Ε Λ Λ Η Ν Ι Κ Η Δ Η Μ Ο Κ Ρ Α Τ Ι Α
ΔΙΙΔΡΥΜΑΤΙΚΟ ΔΙΑΤΜΗΜΑΤΙΚΟ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΑΚΑΔΗΜΑΪΚΟΥ ΕΤΟΥΣ 2022-23
ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ ΚΑΙ
ΛΗΨΗ ΑΠΟΦΑΣΕΩΝ –
MASTER OF SCIENCE IN OPERATIONAL
RESEARCH AND DECISION MAKING
(ΠΔ 59 /2021/ΦΕΚ 145Α'/17.08.2021)

ΣΤΡΑΤΙΩΤΙΚΗ ΣΧΟΛΗ ΕΥΕΛΠΙΔΩΝ
Τμήμα Στρατιωτικών Επιστημών

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
Σχολή Μηχανικών Παραγωγής & Διοίκησης

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

«Εφαρμογή Μεθόδων Βελτιστοποίησης και Τεχνητής Νοημοσύνης σε προβλήματα Διοικητικής Μέριμνας»

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

Μαρία Μωραΐτη

Α.Μ.: 2021018116

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΝΙΚΟΛΑΟΣ ΙΩ.ΔΑΡΑΣ

ΑΘΗΝΑ

ΙΑΝΟΥΑΡΙΟΣ 2023

Η Μεταπτυχιακή Διατριβή της κας Μωραΐτη Μαρίας εγκρίνεται:

ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

Καθηγητής ΟΝΟΜΑΤΕΠΩΝΥΜΟ (Επιβλέπων)

Νικόλαος Δάρας, Καθηγητής

Καθηγητής ΟΝΟΜΑΤΕΠΩΝΥΜΟ

Νικόλαος Ματσατσίνης, Καθηγητής

Nikolaos Matsatsinis

Digitally signed by Nikolaos Matsatsinis
Date: 2023.01.26 18:07:42 +02'00'

Καθηγητής ΟΝΟΜΑΤΕΠΩΝΥΜΟ

Νικόλαος Παπαδάκης, Αν. Καθηγητής

Νικόλαος Παπαδάκης

Για τον Αντώνη,

τη Σόφη,

τον Άγγελο και

τον Μαθιό

Ευχαριστίες

Πρωτίστως θα ήθελα να ευχαριστήσω τον επιβλέποντα της διατριβής κύριο Νικόλαο Δάρα για την έμπνευση του θέματος, καθώς και την υπομονή, την κατανόηση που έδειξε μέχρι την περάτωση της από μεριάς μου.

Επιπροσθέτως θα ήθελα να ευχαριστήσω τους συμφοιτητές μου για την δημιουργία ομαδικού πνεύματος συνεργασίας και την πολύτιμη βοήθεια τους σε όλη τη διάρκεια του μεταπτυχιακού προγράμματος.

Κλείνοντας θα ήθελα να ευχαριστήσω την οικογένεια μου για την αμέριστη συμπαράσταση τον τελευταίο χρόνο.

Μαρία Μωραΐτη

Ιανουάριος 2023

Περίληψη

Η διοικητική μέριμνα αποτελεί σημείο συνάντησης πολλών επιμέρους επιστημών όπως της επιχειρησιακής έρευνας, της πληροφορικής, της στατιστικής, της λογιστικής αλλά και της υπάρχουσας κοινωνικής κατάστασης. Οι επαγγελματίες που θα κληθούν να διοικήσουν θα αντιμετωπίσουν μια σειρά από προβλήματα δομημένα ή αδόμητα, αντιμετωπίσιμα ή όχι, αναμενόμενα ή απροσδόκητα, κρίσιμα ή μικρότερης σημασίας. Το πρώτο βήμα για την επίλυση του προβλήματος είναι η μοντελοποίησή του. Για την μοντελοποίηση και στη συνέχεια την επίλυση αυτών των προβλημάτων έχουν αναπτυχθεί μεθοδολογίες και αλγόριθμοι. Προαπαιτούμενο της επίλυσης είναι η διαχείριση του μεγάλου όγκου πληροφοριών, μη διαχειρίσιμου από έναν άνθρωπο, όσο προετοιμασμένος, μορφωμένος ή έμπειρος και αν είναι. Η ορθή απόφαση δεν είναι πλέον προφανής αλλά απαιτεί μία σειρά πολύπλοκων μαθηματικών υπολογισμών για να βρεθεί. Οι μαθηματικοί υπολογισμοί θα γίνουν με τη χρήση εργαλείων και μεθόδων βελτιστοποίησης. Στη συνέχεια οι υπολογιστές θα «εκπαιδευτούν» με μεθόδους μηχανικής μάθησης ώστε να επιλύουν τέτοιου είδους προβλήματα. Σε αυτή τη διατριβή θα παρατεθούν οι βασικότεροι αλγόριθμοι βελτιστοποίησης και οι εφαρμογές τεχνητής νοημοσύνης που μπορούν να εφαρμοστούν για την ορθή επίλυση των προβλημάτων που αντιμετωπίζει η διοικητική μέριμνα, μαζί με ψευδοκώδικες και παραδείγματα εφαρμογών.

Abstract

Logistics support is the crossroad of many categories of science such as operational research, computer science, statistics, accounting and current social circumstances. The professionals who work on management are facing a plethora of problems that can be structured or non-structured, solvable or non-solvable, expected or unexpected, critical or minor. Solving such problems, if possible, requires modelling. To model and later successfully solve those problems many methodologies and algorithms have been developed. To solve logistics support problems a massive amount of information is required to be managed, such amount is not manageable by individuals, no matter how well prepared, educated or experienced. The best decision is not obvious but requires complex mathematical calculations. The mathematical calculations will be done by optimization technics. Following this process, the computer will be 'educated' through machine learning to solve those kinds of problems. The optimization algorithms and artificial intelligence applications that have been used by logistics support will be presented in this dissertation, along with pseudocodes and real-life examples.

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος διαγραμμάτων

Κατάλογος Σχημάτων

Εισαγωγή

Από την απαρχή του κόσμου διακρίνει κανείς στοιχεία διοικητικής μέριμνας, πολύ διαφορετικά από αυτά που διακρίνουμε σήμερα. Οι αγέλες των άγριων ζώων έχουν έναν αρχηγό, όπως και οι πρώτες κοινωνίες των ανθρώπων είχαν έναν ηγέτη. Η ύπαρξη του ηγέτη-αρχηγού είναι ζωτικής σημασίας για την οργάνωση της ομάδας, της συλλογή των πληροφοριών και τέλος την απόφαση. Ακόμη και μία δημοκρατική ομάδα όπου η απόφαση θα ληφθεί από την πλειοψηφία, έχει ανάγκη να οργανωθεί από έναν επικεφαλής.

