Κύκλου του Νερού	Water Cycle Algorithm	(Eskandar et al., 2012)
Αλγόριθμος Κύκλου του Νερού	Hydrological Cycle Algorithm	(Wedyan, Whalley and Narayanan, 2017)
Αλγόριθμος Τεχνητής Στάλας	Artificial Raindrop Algorithm	(Jiang et al., 2014)
Αλγόριθμος Τεχνητού Καταιωνιστήρα Ύδατος	Artificial Showering Algorithm	(Ali et al., 2015)
Αλγόριθμος Υδάτινης Δίνης	Whirlpool Algorithm	(Zou, 2019)
Βελτιστοποίηση βασισμένη στον Τυφώνα	Hurricane-based Optimization	(Rbough and Imrani, 2014)
Βελτιστοποίηση Διαδικασίας Σύνδεσης Κεραυνών	Lightning Attachment Procedure Optimization	(Nematollahi, Rahiminejad and Vahidi, 2017)
Βελτιστοποίηση Δυναμικής Ανέμων	Wind Driven Optimization	(Z. Bayraktar, M. Komurcu and D. H. Werner, 2010)
Βελτιστοποίηση Εξάτμισης Υδάτων	Water Evaporation Optimization	(Kaveh and Bakhshpoori, 2016)
Βελτιστοποίηση Μοντέλου Ατμοσφαιρικών Σύννεφων	Atmosphere Clouds Model Optimization	(Yan and Hao, 2013)

Βελτιστοποίηση Πυρηνικής Αντίδρασης	Nuclear Reaction Optimization	(Z. Wei et al., 2019)
Βελτιστοποίηση Υδάτινων Κυμάτων	Water Wave Optimization	(Zheng, 2015)
Δυναμική Σχηματισμού Ποταμών	River Formation Dynamics	(Rabanal, Rodríguez and Rubio, 2007)
Κυκλικά Υδάτινα Κύματα	Circular Water Waves	(Colak and Varol, 2015)
Νοήμον Σταγόνα Νερού	Intelligent Water Drop	(Shah-Hosseini, 2009)
Προσομοιωμένη Ανόπτηση	Simulated Annealing	(Kirkpatrick, Gelatt and Vecchi, 1983)

Παράρτημα II
Συγκεντρωτικοί Πίνακες
Εφαρμογών Αλγορίθμων
Εμπνευσμένων από τη Φύση

Πίνακας Π-1: Πλήθος εργασιών αλγορίθμων Νοσημοσύνης Σμήνους ανά Χώρο Εφαρμογής

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Βελτιστοποίηση Αφρικανικού Βούβαλου	-	-	6 (5)	3 (2)	1 (1)
Αλγόριθμο του Αφρικανικού Αγριόσκυλου	1 (1)	-	-	-	-
Βελτιστοποίηση Μετανάστευσης Ζώων	3 (2)	-	2 (1)	1 (0)	1 (1)
Αλγόριθμος Τεχνητής Αναζήτησης Σμήνους	3 (1)	-	-	1 (0)	-
Αλγόριθμος Συρροής Βακτηρίων	1 (0)	-	-	4 (1)	-
Βελτιστοποίηση Αποικίας Μελισσών	3 (3)	-	11 (10)	-	2 (2)
Βελτιστοποίηση Σμήνους Μελισσών	-	-	2 (1)	2 (2)	-
Κυψέλη Μελισσών	-	-	3 (3)	1 (0)	6 (2)
Αλγόριθμος Σμήνους Πτηνών	2 (1)	-	1 (1)	6 (2)	-
Αλγόριθμος Βελτιστοποίησης Πεταλούδας	1 (1)	-	1 (1)	-	-
Βελτιστοποίηση Ζευγαρώματος Βομβίνων	-	-	2 (2)	-	-
Βελτιστοποίηση Σμήνους Κότας	-	-	2 (1)	8 (4)	2 (1)
Βελτιστοποίηση Σμήνους Κατσαριδών	-	-	3 (3)	-	-
Αλγόριθμος Σμήνους Δελφινιών	-	-	-	1 (1)	-
Αλγόριθμος Λιβελούλας	5 (5)	-	5 (1)	28 (16)	4 (3)
Βελτιστοποίηση Κοπαδιού Ελεφάντων	1 (1)	-	5 (2)	8 (4)	1 (1)
Αλγόριθμος Αναζήτησης Ελέφαντα	-	-	2 (2)	-	1 (0)
Αλγόριθμος Σμήνους Ψαριών	23 (12)	1 (1)	42 (25)	21 (8)	9 (4)
Βελτιστοποίηση Σμήνους Φωσφορίζοντος Σκώληκα	-	-	5 (2)	5 (4)	1 (1)
Αναζήτηση Κυνηγιού	6 (4)	-	3 (2)	4 (1)	-
Κοπάδι Κριλ	12 (6)	1 (1)	3 (1)	10 (7)	1 (1)

