

# ΥΠΟΛΟΓΙΣΤΙΚΕΣ ΜΕΘΟΔΟΙ ΛΗΨΗΣ ΑΠΟΦΑΣΕΩΝ ΕΜΠΝΕΥΣΜΕΝΕΣ ΑΠΟ ΤΗ ΦΥΣΗ (NATURE INSPIRED METHODS)

---

Ένα πολύ σημαντικό ζήτημα στη Λήψη Αποφάσεων είναι η εύρεση της καλύτερης επιλογής μεταξύ των πιθανών επιλογών σε ένα πρόβλημα απόφασης. Ειδικότερα, αν μιλάμε για ένα πρόβλημα με μη προφανή λύση ή που η λύση του αλλάζει συνεχώς (δυναμικό πρόβλημα), τότε είναι προφανές πως οι συμβατικές μέθοδοι λήψης αποφάσεων δεν επαρκούν. Τέτοιου είδους προβλήματα μπορεί να είναι η εύρεση του βέλτιστου χαρτοφυλακίου μετοχών (το οποίο αλλάζει τουλάχιστον καθημερινά, λόγω των μεταβολών στις αποδόσεις των μετοχών) ή η βέλτιστη κατανομή πόρων (ανθρώπινων ή υλικών) σε προβλήματα Διαχείρισης Έργων (Project Management). Στην πρώτη περίπτωση, θα έπρεπε να επαναεισάγουμε τις αποδόσεις των μετοχών, καθώς και άλλες παραμέτρους σε ένα χρηματοοικονομικό λογισμικό κάθε φορά που αυτές αλλάζουν. Αυτή η διαδικασία θα ήταν χρονοβόρα και ενδεχομένως να μην υπήρχε μια σαφής πρόταση από το λογισμικό σε σύντομο χρονικό διάστημα (σε τέτοιες περιπτώσεις είναι σημαντικό να προλαβαίνει κανείς την επόμενη μεταβολή των μετοχών, κάτι που μπορεί να γίνεται συχνά σε μια μέρα). Στην περίπτωση της κατανομής πόρων, σε προβλήματα πολλών παράλληλων εργασιών και πολλαπλών περιορισμών, για να ελεγχθούν όλα τα πιθανά σενάρια και να βρεθεί αυτό που συμφέρει την εταιρία και ελαχιστοποιεί το κόστος της, θα έπρεπε να δεσμευτεί τεράστια υπολογιστική δύναμη και να θεωρηθεί δεδομένο ότι δεν θα υπάρξουν καθυστερήσεις στο project και άρα δε θα πρέπει να επαναξιολογηθεί η κατανομή των πόρων.

## 1 Τεχνητή Νοημοσύνη

Μια λύση για τα παραπάνω προβλήματα που προκύπτουν, έρχεται να δώσει ο χώρος της Τεχνητής Νοημοσύνης (Artificial Intelligence). Ήδη από τον Β΄ Παγκόσμιο Πόλεμο, αρχής γενομένης από τον Alan Turing (1) που κλήθηκε να αποκρυπτογραφήσει τα μηνύματα των Γερμανών με μια ευρετική μέθοδο, εμφανίστηκαν κάποιες τεχνικές για την επίλυση δύσκολων προβλημάτων.

Ο χώρος της Τεχνητής Νοημοσύνης (Artificial Intelligence) παρουσιάζει ερευνητικά ένα συνεχώς αυξανόμενο ενδιαφέρον, μετρώντας ήδη 60 (2) χρόνια σταδιακής ανάπτυξης. Στις αρχές της δεκαετίας του 1990 οι ερευνητές εμβάθυναν περισσότερο στην ανάπτυξη υπολογιστικών συστημάτων με νοημοσύνη παρόμοια με την ανθρώπινη. Εστίασαν στην δημιουργία κανόνων μέσα στα συστήματα αυτά, ώστε να μπορούν να ληφθούν σχετικές αποφάσεις. Ωστόσο, και πριν το 1990 είχαν κάνει την εμφάνισή τους νοήμονες προσεγγίσεις όπως Νευρωνικά Δίκτυα (Artificial Neural Networks – δεκαετία του 1960) και Γενετικοί Αλγόριθμοι (Genetic Algorithms – δεκαετία του 1970), που όμως δεν είχαν αξιοποιηθεί καταλλήλως. Στην ουσία ο J. H. Holland με την ομάδα του (3), όταν πρότειναν τον πρώτο Γενετικό Αλγόριθμο (Genetic Algorithm), παρουσίασαν την πρώτη μέθοδο Υπολογιστικής Νοημοσύνης (Computational Intelligence). Επόμενο ήταν να δημιουργηθεί το ερώτημα: μπορούν άλλα φαινόμενα να αποτελέσουν μοντέλα για νέα Νοήμονα Υπολογιστικά Συστήματα; Κατ' επέκταση, η Υπολογιστική Νοημοσύνη (4) (5) εστίασε στους τομείς:

- Μηχανική Μάθηση (Machine learning) (6) (7) (8) (9)
- Συστήματα Ασαφών Κανόνων (Fuzzy rule based systems) (10) (11) (12) (13)
- Νευρωνικά Δίκτυα (Neural Networks) (14) (15)
- Εξελικτικούς Αλγόριθμους (Evolutionary Algorithms)

## 2 Αλγόριθμοι Εμπνευσμένοι από τη Φύση (Nature Inspired Algorithms)

Οι Εξελικτικοί Αλγόριθμοι χρησιμοποιήθηκαν ευρέως για την επίλυση προβλημάτων βελτιστοποίησης, τα οποία συνήθως δεν είχαν διακριτή λύση ή ήταν δυναμικά με αποτέλεσμα να αλλάζει συνεχώς η βέλτιστη λύση. Αυτό το γεγονός, σε συνδυασμό με τις ήδη δημοσιευμένες μεθόδους Simulated Annealing (16), Stochastic Diffusion Search (17), Particle Swarm Optimization (18), Differential Evolution (19) και Grammatical Evolution (20), 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 εκείνη τη στιγμή μέχρι σήμερα, το πλήθος των δημοσιευμένων μεθόδων αυξάνεται με ταχύτατους ρυθμούς. Επιπλέον, αρκετοί ερευνητές πρότειναν διάφορα

Υβριδικά Σχήματα (Hybrid Schemes) μεταξύ 2 ή περισσότερων από αυτές τις μεθόδους, αλλά και αρκετά Προσαρμοστικά Νοήμονα Συστήματα (Adaptive Intelligent Systems) (21).

## 2.1 Ταξινόμηση και Κατηγοριοποίηση

Αρχικά, η εργασία των Fister et al. (22) περιέχει ένα μεγάλο μέρος των δημοσιευμένων Nature Inspired μεθόδων (μέχρι και το 2013). Οι συγγραφείς χώρισαν τους αλγόριθμους σε τέσσερις μεγάλες κατηγορίες βάσει της πηγής έμπνευσης κάθε αλγόριθμου:

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

Η κατηγορία που περιέχει μεθόδους βασισμένες στην Ευφυΐα Σμήνους (Swarm Intelligence) έχει αλγόριθμους που μοντελοποιούν συμπεριφορές ζώων και μικροοργανισμών που δημιουργούν ομάδες (κοπάδια, αγέλες, σμήνη κλπ). Σε μεγάλο βαθμό, οι αλγόριθμοι αυτοί στηρίζονται στην αλληλεπίδραση της εκάστοτε ομάδας και στην επιρροή του καλύτερου (καλύτερης λύσης) στους υπόλοιπους (λύσεις). Αντιπροσωπευτικό παράδειγμα είναι ο Particle Swarm Optimization (23), που βασίζεται στον τρόπο με τον οποίο πετούν ομάδες πτηνών: κάθε πτηνό (λύση) μετακινείται στο χώρο με βάση την μετακίνηση όλου του σμήνους και την μετακίνηση του καλύτερου.

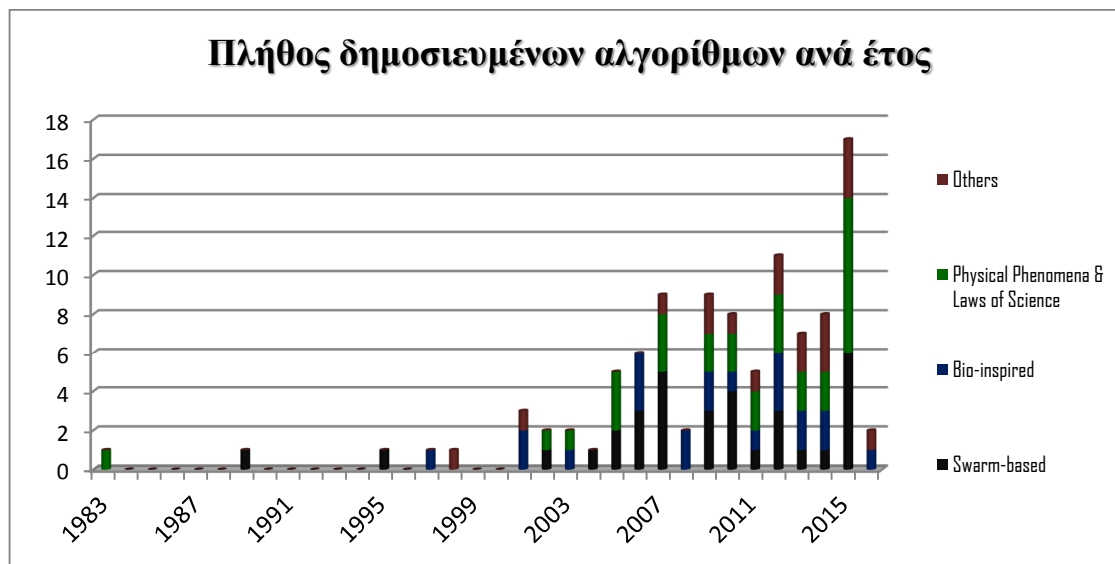
Τα Βιο-εμπνευσμένα (Bio-inspired) σχήματα είναι αλγοριθμικές μέθοδοι που βασίζονται σε συμπεριφορές ζώων και πτηνών ή σε οργανικές λειτουργίες. Οι συμπεριφορές αυτές παρατηρούνται μεμονωμένα από κάθε οργανισμό και δεν είναι σχετικές με συμπεριφορά ομάδας. Ένα παράδειγμα είναι ο Dolphin Echolocation (24), ο οποίος μοντελοποιεί τον τρόπο που αναγνωρίζουν τα δελφίνια τον χώρο γύρω τους μέσω του ηχοεντοπισμού (echolocation).