Τα πρώτα στοιχεία διοίκησης είναι εμφανή από τους μεγάλους πολιτισμούς της αρχαιότητας. Η κατασκευή των μεγαλόπρεπων ναών της αρχαίας Αιγύπτου και της Μεσοποταμίας απαιτούσε αυστηρό χρονικό προγραμματισμό, στην Κίνα βρέθηκαν οι πρώτες γραφές που αφορούσαν τεχνικές διοίκησης του κράτους και στην αρχαία Ελλάδα υπήρχε καταμερισμός των εργασιών με το πόστο και τα καθήκοντα του κάθε εργάτη σαφώς ορισμένα. Σε όλες τις παραπάνω περιπτώσεις υπήρχαν άνθρωποι που αναλάμβαναν τη διεύθυνση, την καθοδήγηση και τις αποφάσεις. Η σημαντική ανάπτυξη των μεθόδων οργάνωσης και διοίκησης από εκείνη την εποχή μέχρι σήμερα απαίτησε την εξειδίκευση στελεχών που θα αναλάβουν αυτό το καθήκον.

Μετά το 1950 η επιστήμη της διοίκησης αναπτύχθηκε σε πανεπιστημιακό επίπεδο. Η ραγδαία ανάπτυξη της τεχνολογίας και ιδιαίτερα της πληροφορικής υπήρξε το κλειδί για αυτή την εξέλιξη. Τα στελέχη που θα διοικήσουν έχουν ανάγκη από διαρκή και έγκυρη πληροφόρηση, καθώς η αγορά αλλάζει με δραματικούς ρυθμούς λόγω της παγκοσμιοποίησης.

Το πιο εξελιγμένο «όπλο» των στελεχών της διοίκησης είναι τα συστήματα μηχανικής μάθησης που αναπτύχθηκαν στα πλαίσια της Τεχνητής Νοημοσύνης και της Βελτιστοποίησης. Τα συστήματα αυτά μπορούν να επιλύσουν προβλήματα που αντιμετωπίζει συχνά η διοίκηση όπως διαχείριση αποθεμάτων, πόρων, πρόβλεψης τάσεων στην αγορά, δρομολόγησης οχημάτων και πολλά άλλα.

1-Βασικές έννοιες

1.1. Διοικητική μέριμνα (Logistics Support)

Η διοικητική μέριμνα συναντάται στη βιβλιογραφία με πολλούς διαφορετικούς όρους όπως επιμελητεία, διαχειριστική ή απλώς διοίκηση επιχειρήσεων. Στην αγγλική γλώσσα με τον όρο logistics εκφράζονται οι επιμέρους λειτουργίες της διοίκησης. Οι επιμέρους λειτουργίες της διοικητικής μέριμνας είναι πολυάριθμες. Ενδεικτικά οι κυριότερες λειτουργίες αφορούν διαχείριση προσωπικού, πόρων, πολλαπλών προμηθευτών, πολλαπλών σημείων παράδοσης, ανάλυση κινδύνων, πρόβλεψη τάσεων της αγοράς.

Σκοπός της διοίκησης είναι η βελτίωση μιας υπάρχουσας κατάστασης/επιχείρησης, η οποία δεν κρίνεται ικανοποιητική ή η δημιουργία μιας ολοκληρωτικά νέας. Η δημιουργία της νέας κατάστασης/επιχείρησης ή η αναβάθμιση της προηγούμενης πραγματοποιείτε με την εφαρμογή γνώσεων, δεξιοτήτων, εργαλείων και τεχνικών. Όλοι οι εμπλεκόμενοι στην διοίκηση έχουν στόχο την ικανοποίηση των απαιτήσεων και των προσδοκιών τους. Οι εμπλεκόμενοι στην διοίκηση έχουν συχνά διαφορετικές αντιλήψεις και αντικρουόμενους στόχους. Παρά τις διαφορετικές τους βλέψεις σε σχέση με την επιχείρηση, οι εμπλεκόμενοι αποτελούν προϋπόθεση για να υπάρξει επιχείρηση.

Οι πόροι(resources) που είναι διαθέσιμοι σε κάθε επιχείρηση κατατάσσονται στις κατηγορίες:

- Ανθρώπινοι πόροι (Human Resources)
- Εξοπλισμός (equipment), μηχανές (machines), εργαλεία (tools)
- Αναλώσιμοι πόροι (consumable resources), συμπεριλαμβανομένων των πρώτων υλών, των ενδυμάτων των εργαζομένων, των υλικών (materials) κ.α.

Η μαεστρία της διοίκησης έγκειται στην βέλτιστη επιτρεπόμενη διαχείριση των πόρων ώστε να υπάρξει η μέγιστη δυνατή ικανοποίηση των στόχων των εμπλεκόμενων μερών. Τα προβλήματα που καλείται να αντιμετωπίσει η διοίκηση χαρακτηρίζονται από επικινδυνότητα και αβεβαιότητα. Η οργάνωση της διοίκησης οφείλει να περιλαμβάνει σχέδιο πρόβλεψης και αποφυγής προβλημάτων ή περιορισμού των συνεπειών τους όταν αυτές εμφανιστούν.

Η διαχείριση κόστους, πόρων, προμηθειών, ποιότητας και κινδύνου είναι ένα μέρος των μεταβλητών που λαμβάνει υπόψιν η διοίκηση ώστε να οργανώσει το στρατηγικό της σχέδιο. Οι νέες τεχνολογίες προσφέρουν τεράστιο όγκο πληροφοριών που αποτελούν πολύτιμο εργαλείο για την διοικητική επιστήμη. Ο όγκος αυτών των πληροφοριών είναι μη διαχειρίσιμος χωρίς την βοήθεια μεθευρετικών και εξελικτικών αλγορίθμων που αναπτύχθηκαν στα πλαίσια της βελτιστοποίησης και της τεχνητής νοημοσύνης.