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Αλγόριθμος Βελτιστοποίησης Λέοντα	1 (0)	-	2 (2)	1 (1)	-
Αλγόριθμος Φυλής Σουρικατών	-	-	1 (1)	-	-
Βελτιστοποίηση Μετανάστευσης Πτηνών	-	-	31 (20)	-	-
Βελτιστοποίηση Νυχτοπεταλούδας και Φλόγας	15 (11)	-	6 (5)	69 (48)	1 (1)
Βελτιστοποίηση της Πεταλούδας Μονάρχη	1 (0)	-	4 (3)	-	-
Βελτιστο-Μέλισσες	-	-	2 (1)	-	-
Αλγόριθμος Βελτιστοποίησης Αναζήτησης Πιγκουίνων	-	-	3 (1)	1 (1)	-
Βελτιστοποίηση εμπνευσμένη από τα Περιστερία	6 (4)	-	7 (5)	6 (5)	15 (12)
Αλγόριθμος Βελτιστοποίησης Κουρνιασματος Κόρακα	-	-	2 (2)	-	-
Βελτιστοποίηση Επιδρομής Κατσαρίδων	1 (0)	-	-	-	-
Βελτιστοποιητής Ιστιοφόρου Ξιφία	1 (1)	-	-	-	-
Αλγόριθμος Σμήνους Σάλπας	6 (3)	-	3 (1)	24 (18)	4 (3)
Βελτιστοποίηση της Πασχαλίτσας	-	-	-	1 (0)	-
Αλγόριθμος του Μοντέλου Κληρονομικότητας Κοπαδιού Προβάτων	1 (1)	-	7 (6)	-	-
Βελτιστοποίηση Κοινωνικής Αράχνης	7 (4)	-	8 (6)	16 (11)	1 (0)
Αλγόριθμος Βελτιστοποίησης του Πιθήκου Αράχνης	4 (3)	1 (1)	4 (4)	19 (13)	-
Βελτιστοποιητής Ύαινας με Βούλες	4 (3)	-	1 (0)	1 (1)	-
Αλγόριθμος Βελτιστοποίησης Σμήνους Χελιδονιών	-	-	1 (0)	-	-
Βελτιστοποίηση Αποικίας Τερμιτών	-	-	1 (0)	1 (1)	-
Αλγόριθμος των Μελισσών	32 (17)	-	8 (5)	5 (1)	6 (4)
Αλγόριθμος Αβαρούς Σμήνους	1 (1)	-	-	1 (0)	-
Αλγόριθμος Βελτιστοποίησης Φαλαινών	37 (30)	-	35 (26)	98 (69)	6 (5)

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Αλγόριθμος Σμήνους Φαλαινών	1 (1)	-	-	1 (1)	-
Αλγόριθμος Αγέλης Λύκων	1 (1)	-	6 (4)	1 (1)	-
Αναζήτηση Λύκου	-	-	-	2 (2)	-
Τροφοληψία	88 (62)	-	78 (58)	129 (83)	22 (14)
Κοινωνική Συμπεριφορά	50 (27)	3 (3)	85 (53)	85 (52)	16 (9)
Άλλη Συμπεριφορά Σμήνους	44 (29)	-	72 (47)	144 (96)	27 (22)