Σύμφωνα με πρόσφατη έρευνα, η οποία έγινε στα πλαίσια της διδακτορικής διατριβής του Αλ. Τζανέτου (25), η τρίτη κατηγορία μετονομάστηκε σε Φυσικά Φαινόμενα και Νόμοι των Επιστημών (Physical Phenomena & Laws of Science) και εκτός από τις ήδη υπάρχουσες μεθόδους που ήταν εμπνευσμένες από φαινόμενα της Φυσικής και της Χημείας, προστέθηκαν επίσης όσες μέθοδοι βασίζονται στον ευρύτερο χώρο των επιστημών, αλλά και όσες μοντελοποιούν φυσικά φαινόμενα, όπως αυτό της διάδοσης των κυμάτων (26), του τυφώνα (27) κ.ά.

Επιπλέον, υπό συζήτηση βρίσκεται η ιδέα να δημιουργηθεί μια νέα κατηγορία, η οποία να περιέχει το μεγαλύτερο μέρος των αλγορίθμων από μια ήδη υπάρχουσα κατηγορία αταξινομήτων ως τώρα αλγορίθμων (others). Η εν λόγω κατηγορία θα περιέχει μεθόδους που

βασίζονται σε κοινωνικά ή πολιτικά φαινόμενα (Social theory), όπως το αναρχικό μοντέλο (28) ή τον τρόπο εύρεσης του καλύτερου παίκτη από ένα πρωτάθλημα (29).

Οι κατηγορίες και οι δημοσιευμένες μέθοδοι ανά έτος παρουσιάζονται στο Γράφημα 1. παρακάτω.



Γράφημα 1. Πλήθος δημοσιευμένων μεθόδων εμπνευσμένων από Φαινόμενα της Φύσης ανά έτος

Διερευνώντας κανείς τις ήδη δημοσιευμένες μεθόδους (μέχρι τώρα, επίσημα, οι μέθοδοι είναι 118), μπορεί να ορίσει ένα μοτίβο (pattern) που σε μεγάλο βαθμό ακολουθούν οι μέθοδοι αυτές.

Αρχικά, συνήθως γίνεται αρχικοποίηση παραμέτρων και παραγωγή λύσεων ή λύσης με τυχαίο τρόπο. Αν ο αλγόριθμος βασίζεται στη βελτίωση ενός πληθυσμού (population-based), τότε παράγονται η 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: Big Bang – Big Crunch, League Championship Algorithm, Termite Colony Optimization και οι περισσότερες Swarm Intelligence μέθοδοι. Αντιπροσωπευτικά παραδείγματα αλγορίθμων που βασίζονται στο μονοπάτι που ακολουθεί μια λύση (trajectory-based) είναι οι Intelligent Water Drop και River Formation Dynamics, οι οποίοι ακολουθούν τη βέλτιστη λύση, καθώς αυτή βελτιώνεται μέσα από κάποια βήματα.

Το επόμενο βήμα είναι η βελτίωση των λύσεων, είτε μέσω κάποιων κανόνων (rule-based), είτε 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 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, ο 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 Gravitational Search (GSA), ο οποίος βελτιώνει με 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 attraction ή non-attraction-based αντίστοιχα. Στην ουσία οι attraction-based μέθοδοι 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 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 ο Electromagnetism-like Optimization, ενώ γνωστοί αλγόριθμοι που δεν έχουν τέτοια έλξη μεταξύ λύσεων είναι οι Chemical-reaction-inspired Optimization, Biogeography-based Optimization και Colliding Bodies Optimization. Οι Swarm Intelligence αλγόριθμοι είναι σε μεγάλο ποσοστό attraction-based, καθώς η θεωρία του σμήνους στηρίζεται στην αλληλεπίδραση των έμβιων οργανισμών μεταξύ τους ανήκοντας σε ομάδες.

Ύστερα γίνεται αξιολόγηση των νέων λύσεων (fitness evaluation). Εδώ συνήθως χρησιμοποιείται κάποιος μηχανισμός αποδοχής-απόρριψης λύσεων, όπως το Metropolis Criterion (30) που περιέχεται στον Simulated Annealing. Πολλές φορές ο μηχανισμός αυτός έχει σχέση με Πιθανοθεωρητικά Μοντέλα, με σκοπό μια πιθανότητα (που σταδιακά μειώνεται) να κάνει πιο αυστηρό τον αλγόριθμο στην αποδοχή νέων λύσεων, ώστε να μην «κολλάει» σε τοπικά βέλτιστα.