1.2. Βελτιστοποίηση (Optimization)

Βελτιστοποίηση είναι η διαδικασία υπολογισμού του βέλτιστου αποτελέσματος κάτω από περιορισμούς. Σε κάθε στάδιο του επιχειρησιακού σχεδίου οι αποφασίζοντες καλούνται να αντιμετωπίσουν τεχνολογικά ή διοικητικά προβλήματα. Στόχος των αποφάσεων είναι είτε η ελαχιστοποίηση του κόστους είτε η μεγιστοποίηση του κέρδους. Η βελτιστοποίηση μπορεί να καθοριστεί ως διαδικασία εύρεσης των συνθηκών που δίνουν τη μέγιστη ή την ελάχιστη τιμή μιας συνάρτησης κόστους με καθορισμένες μεταβλητές απόφασης. Οι τεχνικές μαθηματικού προγραμματισμού είναι οι μέθοδοι αναζήτησης του βέλτιστου/ελαχίστου και αποτελούν μέρος της επιχειρησιακής έρευνας.

Ένα πρόβλημα βελτιστοποίησης περιλαμβάνει τα ακόλουθα σύνολα στοιχείων:

1. Μεταβλητές απόφασης (decision variables). Οι μεταβλητές απόφασης είναι οι άγνωστοι που πρέπει να καθοριστούν από την επίλυση του προβλήματος. Οι παράμετροι αποτελούν τις μεταβλητές ελέγχου του συστήματος.
2. Περιορισμοί (constraints). Οι περιορισμοί του συστήματος περιορίζουν τις μεταβλητές αποφάσεις στις επιτρεπτές τιμές.
3. Αντικειμενική συνάρτηση (objective function). Η αντικειμενική συνάρτηση είναι το μέτρο αποτελεσματικότητας του συστήματος, ως

μαθηματική συνάρτηση των μεταβλητών απόφασης και των παραμέτρων τους.

Η βέλτιστη λύση ενός μοντέλου επιτυγχάνεται όταν οι αντίστοιχες τιμές των μεταβλητών απόφασης οδηγούν στην βέλτιστη τιμή της αντικειμενικής συνάρτησης, ικανοποιώντας ταυτόχρονα όλους τους περιορισμούς.

<u>Κατηγορίες προβλημάτων βελτιστοποίησης [23]:</u>

Κατά τον τύπο των μεταβλητών

- Συνεχούς βελτιστοποίησης (continuous optimization problem). Οι μεταβλητές του προβλήματος βελτιστοποίησης είναι συνεχείς.
- Ακέραιας ή διακριτής βελτιστοποίησης (integer/discrete optimization problem). Οι μεταβλητές του προβλήματος είναι ακέραιες.
- Μεικτής ακέραιας βελτιστοποίησης (mixed integer optimization problem). Ένα πρόβλημα με συνεχείς και ακέραιες μεταβλητές.
- Συνδυαστικής βελτιστοποίησης (combinatorial optimization problem). Πρόβλημα ακέραιας βελτιστοποίησης όπου η μοντελοποίηση και οι λύσεις αντιστοιχούν σε μοντέλα δικτύου που περιγράφονται από ένα γράφημα.

Κατά το βαθμό της γραμμικότητας της αντικειμενικής συνάρτησης

- Γραμμικά προβλήματα βελτιστοποίησης (linear programming problem). Η αντικειμενική συνάρτηση και οι περιορισμοί είναι γραμμικοί.
- Μη γραμμικά προβλήματα βελτιστοποίησης (nonlinear programming problem). Η αντικειμενική συνάρτηση ή/και οι περιορισμοί είναι μη γραμμικοί.
- Προβλήματα τετραγωνικού προγραμματισμού (quadric optimization problem). Πρόβλημα μη γραμμικού περιορισμού με τετραγωνική αντικειμενική συνάρτηση και γραμμικούς περιορισμούς.

Ύπαρξη ή όχι περιορισμών ισότητας ή/και ανισότητας

- Πρόβλημα βελτιστοποίησης χωρίς περιορισμούς (unconstrained optimization problem).
- Πρόβλημα βελτιστοποίησης με περιορισμούς (constrained optimization problem).

Πλήθος λύσεων

- Μία λύση του προβλήματος (unimodal)
- Πολλαπλές λύσεις (multimodal)

Πλήθος κριτηρίων-στόχων

- Πρόβλημα με ένα κριτήριο (single objective optimization problem)
- Πρόβλημα με πολλαπλά κριτήρια (multi-objective optimization problem)

Ύπαρξη πιθανολογικών μεταβλητών

- Πρόβλημα στοχαστικής βελτιστοποίησης (stochastic optimization problem). Το πρόβλημα έχει μία ή παραπάνω πιθανολογικές (μη αιτιοκρατικές ή στοχαστικές) μεταβλητές.

Μεταβολή αντικειμενικής συνάρτησης ή περιορισμών στο χρόνο

- Πρόβλημα δυναμικής βελτιστοποίησης(dynamic optimization problem). Η αντικειμενική συνάρτηση ή/και οι περιορισμοί του προβλήματος μεταβάλλονται στο χρόνο.

Το γενικευμένο πρόβλημα βελτιστοποίησης με περιορισμούς μοντελοποιείται ως εξής:

$$\min f(x) \tag{1.1}$$

υπό

$$h_j(x) = 0, j = 1, \ldots, p \tag{1.2}$$

$$g_k(x) \leq 0, k = 1, \dots, l \qquad (1.3)$$

$$x_i, i = 1, \dots, n \in dom(x_i) \qquad (1.4)$$

όπου f είναι η αντικειμενική συνάρτηση του προβλήματος, x είναι οι μεταβλητές, $dom(x)$ είναι το πεδίο τιμών της κάθε μεταβλητής, h_j είναι οι p περιορισμοί ισότητας και g_k είναι οι l περιορισμοί ανισότητας.

Όπως αναφέρθηκε στην παράγραφο 1.1. οι αποφασίζοντες έχουν συχνά αντικρουόμενους στόχους. Στην μαθηματική αποτύπωση του προβλήματος οι πολλαπλοί στόχοι εκφράζονται ως πολλαπλές αντικειμενικές συναρτήσεις. Το πρόβλημα που προκύπτει είναι η εύρεση εκείνου του μοντέλου που ικανοποιεί όλες τις αντικρουόμενες συναρτήσεις. Αυτό το πρόβλημα ονομάζεται πρόβλημα βελτιστοποίησης πολλαπλών κριτηρίων.

Το πρόβλημα βελτιστοποίησης πολλαπλών κριτηρίων μοντελοποιείται ως ακολούθως:

$$min\, f(x) = [f_1(x), f_2(x), \dots, f_m(x)] \qquad (1.5)$$

υπό

$$h_j(x) = 0, j = 1, \dots, p \qquad (1.6)$$

$$g_k(x) \leq 0, k = 1, \dots, l \qquad (1.7)$$

όπου $x = [x_1, x_2, \dots, x_n]$ το διάνυσμα των μεταβλητών απόφασης, $f_i, i = 1, \dots, m$ οι αντικειμενικές συναρτήσεις του προβλήματος, $h_j, j = 1, \dots p$ είναι οι p περιορισμοί ισότητας και $g_k, k = 1, \dots, l$ είναι οι l περιορισμοί ανισότητας.