Πίνακας Π-2: Πλήθος εργασιών Αλγορίθμων Εμπνευσμένων από Οργανισμούς

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Βελτιστοποιητής Μηρυλεοντίδας	19 (13)	1 (1)	21 (15)	51 (35)	1 (1)
Αλγόριθμος Τεχνητών Φυκιών	1 (1)	-	2 (2)	4 (4)	-
Τεχνητό Σύστημα Στεφαναϊάς Κυκλοφορίας	2 (2)	-	-	-	-
Αλγόριθμος Βελτιστοποίησης Τεχνητού Φυτού	-	-	-	-	5 (5)
Χημειοταξία Βακτηρίων	1 (1)	-	-	6 (3)	1 (1)
Βελτιστοποιητήν Ζευγαρώματος Θυσανόποδων	-	-	-	1 (0)	-
Αλγόριθμος Αναζήτησης Κεραιών Σκαθαριού	2 (1)	-	-	-	3 (3)
Βελτιστοποιητής Ζευγαρώματος Πουλιών	1 (0)	-	2 (2)	3 (3)	-
Βελτιστοποίηση Σμήνους Γατών	2 (1)	-	17 (8)	15 (11)	4 (1)
Αλγόριθμος Βελτιστοποίησης Βασισμένος στον Γατόπαρδο	-	-	-	1 (1)	-
Αλγόριθμος Επιστήμης Χημειοθεραπείας	-	-	1 (1)	-	-
Ανταγωνισμόν στου Πόρους	-	-	-	5 (2)	-
Αλγόριθμο Βελτιστοποίησης Κοραλλιογενούς Ύφαλου	2 (2)	-	2 (2)	5 (5)	1 (1)
Αλγόριθμος Αναζήτησης Κόρακα	7 (3)	-	8 (5)	35 (17)	2 (1)
Αλγόριθμος Σουπιάς	1 (1)	-	-	-	1 (1)
Ηχοεντοπισμός Δελφινιών	15 (13)	-	-	2 (1)	-
Βελτιστοποίηση Δεξαμενής Δελφινιών	1 (0)	-	-	-	-
Αλγόριθμος Βελτιστοποίησης Αιγυπτιακού Γύπα	2 (2)	-	3 (3)	1 (1)	-
Βελτιστοποίηση εντοπισμού μέσω ηλεκτρικών πεδίων των Ψαριών	-	-	2 (0)	-	-
Αναζήτηση Κοπαδιού Ψαριών	2 (1)	-	4 (2)	1 (0)	-
Αλγόριθμος Βελτιστοποίησης Δάσους	-	-	4 (2)	1 (0)	-

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Βελτιστοποίηση Φρουτόμυγας	37 (30)	1 (1)	44 (28)	25 (16)	8 (7)
Αλγόριθμος Βελτιστοποίηση Ακρίδας	-	-	1 (1)	8 (7)	1 (1)
Διαδρομή του Μεγάλου Σολομού	2 (2)	-	-	1 (1)	-
Αλγόριθμος Βελτιστοποίησης Πράσινου Ερωδιού	-	-	1 (0)	-	-
Βελτιστοποιητής Αναζήτησης Ομάδας	49 (42)	-	2 (2)	21 (18)	-
Αλγόριθμος της Καρδιάς	-	-	2 (2)	-	-
Βελτιστοποίηση Ζευγαρώματος Μελισσών	4 (1)	-	16 (12)	36 (27)	1 (1)
Ιαπωνικοί Δενδροβάτραχοι	-	-	-	-	1 (0)
Αλγόριθμος της Πυγολαμπίδας Άλτη	-	-	1 (0)	1 (1)	-
Αλγόριθμος Πάπιας Keshtel	-	-	1 (1)	-	-
Αλγόριθμο εμπνευσμένο από το Συκώτι	1 (1)	-	-	5 (5)	1 (1)
Αλγόριθμος Φάλαινας Δολοφόνου	-	1 (1)	-	1 (0)	-
Γάμος Μελισσών	51 (46)	-	5 (4)	11 (10)	3 (2)
Αλγόριθμος Πιθήκου	-	-	5 (3)	3 (2)	10 (10)
Αναζήτηση Πιθήκων	-	-	1 (1)	2 (1)	-
Ωτοκία Κουνουπιών	-	-	-	2 (0)	-
Αλγόριθμος Ψαριού που αναπαράγεται από το Στόμα	-	1 (1)	-	-	-
Βελτιστοποίηση Περιπλάνησης Μυδιών	-	-	1 (1)	1 (1)	3 (2)
Φυσική Ανάπλαση Δάσους	1 (1)	-	-	-	-
Βελτιστοποίηση Φύσαρου	-	-	5 (4)	3 (2)	3 (1)
Βελτιστοποίηση Ανάπτυξης Φυτών	-	-	1 (1)	19 (13)	-
Αλγόριθμος Πολλαπλασιασμού Φυτών	1 (1)	-	1 (1)	1 (1)	-