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

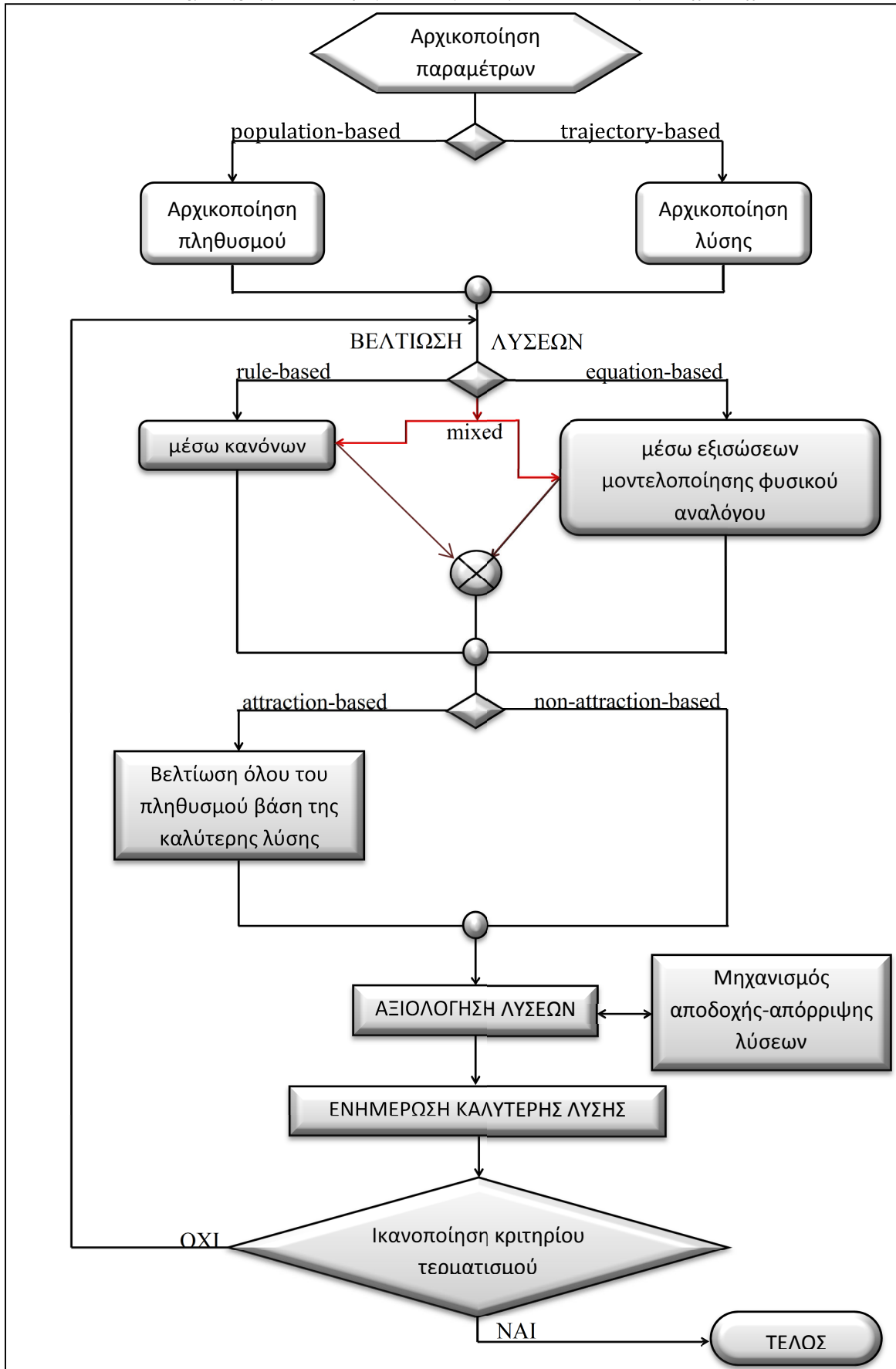
Διαγραμματικά, αυτό το μοτίβο μπορεί να αποδοθεί, όπως στο Σχεδιάγραμμα 1.