Η Δυναμική βελτιστοποίηση αποτελείται από προβλήματα που τα δεδομένα τους αλλάζουν μέσα στο χρόνο, δηλαδή ο χρόνος εκφράζεται ως μια επιπλέον μεταβλητή.

Παρακάτω η μοντελοποίησης ενός γενικευμένου προβλήματος δυναμικής βελτιστοποίησης:

$$\max f(X, t) \tag{1.8}$$

υπό

$$h_j(X, t) = 0, j = 1, \dots, p \tag{1.9}$$

$$g_k(X, t) \leq 0, k = 1, \dots, l \tag{1.10}$$

$$X = [x_1, x_2, \dots, x_n]^T \tag{1.11}$$

όπου f είναι η αντικειμενική συνάρτηση του προβλήματος, X το διάνυσμα των μεταβλητών απόφασης, t το χρονικό διάστημα που θα εξεταστεί, h_j οι περιορισμοί ισότητας και g_k οι περιορισμοί ανισότητας.

1.3. Επιχειρησιακή έρευνα (Operational Research)

Η επιχειρησιακή έρευνα είναι ο κλάδος των μαθηματικών που ασχολείται με την εφαρμογή επιστημονικών μεθόδων και τεχνικών σε προβλήματα λήψης αποφάσεων με στόχο την εύρεση και την εφαρμογή της καλύτερης λύσης. Το πεδίο της επιχειρησιακής έρευνας και της διοικητικής επιστήμης (management science) ασχολείται με την ανάπτυξη και την εφαρμογή ποσοτικών τεχνικών για την επίλυση προβλημάτων σε διάφορους τομείς.

Ο στρατός, στους χώρους του οποίου αναπτύχθηκε η επιχειρησιακή έρευνα, αντιμετωπίζει καθημερινά θέματα διαχείρισης αποθεμάτων και εφοδιαστικής αλυσίδας. Τα ίδια θέματα μπορεί να αντιμετωπίσει μια μονάδα παραγωγής τροφίμων ή μια βιομηχανία. Η τροχαία έχει ανάγκη πρόβλεψης της κίνησης των δρόμων, η πυροσβεστική πρόβλεψη των περιοχών υψηλού κινδύνου πυρκαγιάς και η πολιτική προστασία πρόβλεψη φυσικών καταστροφών και αντιμετώπιση των κρίσεων. Όλες οι παραπάνω υπηρεσίες έχουν ανάγκη από την επίλυση προβλημάτων με διαφορετικούς συντελεστές και αντικρουόμενα κριτήρια. Ο τρόπος μοντελοποίησης και επίλυσης αυτών των προβλημάτων παραμένει ο ίδιος και αποτελεί το αντικείμενο της επιχειρησιακής έρευνας.

Ο τρόπος επίλυσης περιλαμβάνει την συγκέντρωση πληροφοριών, την επεξεργασία τους και την εξαγωγή συμπερασμάτων, τον ορισμό παραμέτρων και περιορισμών. Από τα συμπεράσματα προκύπτει εύρεση μίας τυχαίας λύσης, στην συνέχεια θα εξεταστεί αν η λύση είναι βέλτιστη ή αν υπάρχει περιθώριο βελτίωσης. Θα υπολογιστεί η βέλτιστη λύση ώστε το κέρδος να υπερισχύσει του κόστους. Ο βασικός παράγοντας που θα καθορίσει την εγκυρότητα των προτεινόμενων λύσεων είναι η σωστή πληροφόρηση και εδώ ξεκινά η μελέτη των εφαρμογών της Τεχνητής Νοημοσύνης.

1.4. Τεχνητή Νοημοσύνη (Artificial Intelligence-AI)

Η τεχνητή νοημοσύνη είναι η επιστήμη του κλάδου της πληροφορικής που δημιουργεί συστήματα τα οποία έχουν τη δυνατότητα να μιμούνται ανθρώπινες συμπεριφορές. Παραδείγματα τέτοιων συμπεριφορών είναι η αναγνώριση και αναπαραγωγή φωνής, η επεξεργασία πληροφοριών και η μηχανική μάθηση. Θα δημιουργηθεί δηλαδή ένα τεχνητό σύστημα που θα έχει τη δυνατότητα να αντιμετωπίσει προβλήματα που μέχρι πρότινος μόνο ένας άνθρωπος θα μπορούσε να επιλύσει.

Ο υπολογιστής που θα προγραμματιστεί να συμπεριφερθεί σαν άνθρωπος έχει ένα προτέρημα. Ο υπολογιστής θα λειτουργήσει πάντα ορθολογικά, δηλαδή θα κάνει το σωστό με βάση τον προγραμματισμό του, ενώ ένας άνθρωπος δεν θα κάνει πάντα το σωστό. Αν ο υπολογιστής κάνει λάθος, υπάρχει σφάλμα στον προγραμματισμό του και όχι στην εκτέλεση.

Η πρώτη εφαρμογή της τεχνητής νοημοσύνης έγινε το 1943 από τους McCulloch και Pitts [1] και βασίστηκε στη λειτουργία των νευρώνων του εγκεφάλου. Οι επιστήμονες εμπνεύστηκαν από τον τρόπο που οι νευρώνες μεταφέρουν και επεξεργάζονται την πληροφορία από τον εγκέφαλο στο υπόλοιπο σώμα και δημιούργησαν ένα σύστημα τεχνητών νευρώνων όπου θα έχει τις ίδιες ιδιότητες σε έναν υπολογιστή. Περισσότερα για τη λειτουργία των τεχνητών νευρωνικών δικτύων θα αναλυθούν στην ενότητα 3.1.

Κάποιες εφαρμογές της τεχνητής νοημοσύνης που αποτελούν εργαλεία της διοικητικής μέριμνας είναι οι εικονικοί βοηθοί, η αναγνώριση ομιλίας και προσώπου, οι μηχανές αναζήτησης κ.α.

1.5. Μηχανική Μάθηση (Machine Learning)

Με τον όρο μηχανική μάθηση (machine learning) περιγράφεται ένα σύστημα που έχει τη δυνατότητα να «μαθαίνει» από τα δεδομένα με τρόπο παρόμοιο με τον άνθρωπο και στη συνέχεια να προβλέπει μελλοντικές καταστάσεις. Η μηχανική μάθηση μπορεί να είναι επιβλεπόμενη (supervised learning) ή μη επιβλεπόμενη (unsupervised learning).