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Αλγόριθμο Βελτιστοποίησης Πολικής Αρκούδας	-	-	-	2 (2)	-
Αλγόριθμος Θύματος-Θύτη	1 (1)	-	2 (2)	1 (1)	-
Αλγόριθμος Βελτιστοποίησης Ριζών Δέντρου	-	-	8 (2)	-	-
Βελτιστοποιητής Σατέν Πτιλονορρυγίδας	-	-	2 (2)	2 (2)	-
Βελτιστοποιητής Εγλωιστικού Κοπαδιού	-	-	-	1 (1)	1 (1)
Βελτιστοποίηση Όσφρησης Καρχαρία	1 (1)	-	1 (1)	2 (2)	1 (1)
Αλγόριθμος Φάλαινας Φυσητήρα	-	-	-	1 (1)	-
Αλγόριθμος Βελτιστοποίησης Βλαστοκύτταρων	1 (0)	-	-	-	-
Αναζήτηση Συμβιωτικών Οργανισμών	30 (27)	1 (1)	38 (24)	43 (33)	-
Βελτιστοποίηση Βασισμένη στο Μοντέλο της Γρίπης των Χοίρων	1 (0)	-	1 (1)	4 (3)	-
Αλγόριθμος του Λιονταριού	-	-	3 (3)	-	-
Αλγόριθμος Παρακλαδιών Ρίζας	-	-	-	2 (2)	-
Αλγόριθμος Ανάπτυξης Δέντρων	-	-	2 (2)	-	-
Αλγόριθμος Σπόρου Δέντρου	4 (3)	-	1 (1)	3 (2)	2 (1)
Ιογενές Σύστημα	-	-	15 (8)	-	-
Αναζήτηση Αποικίας Ιών	-	-	1 (1)	4 (1)	-
Αλγόριθμος Υδροχαρούς Ζιζανίου	-	-	1 (0)	-	-
Πανίδα	230 (189)	5 (5)	198 (132)	293 (206)	40 (32)
Χλωρίδα	6 (5)	-	15 (11)	37 (26)	10 (7)
Λοιπά	4 (3)	-	3 (3)	5 (5)	1 (1)

Πίνακας Π-3: Πλήθος εργασιών Αλγορίθμων Εμπνευσμένων από Φυσικά Φαινόμενα και Νόμους των Επιστημών ανά Χώρο Εφαρμογής

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Βελτιστοποίηση Τεχνητής Φυσικής	0	0	0	0	3 (3)
Μεγάλη Έκρηξη – Μεγάλη Σύνθλιψη	20 (20)	5 (5)	1 (1)	16 (16)	13 (9)
Μαύρη Τρύπα	0	0	0	4 (4)	1 (0)
Βελτιστοποίηση Κεντρομόλου Δυνάμεως	7 (7)	0	0	7 (7)	0
Αναζήτηση Φορτισμένου Συστήματος	48 (42)	5 (5)	3 (3)	6 (6)	0
Βελτιστοποίηση εμπνευσμένη από τις Χημικές Αντιδράσεις	4 (2)	11 (4)	12 (6)	8 (7)	16 (13)
Βελτιστοποίηση Σύγκρουσης Σωμάτων	24 (23)	1 (1)	0	0	0
Βελτιστοποίηση βάσει του φαινομένου του Ηλεκτρομαγνητισμού	38 (7)	5 (5)	32 (25)	9 (9)	35 (34)
Βελτιστοποίηση Κίνησης των Αερίων κατά τον Robert Brown	1 (1)	0	0	1 (1)	0
Αλγόριθμος Αναζήτησης βασισμένος στον Γαλαξία	0	2 (2)	0	2 (2)	3 (2)
Αλγόριθμος Βαρυτικής Αναζήτησης	50 (27)	17 (12)	7 (7)	64 (57)	49 (36)
Βελτιστοποίηση βασισμένη στην Υστέρηση	1 (1)	0	1 (0)	0	3 (2)
Νοήμονα Σταγόνα Νερού	2 (2)	2 (1)	18 (10)	3 (3)	13 (5)
Βελτιστοποίηση βασισμένη στο φαινόμενο Έκρηξης Νάρκης	2 (1)	1 (1)	0	2 (1)	0
Βελτιστοποίηση Ακτίνων	2 (2)	0	0	0	0
Δυναμική Σχηματισμού Ποταμών	1 (0)	1 (1)	1 (1)	0	5 (1)
Αλγόριθμος Δυναμικής της Σπειροειδούς Κινήσεως	1 (0)	0	0	0	0

	Μηχανική	Χρηματοοικονομικά	Επιχειρησιακή Έρευνα	Ενεργειακά Συστήματα	Άλλες Εφαρμογές
Αλγόριθμος Κύκλου του Νερού	6 (1)	0	0	5 (5)	1 (1)
Φαινόμενα του Σύμπαντος	70 (47)	24 (19)	8 (8)	86 (79)	65 (47)
Νόμοι Θετικών Επιστημών	126 (86)	23 (16)	48 (34)	33 (31)	57 (52)
Φυσικά Φαινόμενα	9 (3)	3 (2)	19 (11)	8 (8)	19 (7)