Συνοψίζοντας, από τα παραπάνω, μπορεί κανείς να ταξινομήσει έναν αλγόριθμο βάσει:

- του πλήθους των λύσεων που παράγει, σε:
  - ❖ **population-based**, όπου παράγει έναν πληθυσμό από λύσεις
  - ❖ **trajectory-based**, όπου παράγει μία λύση και την βελτιώνει ή την ενημερώνει

- του τρόπου που βελτιώνει τις λύσεις, σε:
  - ❖ **rule-based**, όπου η βελτίωση των λύσεων στηρίζεται σε κανόνες
  - ❖ **equation-based**, όπου οι λύσεις βελτιώνονται μέσω κάποιας εξίσωσης φυσικού αναλόγου
  - ❖ **mixed**, όπου συντελούνται και τα δύο
- της διάδρασης των λύσεων, σε:
  - ❖ **attraction-based**, όπου η βέλτιστη λύση έλκει κοντά της τις υπόλοιπες
  - ❖ **non-attraction-based**, όπου δεν υπάρχει κάποια διάδραση μεταξύ των λύσεων

Σχεδιάγραμμα 1. Μοτίβο μεθόδων εμπνευσμένων από Φαινόμενα της Φύσης



### 3 Εφαρμογή στη Λήψη Αποφάσεων

Ο χώρος της Λήψης Αποφάσεων ασχολείται με την επιλογή της καλύτερης εναλλακτικής μεταξύ των διαθέσιμων επιλογών (31). Συχνά, το να προσδιοριστούν όλες οι διαθέσιμες επιλογές είναι εξαιρετικά δύσκολο. Για παράδειγμα, αν έχουμε ένα πλήθος  $n = 30$  διαθέσιμων μετοχών και θέλουμε να τις συνδυάσουμε ανά  $r = 5$ , για να ελέγξουμε όλους τους πιθανούς συνδυασμούς, θα έπρεπε να κάνουμε

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = \frac{30!}{5!(30-5)!} = 142.506$$

δηλαδή 142.506 συνδυασμούς και να ελέγξουμε για κάθε συνδυασμό την απόδοση που δίνει. Το εγχείρημα αυτό είναι χρονοβόρο και απαιτεί δέσμευση τεράστιων πόρων (π.χ. υπολογιστικής δύναμης).

Στη Λήψη Αποφάσεων μας ενδιαφέρει τόσο η εύρεση της βέλτιστης επιλογής, αλλά και σε όσο το δυνατόν συντομότερο χρονικό διάστημα. Γι' αυτό το λόγο επιστρατεύουμε μετα-ευρετικές τεχνικές (metaheuristics) που είναι εμπνευσμένες από τη φύση (Nature Inspired techniques). Γενικά, τα προβλήματα τα οποία καλείται να αντιμετωπίσει κανείς στον ευρύτερο κλάδο της Λήψης Απόφασης, διακρίνονται σε:

- 1) προβλήματα βελτιστοποίησης (optimization)
- 2) προβλήματα κατηγοριοποίησης (classification) ή συσταδοποίησης (clustering)
- 3) προβλήματα πρόβλεψης (forecasting)

Σε γενικές γραμμές, όλα τα προβλήματα μπορούν να μετατραπούν σε προβλήματα βελτιστοποίησης, εφόσον οριστεί μια αντικειμενική συνάρτηση που χρήζει βελτιστοποίησης.

#### 3.1 Προβλήματα Βελτιστοποίησης

Τα προβλήματα βελτιστοποίησης (optimization problems) είναι προβλήματα στα οποία πρέπει να επιλεγεί η καλύτερη μεταξύ των αποδεκτών (feasible) λύσεων. Ανάλογα με τις μεταβλητές του εκάστοτε προβλήματος, αυτά μπορούν να διαχωριστούν σε συνεχούς (continuous) ή διακριτού (discrete) χώρου.

Η γενική έκφραση ενός προβλήματος βελτιστοποίησης συνεχούς χώρου είναι (32):

$$\begin{aligned} &\mathbf{minimize} && f(x) \\ & && \\ &\mathbf{subject\ to} && g_i(x) \leq 0, \quad i = 1, 2, \dots, m \\ & && h_i(x) = 0, \quad i = 1, 2, \dots, p \end{aligned}$$



όπου

- $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  είναι η αντικειμενική συνάρτηση που πρέπει να βελτιστοποιηθεί ως προς  $x$
- $g_i(x) \leq 0$  λέγονται περιορισμοί ανισότητας (inequality constraints)
- $h_i(x) = 0$  λέγονται περιορισμοί ισότητας (equality constraints)

Εδώ έχουμε πρόβλημα ελαχιστοποίησης. Για προβλήματα μεγιστοποίησης, απλώς πολλαπλασιάζεται η αντικειμενική συνάρτηση με  $-1$ .