Στην επιβλεπόμενη μάθηση το σύστημα δέχεται ως είσοδο(input) δεδομένα και αποτελέσματα και στόχος του(output) είναι να βρει τη σχέση που συνδέει τα δύο μέρη. Ο υπολογιστής θα εκπαιδευτεί βρίσκοντας τη σχέση που συνδέει τα δεδομένα εισόδου και εξόδου. Μόλις υπολογιστεί αυτή η σύνδεση, όταν ο υπολογιστής λάβει παρόμοια δεδομένα εισόδου θα προβλέψει τα δεδομένα εξόδου.

Στην μη επιβλεπόμενη μάθηση το σύστημα δέχεται ως είσοδο δεδομένα που ίσως να μην είναι κατηγοριοποιημένα και στόχος του είναι να βρει τη κατηγορία που ανήκουν ή το μοτίβο που ακολουθούν(pattern). Δηλαδή ο υπολογιστής ομαδοποιεί διάσπαρτα δεδομένα, αναγνωρίζοντας σχέσεις μεταξύ τους. Υποπεδίο της μη επιβλεπόμενης μάθησης είναι η εξόρυξη δεδομένων (data mining).

Μέσω της μηχανικής μάθησης δίνεται η δυνατότητα ανάλυσης των πληροφοριών και των υπαρχόντων δεδομένων από παρόμοιες καταστάσεις που έχουν συμβεί στο παρελθόν. Στην μηχανική μάθηση βασίζονται οι μηχανές διανυσμάτων υποστήριξης (ενότητα 3.2) και τα τεχνητά νευρωνικά δίκτυα (ενότητα 3.1). Με αυτά τα συμπεράσματα ο επιχειρησιακός σχεδιασμός θα είναι πιο έγκυρος αφού βασίζεται σε ρεαλιστικά δεδομένα.

1.6. Εξόρυξη δεδομένων (Data Mining)

Ως εξόρυξη δεδομένων ορίζεται η διαδικασία της διερεύνησης και της ανάλυσης μεγάλης ποσότητας δεδομένων (data sets) με σκοπό την ανακάλυψη σχέσεων και δομών μεταξύ τους που δεν είναι εμφανείς.

Ο όγκος των διαθέσιμων δεδομένων σήμερα είναι τεράστιος. Τα δεδομένα αυτά εισάγονται σε ηλεκτρονικές αποθήκες και αποτελούν το υλικό μελέτης πολλών επιστημών. Η ηλεκτρονική τους μορφή τα κάνει εύκολα προσβάσιμα και διαχειρίσιμα.

Πίνακας 1. Παραδείγματα όγκου δεδομένων

	Όγκος δεδομένων
Ιστοσελίδες στο διαδίκτυο	50.000.000.000
Χρήστες μέσων κοινωνικής δικτύωσης	2.000.000.000
Συστατικά ανθρώπινου γονιδιώματος	310^9

Η συλλογή δεδομένων σχετικά με την επισκεψιμότητα ιστοσελίδων, οδηγεί σε στοχευμένες διαφημίσεις. Η συλλογή δεδομένων από τη λίστα επαφών του κινητού τηλεφώνου προτείνει τα «άτομα που ίσως γνωρίζεται» στα κοινωνικά δίκτυα. Η συλλογή των δεδομένων γίνεται μέσω των cookies. Η ανάλυση των δεδομένων των συστατικών του ανθρώπινου γονιδιώματος προβλέπει την τάση εκδήλωσης κάποιας πάθησης ή βρίσκει κοινά χαρακτηριστικά σε ανθρώπους που εκδήλωσαν την ίδια πάθηση.

Ο πιο γνωστός αλγόριθμος εξόρυξης δεδομένων ονομάζεται k-means και θα παρουσιαστεί στην ενότητα 3.3.

2-Αλγόριθμοι Βελτιστοποίησης

2.1. Εύρεση συντομότερης διαδρομής

Το βασικότερο πρόβλημα στα δίκτυα είναι η εύρεση της συντομότερης διαδρομής. Οι βασικότεροι αλγόριθμοι επίλυσης προβλημάτων εύρεσης της συντομότερης διαδρομής είναι οι Dijkstra και Bellman-Ford. Με αυτούς τους αλγορίθμους μπορούν να επιλυθούν προβλήματα δρομολόγησης οχημάτων, γραμμών ηλεκτροδότησης, σωληνώσεων κ.α. Τα προβλήματα εύρεσης της συντομότερης διαδρομής εξετάζουν τα κόστη (μήκη) των τόξων. Ζητούμενο είναι η συντομότερη διαδρομή που περνά από n σταθμούς. Η κάθε διαδρομή μεταξύ δύο διαδοχικών σταθμών έχει διαφορετικό μήκος/κόστος.

<u>Γενική περιγραφή Dijkstra</u>

Ο αλγόριθμος Dijkstra [2] δημιουργήθηκε το 1956 και ευρίσκει την συντομότερη διαδρομή μεταξύ κορυφών ενός μη κατευθυνόμενου γραφήματος. Ανήκει στην κατηγορία των άπληστων αλγορίθμων, δηλαδή σε κάθε βήμα κάνει την βέλτιστη επιλογή.

Δημιουργείτε ένας πίνακας όπου θα καταγράφεται η συντομότερη διαδρομή από τους ελεγμένους σταθμούς και ένας πίνακας όπου καταγράφει τους ελεγμένους σταθμούς. Οι πίνακες θα ανανεώνονται σε κάθε επανάληψη. Ο πίνακας αποστάσεων αρχικοποιείται με ∞. Έστω το δίκτυο G και οι κορυφές του V, με S την συνολική απόσταση.

Βήματα αλγορίθμου

- Επιλέγεται τυχαία ένας σταθμός
- Καταγράφεται η απόσταση αυτού του σταθμού με τους γειτονικούς του στον πίνακα διαδρομών
- Ο σταθμός καταγράφεται στον πίνακα με τους ελεγμένους σταθμούς.
- Επιλέγεται ο γειτονικός σταθμός με τη μικρότερη απόσταση και επαναλαμβάνεται η διαδικασία

Γενική περιγραφή Bellman-Ford

Ο αλγόριθμος Bellman-Ford [3] [4] δημιουργήθηκε από τον Shimbel [5] το 1955 και δημοσιεύτηκε από τους Bellman και Ford το 1958 και το 1956 αντίστοιχα. Ο Bellman-Ford είναι πιο αργός από τον Dijkstra αλλά πιο ευέλικτος καθώς μπορεί να επιλύσει προβλήματα και με αρνητικά βάρη. Εφαρμόζεται σε κατευθυνόμενα γραφήματα.

2.2. Πρόβλημα μέγιστης ροής.

Συμπληρωματικό του προβλήματος εύρεσης συντομότερης διαδρομής, καθώς μπορεί να θεωρηθεί υποπρόβλημα της ίδιας κατάστασης. Τα προβλήματα μέγιστης ροής εξετάζουν τις χωρητικότητες των τόξων. Ο βασικότερος αλγόριθμος επίλυσης αυτών των προβλημάτων είναι ο αλγόριθμος Ford-Fulkerson [6]. Με τον αλγόριθμο Ford-Fulkerson λύνονται προβλήματα ροής ρευστών σε σωλήνες, ρεύματος σε δίκτυο, εφοδιασμού κ.α. Τα γραφήματα αυτά είναι προσανατολισμένα και υπάρχουν περιορισμοί στις χωρητικότητες.

Τα γραφήματα που αντιπροσωπεύουν προβλήματα μέγιστης ροής είναι κατευθυνόμενα. Στο δίκτυο G=(V, E) οι χωρητικότητες c είναι μη αρνητικές, η πηγή ορίζεται ως s και το πέρας ως t. Η ροή από τον κόμβο u στον κόμβο v ορίζεται ως f(u, v) και η χωρητικότητα c(u, v).

Γενική περιγραφή αλγορίθμου Ford-Fulkerson

Ο αλγόριθμος Ford-Fulkerson δημιουργήθηκε το 1956 και είναι ένας άπληστος αλγόριθμος επίλυσης προβλημάτων μέγιστης ροής.

Έστω το δίκτυο G(V, E), η πηγή s και το πέρας t, με f(u, v) η ροή και c(u, v) η χωρητικότητα.

Αν η ροή f μεταξύ δύο κορυφών είναι μικρότερη από την χωρητικότητα c, τότε η διαφορά τους c_f ονομάζεται εναπομείνασα χωρητικότητα.

$$c_f(u, v) = c(u, v) - f(u, v) \qquad (2.1)$$

Σε μία διαδρομή η μικρότερη εναπομείνασα χωρητικότητα μεταξύ όλων των ακμών ορίζεται ως η εναπομείνασα χωρητικότητα όλης της διαδρομής από την αρχή ως το πέρας.

$$c_{f,total}(s,t) = min\{c(s,a) - f(s,a), c(a,b) - f(a,b), \dots, c(z,t) - f(z,t)\} \quad (2.2)$$

2.3. Πρόβλημα του ελάχιστου τανύοντος δέντρου (Minimal Spanning Tree - MST)

Το πρόβλημα της εύρεσης ενός ελάχιστου τανύοντος (επικαλύπτον) δέντρου είναι η διαδικασία εύρεσης ενός υποσυνόλου T των ακμών ενός γραφήματος G, που να συνδέει όλες τις κορυφές του G, με το ελάχιστο δυνατό κόστος. Οι βασικότεροι αλγόριθμοι επίλυσης αυτού του προβλήματος είναι οι Prim και Kruskall.

<u>Γενική περιγραφή αλγορίθμου Kruskall</u>

Ο αλγόριθμος του Kruskall [7] παρουσιάστηκε το 1956. Είναι ένας άπληστος αλγόριθμος που σε κάθε βήμα επιλέγει την ακμή με το μικρότερο κόστος. Ο αλγόριθμος συνεχίζεται εφόσον δεν δημιουργούνται «κύκλοι» και τερματίζει όταν το δέντρο επικαλύψει όλες τις κορυφές.

Οι ακμές κατατάσσονται κατά αύξουσα σειρά και επιλέγεται εκείνη με το μικρότερο κόστος. Οι κόμβοι μεταξύ των οποίων υπάρχει η ακμή, μεταφέρονται στη λίστα με τους ελεγμένους. Στο επόμενο βήμα ελέγχεται ποια από τις ακμές που συνδέονται με τους ήδη ελεγμένους κόμβους έχει το μικρότερο κόστος και επιλέγετε. Η διαδικασία επαναλαμβάνεται για όλες τις ακμές εφόσον δεν δημιουργείτε κύκλος.

<u>Γενική περιγραφή αλγορίθμου Prim</u>

Ο αλγόριθμος Prim [8] δημοσιεύτηκε το 1957 και είναι παρεμφερής με τον Kruskall.

Ο αλγόριθμος Prim λειτουργεί σε τρόπο παρόμοιο με τον Kruskall, με την διαφορά ότι ξεκινά από μία τυχαία κορυφή και αναπτύσσει το δέντρο από εκείνη.

Έστω ότι ο αλγόριθμος ξεκινά από την κορυφή Α. Η κορυφή Α συνδέεται με τις Β, Γ. Από αυτές τις 2 επιλέγεται εκείνη που η ακμή της έχει το μικρότερο βάρος/κόστος. Η διαδικασία επαναλαμβάνεται για την κορυφή που επιλέχθηκε. Ο αλγόριθμος περατώνεται όταν ελεγχθούν όλες οι κορυφές.

2.4. Απλοί ευρετικοί αλγόριθμοι

Σε πραγματικές συνθήκες ένα πρόβλημα συνδυαστικής βελτιστοποίησης είναι τόσο μεγάλο, που ο υπολογισμός της βέλτιστης λύσης σε περιορισμένο χρόνο είναι πρακτικά αδύνατος. Τα προβλήματα της διοίκησης είναι κατηγορίας NP-hard, μη πολυωνυμικά δύσκολα, δηλαδή δεν είναι γνωστός κάποιος αλγόριθμος επίλυσης και δεν είναι δυνατόν να υπάρξει τέτοιος. Η επίλυση αυτών των προβλημάτων γίνεται με τεχνικές που οδηγούν σε μία ικανοποιητική λύση. Στη συνέχεια ελέγχεται η απόκλιση της ικανοποιητικής λύσης από τη βέλτιστη, η ευκολία απόκτησης μίας λύσης αλλά και ο χρόνος επίλυσης. Η βέλτιστη λύση του προβλήματος μπορεί να δοθεί από πολλούς ευρετικούς αλγορίθμους.