Αντίστοιχα, η γενική έκφραση ενός προβλήματος βελτιστοποίησης  $A$ , διακριτού χώρου (που αλλιώς ονομάζεται «συνδυαστικό πρόβλημα βελτιστοποίησης» ή combinatorial optimization problem) είναι (33):

$$A : (I, f, m, g)$$

όπου

- $I$  είναι μια σειρά περιπτώσεων
- δεδομένης της περίπτωσης  $x \in I$ ,  $f(x)$  είναι ένα σύνολο αποδεκτών λύσεων
- δεδομένης της περίπτωσης  $x$  και της αποδεκτής λύσης  $y$  του  $x$ , το  $m(x, y)$  δίνει το μέτρο του  $y$ , το οποίο είναι συνήθως θετικός αριθμός από το σύνολο των πραγματικών αριθμών
- $g$  είναι η συνάρτηση βελτιστοποίησης, όπου ανάλογα είναι  $\min$  ή  $\max$

Ο στόχος είναι να βρεθεί η βέλτιστη περίπτωση  $x$ , για την οποία η αποδεκτή λύση  $y$  θα δίνει:

$$m(x, y) = g\{m(x, y') \mid y' \in f(x)\}$$

Αν στα παραπάνω προσθέσουμε και τις συνθήκες:

- το μέγεθος κάθε αποδεκτής λύσης  $y \in f(x)$  ορίζεται πολυωνυμικά από το μέγεθος της δεδομένης περίπτωσης  $x$
- οι εκφράσεις  $\{x \mid x \in I\}$  και  $\{(x, y) \mid y \in f(x)\}$  αναγνωρίζονται σε πολυωνυμικό χρόνο
- η  $m$  υπολογίζεται σε πεπερασμένο χρόνο

τότε έχουμε ένα πρόβλημα απόφασης «μη ντετερμινιστικό, πολυωνυμικού χρόνου» (nondeterministic, polynomial time) ή εν συντομία NP (34).

Μια άλλη κατηγοριοποίηση (όλων των ειδών) των προβλημάτων είναι σε (α) στατικά και (β) δυναμικά. Τα στατικά προβλήματα έχουν διακριτές λύσεις ή υπολογίζεται η βέλτιστη λύση μια φορά. Αντίθετα, στα δυναμικά, η βέλτιστη λύση αλλάζει συνεχώς βάσει κάποιων

παραμέτρων. Αυτό τα καθιστά ιδιαίτερη πρόκληση για τους ερευνητές που ασχολούνται με αλγόριθμους βελτιστοποίησης.

### 3.2 Προβλήματα Κατηγοριοποίησης και Συσταδοποίησης

Στα προβλήματα κατηγοριοποίησης (classification problems) έχουμε ως αντικειμενικό στόχο να κατηγοριοποιήσουμε τα δεδομένα σε ομάδες με κοινά χαρακτηριστικά. Παρομοίως, στα προβλήματα συσταδοποίησης (clustering problems) δημιουργούμε κάποιες κλάσεις (clusters) στις οποίες και ταξινομούμε δεδομένα. Ένα παράδειγμα προβλήματος κατηγοριοποίησης είναι η κατηγοριοποίηση των ασθενών βάσει κάποιων χαρακτηριστικών (βήχας, πυρετός κλπ) στο διακριτό ερώτημα «έχει ο ασθενής γρίπη ή όχι;». Άλλο παράδειγμα είναι η κατηγοριοποίηση γυναικών σε ένα από τα στάδια καρκίνου του μαστού, βάσει της μαστογραφίας, όπου εκεί έχουμε ως δεδομένα το μέγεθος του όγκου και άλλες πληροφορίες σχετικές με αυτόν. Ένα πρόβλημα συσταδοποίησης ή ομαδοποίησης δεδομένων (clustering analysis) θα μπορούσε να θεωρηθεί το πρόβλημα ομαδοποίησης των χρηστών του γνωστού ιστότοπου Netflix, όπου κάθε χρήστης ομαδοποιείται σε σχέση με άλλους χρήστες σύμφωνα με τις κοινές ταινίες που έχει παρακολουθήσει με αυτούς.

Και τα δύο είδη προβλημάτων μπορούν να αντιμετωπιστούν ως προβλήματα βελτιστοποίησης αρκεί να οριστεί ως αντικειμενική συνάρτηση η Ευκλείδεια απόσταση ή η απόσταση Manhattan (ή Minkowski γενικότερα) μεταξύ των δεδομένων:

$$Euclidean\ distance = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

$$Manhattan\ distance = \sum_{i=1}^n |q_i - p_i|$$

όπου αθροίζουμε την απόλυτη διαφορά του  $q$  στοιχείου (δεδομένου) με το  $p$  στοιχείο, σε κάθε  $i$  διάσταση του προβλήματος στην μία περίπτωση και την ίδια απόσταση υψωμένη στο τετράγωνο στην άλλη περίπτωση. Με στόχο την ελαχιστοποίηση αυτών των αποστάσεων, μπορούμε να χωρίσουμε τα δεδομένα σε ομάδες γειτόνων, όπως περίπου γίνεται σε αντίστοιχες ευρετικές τεχνικές (35).

Συνήθως, έχοντας ήδη κάποια δεδομένα, τα χωρίζουμε σε σύνολα εκμάθησης (training set) του αλγόριθμου και σε σύνολα ελέγχου (test set), ώστε να ελέγξουμε σε τι βαθμό ομαδοποιεί τα δεδομένα σωστά ένας αλγόριθμος.

### 3.3 Προβλήματα Πρόβλεψης

Σε ένα πρόβλημα πρόβλεψης (forecasting problem) ο στόχος είναι να βρεθούν οι τιμές των σταθμίσεων (weights), οι οποίες μπορούν να δώσουν ένα τέτοιο (γραμμικό) μοντέλο συσχέτισης κάποιων δεδομένων με μια μεταβλητή στόχου (target variable). Για να ελέγξουμε το μοντέλο, πάλι ακολουθούμε τη διαδικασία χωρισμού των δεδομένων σε σύνολο εκμάθησης και σύνολο ελέγχου. Σε αυτή την κατηγορία προβλημάτων βρίσκουμε λίγες εφαρμογές των μεθόδων που είναι εμπνευσμένες από τη φύση, καθώς συνήθως τα μοντέλα πρόβλεψης πρέπει να είναι μη γραμμικά στην πραγματικότητα και γι' αυτό υπάρχουν πιο έγκυρες τεχνικές εύρεσης τέτοιων μοντέλων, όπως ο Γενετικός Προγραμματισμός (Genetic Programming) (36).

### 3.3 Παραδείγματα αλγορίθμων ανά χώρο εφαρμογής

Παρακάτω ακολουθεί ένας πίνακας με ενδεικτικές εργασίες ανά χώρο εφαρμογής, όπου έχουν χρησιμοποιηθεί αλγόριθμοι εμπνευσμένοι από τη φύση:

| Κατηγορία προβλημάτων | Αλγόριθμος                      | Εργασία |
|-----------------------|---------------------------------|---------|
| Βελτιστοποίησης       |                                 |         |
| -Εφαρμογές Μηχανικού  | Bat algorithm                   | (37)    |
| -Χρηματοοικονομικά    | Particle Swarm Optimization     | (38)    |
| -Ενεργειακά           | Shuffled Frog leaping algorithm | (39)    |
| -Επιχειρησιακή Έρευνα | Ant System                      | (40)    |
| -Άλλες εφαρμογές      | Chemical Reaction Optimization  | (41)    |
| Κατηγοριοποίησης      | Honey bees mating               | (42)    |
| Συσταδοποίησης        | Black Hole                      | (43)    |
| Πρόβλεψης             | Firefly algorithm               | (44)    |

## Βιβλιογραφία

1. *Alan Turing: Mathematical Mechanist*. **Beavers, Anthony**. s.l. : Waltham: Elsevier, 2013, Cooper, S. Barry; van Leeuwen, Jan. Alan Turing: His Work and Impact, pp. pp. 481–485.
2. *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. **McCarthy, J., Minsky, M., Rochester, N., Shannon, C.E.** August 1955.
3. *Adaption in natural and artificial systems*. **Holland, J. H.** [ed.] Ann Arbor. s.l. : MI: University of Michigan Press, 1975.
4. **Nilsson, N. J.** *Artificial intelligence: a new synthesis*. s.l. : Elsevier, 1998.
5. **Chen, Z.** *Computational intelligence for decision support*. s.l. : CRC Press, 1999.
6. **Michalski, R. S., Carbonell, J., & Mitchell, T. M.** *Machine learning I: An AI Approach*. 1983.
7. *Machine learning: An artificial intelligence approach*. **Mitchell, T. M., Anderson, J. R., Carbonel, J. G., & Michalski, R. S.** 1983, Vol. II.
8. **Kodratoff, Y., & Michalski, R. S.** *Machine learning: an artificial intelligence approach*. 1990. Vol. III.
9. **Mitchell, T. M.** *Machine learning. Artificial Intelligence*. 1997.
10. *Fuzzy sets*. **Zadeh, L. A.** 3, 1965, Information and control, Vol. 8, pp. 338-353.
11. *Research commentary: Technology-mediated learning—A call for greater depth and breadth of research*. **Alavi, M., & Leidner, D. E.** 1, 2001, Information systems research, Vol. 12, pp. 1-10.
12. *Multiagent systems: a modern approach to distributed artificial intelligence*. **Weiss, G.** s.l. : MIT press, 1999.
13. *Foundations of fuzzy control: a practical approach*. **Jantzen, J.** s.l. : John Wiley & Sons, 2013.
14. *Two theorems of statistical separability in the perceptron*. **Rosenblatt, F.** s.l. : United States Department of Commerce, 1958.
15. *Adaptive switching circuits*. **Widrow, B., & Hoff, M. E.** 1, August 1960, IRE WESCON convention record, Vol. 4, pp. 96-104.
16. *Optimization by simulated annealing*. **Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P.** 4598, 1983, science, Vol. 220, pp. 671-680.
17. *Stochastic searching networks*. **Bishop, J. M.** London, UK : s.n., 1989. 1st IEE Conf. on Artificial Neural Networks. pp. 329-331.