Η επαλήθευση της λύσης που θα υποδείξει ο αλγόριθμος μπορεί να ελεγχθεί με τον εξής απλό τρόπο. Επιλύεται από τον αλγόριθμο ένα απλό πρόβλημα του οποίου η βέλτιστη λύση είναι εύκολο να υπολογιστεί με αναλυτικό τρόπο. Στη συνέχεια τα αποτελέσματα του αλγορίθμου και της αναλυτικής επίλυσης συγκρίνονται. Όσο μικρότερη η απόκλιση μεταξύ των δύο μεθόδων, τόσο αποτελεσματικότερος είναι ο αλγόριθμος. Ένας άλλος τρόπος ελέγχου του αλγορίθμου είναι η δημιουργία ενός φράγματος αποδεκτής λύσης όπως με την διαδικασία διακλάδωσης και οριοθέτησης (branch and bound) [9]. Με αυτό τον τρόπο επιλύεται ένα χαλαρωμένο πρόβλημα του οποίου οι τιμές αφού δεν παραβιάζουν το φράγμα θα είναι ικανοποιητικές.

2.5. Αλγόριθμος τοπικής αναζήτησης

Ο πιο απλός και ίσως ο πρώτος ιστορικά αλγόριθμος βελτιστοποίησης είναι ο αλγόριθμος τοπικής αναζήτησης. Επιλέγεται τυχαία μια εφικτή λύση και συγκρίνεται με μία άλλη τυχαία εφικτή λύση. Αν η δεύτερη λύση είναι βέλτιστη, ορίζεται ως αρχική και η διαδικασία επαναλαμβάνεται μέχρι να βρεθεί η βέλτιστη εφικτή λύση που υπάρχει στο εξεταζόμενο σύνολο. Ο αλγόριθμος αυτός δυστυχώς θα εγκλωβιστεί στο πρώτο τοπικό βέλτιστο καθώς θα τερματίσει στην πρώτη λύση που είναι βέλτιστη από την προηγούμενη. Παρακάτω θα παρουσιαστούν βελτιώσεις αυτού του αλγορίθμου που ξεπερνούν αυτό το πρόβλημα,

Ψευδοκώδικας

INSERT S % S τυχαία εφικτή λύση %

FIND NEW_S % βρίσκεται νέα τυχαία λύση %

WHILE NEW_S<S %αν η νέα λύση είναι καλύτερη από την προηγούμενη%

{

S=NEW_S %η λύση ορίζεται ως αρχική%

FIND NEW_S % βρίσκεται νέα λύση και συγκρίνεται με την προηγούμενη%

}

RETURN NEW_S

2.6. Προσομοιωμένη ανόπτηση (Simulated Annealing, SA)

Ο αλγόριθμος προσομοιωμένης ανόπτησης αναπτύχθηκε από τον Kirkpatrick to 1982 [10]. Η ανόπτηση είναι μία θερμική κατεργασία των μετάλλων κατά την οποία ένα υλικό θερμαίνεται μέχρι το σημείο τήξης του, υπόκειται σε κατεργασία και στη συνέχεια ψύχεται αργά μέχρι να φτάσει στην θερμοκρασία δωματίου. Η αργή ψύξη προσδίδει στο κατεργασμένο μέταλλο ευκαμψία και το κάνει πιο εύχρηστο.

Αν η διαδικασία της αργής ψύξης θεωρηθεί ως μεταβολή της ενέργειας και η τελική κατάσταση θερμοκρασίας δωματίου θεωρηθεί ως το σημείο ισορροπίας τότε γίνεται κατανοητό πως ο αλγόριθμος της προσομοιωμένης ανόπτησης μιμείται τις ενεργειακές αλλαγές ενός συστήματος μέχρι αυτό να καταλήξει στο σημείο ισορροπίας. Οι ενεργειακές αλλαγές αντιστοιχούν στις εφικτές λύσεις και το σημείο ισορροπίας στην βέλτιστη λύση. Επειδή υπάρχει ο κίνδυνος εγκλωβισμού σε τοπικά ακρότατα, ο αλγόριθμος αποδέχεται κάποιες κακές λύσεις με πιθανότητα.

<u>Γενική περιγραφή αλγορίθμου Προσομοιωμένης Ανόπτησης</u>

Η πιθανότητα αποδοχής μίας κακής λύσης δίνεται από τον τύπο:

$$P_{ac} = e^{-\frac{f(s')-f(s)}{T}} \qquad (2.3)$$

όπου, s η αρχική κατάσταση, s’ η τρέχουσα κατάσταση, f(x) η αντικειμενική συνάρτηση, T η θερμοκρασία. Αν $f(s') - f(s) < 0$, δηλαδή η νέα λύση είναι καλύτερη από την αρχική, η νέα λύση γίνεται πάντα αποδεκτή, δηλαδή $P_{ac} = 1$, αλλιώς η πιθανότητα αποδοχής δίνεται από τον τύπο 2.3.

Η μείωση της θερμοκρασίας δίνεται από την συνάρτηση $a(t)$ και η επιλογή της αρχικής θερμοκρασίας είναι πολύ σημαντική. Υπερβολικά μεγάλη αρχική θερμοκρασία θα προκαλέσει αποδοχή πολλών λύσεων και θα μετατρέψει την αναζήτηση σε τυχαία. Από την άλλη πολύ μικρή αρχική θερμοκρασία δεν επιτρέπει αποδοχή λύσεων και η αναζήτηση μετατρέπεται σε τοπική μίας κίνησης.

Ο αλγόριθμος τερματίζεται όταν η θερμοκρασία φτάσει στην τελική επιτρεπόμενη τιμή(σημείο ισορροπίας) ή εξαντληθούν οι επιτρεπτές επαναλήψεις.

<u>Ψευδοκώδικας Προσομοιωμένης Ανόπτησης</u>

INSERT s_0 t_0 #επιλογή αρχικής λύσης και αρχικής θερμοκρασίας#

INSERT t_0 #επιλογή αρχικής θερμοκρασίας#

INSERT t_{max} #επιλογή τελικής θερμοκρασίας/σημείου ισορροπίας#

INSERT N #επιλογή μέγιστου αριθμού επαναλήψεων#

INSERT $a(t)$ #επιλογή συνάρτησης μείωσης θερμοκρασίας#

WHILE (t< t_{max} OR n<N) {

FIND s

if $(f(s^r) - f(s) < 0)$ then

s_0=s

else

FIND x$\sim(0,1)$

```
                if (x<e^(-(f(s')-f(s))/t)) then
                        s0=s
                end if
        end if
t=a(t)
}
RETURN s0
```

2.7. Περιορισμένη αναζήτηση (Tabu Search, TS)

Ο αλγόριθμος περιορισμένης αναζήτησης αναπτύχθηκε από τον Glover το 1989 [11]. Είναι ένας μεθευρετικός αλγόριθμος αναζήτησης της βέλτιστης λύσης ανάμεσα στις εφικτές λύσεις με τρόπο παρόμοιο με εκείνον της προσομοιωμένης ανόπτησης (ενότητα 2.6). Όταν ο αλγόριθμος περιορισμένης αναζήτησης εγκλωβιστεί σε τοπικό ακρότατο, ξεφεύγει ακολουθώντας στρατηγική με μνήμη, δηλαδή αποθηκεύονται οι πιο πρόσφατες κινήσεις ώστε να μην επαναληφθούν. Η αποθήκευση των περιορισμένων κινήσεων γίνεται στη λίστα tabu, η οποία έδωσε το όνομα της στον αλγόριθμο.