18. *A new optimizer using particle swarm theory*. **Eberhart, R. C., & Kennedy, J.** 1995, October. Proceedings of the sixth international symposium on micro machine and human science. Vol. 1, pp. 39-43.
19. *Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces*. **Storn, R., & Price, K.** 4, 1997, Journal of global optimization, Vol. 11 , pp. 341-359.
20. *Grammatical evolution: Evolving programs for an arbitrary language*. **Ryan, C., Collins, J. J., & Neill, M. O.** s.l. : Springer Berlin Heidelberg, 1998, April. European Conference on Genetic Programming. pp. 83-96.
21. *Computational intelligence: synergies of fuzzy logic*. **Siddique, N., & Adeli, H.** s.l. : John Wiley & Sons, 2013, neural networks and evolutionary computing.
22. *A brief review of nature-inspired algorithms for optimization*. **Fister Jr, I., Yang, X. S., Fister, I., Brest, J., & Fister, D.** 2013. arXiv preprint arXiv:1307.4186.
23. *A new optimizer using particle swarm theory*. **Eberhart, R., & Kennedy, J.** s.l. : IEEE, 1995, October. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. Vol. MHS'95, pp. 39-43.
24. *A new optimization method: dolphin echolocation*. **Kaveh, A., & Farhoudi, N.** 2013, Advances in Engineering Software, Vol. 59, pp. 53-70.
25. *Nature Inspired Optimization Algorithms Related to Physical Phenomena and Laws of Science: A survey*. **Tzanetos, A. and Dounias, G.** (under review).
26. *Water wave optimization: a new nature-inspired metaheuristic*. **Zheng, Y. J.** 2015, Computers & Operations Research, Vol. 55, pp. 1-11.
27. *Hurricane-based optimization algorithm*. **Rbough, I., & El Imrani, A. A.** 2014, AASRI Procedia, Vol. 6, pp. 26-33.
28. *Anarchic Society Optimization: A human-inspired method*. **Ahmadi-Javid, A.** s.l. : IEEE, 2011, June. 2011 IEEE Congress of Evolutionary Computation (CEC). pp. 2586-2592.
29. *Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts*. **Osaba, E., Diaz, F., & Onieva, E.** 1, 2014, Applied Intelligence, Vol. 41, pp. 145-166.
30. *Equations of State Calculations by Fast Computing Machines*. **Metropolis, N., et al., et al.** 6, 1953, Journal of Chemical Physics, Vol. 21, pp. 1087–1092.
31. **Τσάκωνας Α., Δούνιας Γ.** *Εξελικτικός Υπολογισμός και Εξόρυξη Δεδομένων*. s.l. : Εκδόσεις Κλειδάριθμος, 2008. ISBN 978-960-461-189-8.
32. *Convex Optimization*. **Boyd, Stephen P. and Vandenberghe, Lieven.** s.l. : Cambridge University Press, 2004, p. 129.

33. **Ausiello, Giorgio and al., et.** *Complexity and Approximation* . (Corrected ed.). s.l. : Springer, 2003.
34. **Kann, Viggo.** *On the Approximability of NP-complete Optimization Problems*. s.l. : Royal Institute of Technology, 1992.
35. *Constrained k-means clustering with background knowledge.* **Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S.** June 2001, ICML, Vol. 1, pp. 577-584.
36. *Real-time wave forecasting using genetic programming.* **Gaur, S., & Deo, M. C.** 11, 2008, Ocean Engineering, Vol. 35, pp. 1166-1172.
37. *Bat algorithm: a novel approach for global engineering optimization.* **Yang, X. S., & Hossein Gandomi, A.** 5, 2012, Engineering Computations, Vol. 29, pp. 464-483.
38. *Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem.* **Zhu, H., Wang, Y., Wang, K., & Chen, Y.** 8, 2011, Expert Systems with Applications, Vol. 38, pp. 10161-10169.
39. *Optimization of water distribution network design using the shuffled frog leaping algorithm.* **Eusuff, M. M., & Lansey, K. E.** 3, Journal of Water Resources Planning and Management : s.n., 2003, Vol. 129, pp. 210-225.
40. *Ant system for job-shop scheduling.* **Coloni, A., Dorigo, M., Maniezzo, V., & Trubian, M.** 1, s.l. : Statistics and Computer Science, 1994, Belgian Journal of Operations Research, Vol. 34, pp. 39-53.
41. *Error-correction-code allocation using the chemical reaction optimization algorithm.* **Eldos, T., Nazih, W., & Eslemery, H.** 3, 2013, International Journal of Electrical & Computer Sciences, Vol. 13, pp. 54-57.
42. *Honey bees mating optimization algorithm for financial classification problems.* **Marinaki, M., Marinakis, Y., & Zopounidis, C.** 3, 2010, Applied Soft Computing, Vol. 10, pp. 806-812.
43. *Black hole: A new heuristic optimization approach for data clustering.* **Hatamlou, A.** 2013, Information sciences, Vol. 222, pp. 175-184.
44. *Support vector regression with chaos-based firefly algorithm for stock market price forecasting.* **Kazem, A., Sharifi, E., Hussain, F. K., Saberi, M., & Hussain, O. K.** 2, 2013, Applied soft computing, Vol. 13, pp. 947-958.