Γενική περιγραφή αλγορίθμου

Επιλέγεται μια αρχική λύση s και αξιολογείτε με τη συνάρτηση κόστους $c(s)$. Ορίζεται ένα κόστος μετακίνησης από τη μία λύση στην επόμενη d και χρησιμοποιείτε η συνάρτηση $dc(s)$. Ορίζεται το μέγεθος της λίστας tabu k θέσεων. Σε κάθε επανάληψη η νέα λύση προστίθεται στην κορυφή της λίστας, τα υπόλοιπα στοιχεία μετακινούνται μία θέση κάτω και το τελευταίο αφαιρείται. Το μέγεθος της λίστας μπορεί να είναι σταθερό ή να μεταβάλλεται ανάλογα με τα δεδομένα του προβλήματος. Ο αλγόριθμος τερματίζεται όταν ολοκληρωθούν όλες οι επαναλήψεις ανεξάρτητα αν βρέθηκε η βέλτιστη λύση ή όχι.

Ψευδοκώδικας περιορισμένης αναζήτησης

FIND s %εύρεση μίας αρχικής λύσης%

c(s) %υπολογισμός του κόστους%

INSERT d %επιλογή κόστους μετακίνησης%

INSERT k %επιλογή μεγέθους λίστας tabu%

INSERT M %αριθμός επαναλήψεων%

WHILE (i<M){

    FIND s’ %%

    IF (d*c(s’) < c(s)) THEN

        s=s’

    END IF

    INSERT s’ στη λίστα tabu

}

RETURN s’

2.8. Γενετικοί αλγόριθμοι (Genetic Algorithms, GA)

Ο πρώτος των γενετικών αλγορίθμων δημιουργήθηκε από τον Holland το 1975 [12] στην προσπάθεια του να εφαρμόσει την εξελικτική διαδικασία της φύσης για την επίλυση πολύπλοκων και πολυπαραγοντικών προβλημάτων.

Η εξέλιξη στη φύση χαρακτηρίζεται από την επιβίωση των ισχυρότερων πληθυσμών κάθε είδους. Ο πληθυσμός μέσα στον οποίο αναζητείται η βέλτιστη λύση των γενετικών αλγορίθμων αποτελείται από άτομα (individuals). Η κάθε επανάληψη του αλγορίθμου ονομάζεται γενιά (generation). Η εξέλιξη είναι το αποτέλεσμα της διασταύρωσης(crossover) δύο τελεστών-γονέων (parents) και της μετάλλαξης(mutation) ενός τελεστή. Ένα απλό παράδειγμα της διασταύρωσης παρουσιάζεται στον πίνακα 2 και της μετάλλαξης στον πίνακα 3.

Πίνακας 2. Δύο γονείς με 6 μεταβλητές και οι δύο απόγονοί τους με διασταύρωση στο σημείο 4

	1ος Τελεστής	2ος Τελεστής	3ος Τελεστής	4ος Τελεστής	5ος Τελεστής	6ος Τελεστής
1ος Γονέας	0	0	1	1	0	1
2ος Γονέας	0	1	0	1	1	0
1ος Απόγονος	0	0	1	1	1	0
2ος Απόγονος	0	1	0	1	0	1

Πίνακας 3. Δύο γονείς με 6 μεταβλητές και οι δύο απόγονοί τους με διασταύρωση στο σημείο 4 και μετάλλαξη στο σημείο 1

	1ος Τελεστής Μετάλλαξη	2ος Τελεστής	3ος Τελεστής	4ος Τελεστής	5ος Τελεστής	6ος Τελεστής
1ος Γονέας	0	0	1	1	0	1

2ος Γονέας	0	1	0	1	1	0
1ος Απόγονος	1	0	1	1	1	0
2ος Απόγονος	1	1	0	1	0	1

<u>Γενική περιγραφή Γενετικού αλγορίθμου</u>

1. Κωδικοποίηση (encoding) των λύσεων με τέτοιο τρόπο ώστε να αντιστοιχούν σε χρωμοσώματα. Φυσικά ο αλγόριθμος δεν αναφέρεται μόνο σε βιολογικά χρωμοσώματα. Αν για παράδειγμα η λύση του πίνακα # για τον 2ο απόγονο 110101 αντιστοιχεί σε ένα ανταλλακτικό εξάρτημα που πωλείται σε ένα κατάστημα, το πρώτο ψηφίο αντιστοιχεί στην παρτίδα, το δεύτερο στο υλικό, κ.τ.λ. Δηλαδή το κάθε ψηφίο/χρωμόσωμα αντιστοιχεί σε ένα χαρακτηριστικό/μεταβλητή. Οι μεταβλητές του προβλήματος είναι τα χρωμοσώματα.
2. Δημιουργία συνάρτησης καταλληλότητας (fitness) του κάθε ατόμου του πληθυσμού. Η συνάρτηση καταλληλότητας μπορεί να θεωρηθεί και ως η αντικειμενική συνάρτηση αφού από αυτή θα χαρακτηριστεί βέλτιστη μία λύση. Η αντικειμενική συνάρτηση μπορεί να είναι άθροισμα επιμέρους συναρτήσεων ανάλογα με το πρόβλημα ή μπορεί να μεταβάλλεται με το χρόνο όπως σε περιπτώσεις δυναμικών προβλημάτων. Επίσης η συνάρτηση καταλληλότητας δεν αναπαριστά πάντα κόστος.
3. Επιλογή αρχικού πληθυσμού N. Ο πληθυσμός είναι οι λύσεις μέσα στις οποίες θα αναζητηθεί η βέλτιστη. Για να είναι ο αρχικός πληθυσμός ομοιόμορφα κατανεμημένος στον χώρο των λύσεων, οι τιμές επιλέγονται τυχαία. Επιπλέον ο πληθυσμός οφείλει να είναι αρκετός ώστε να καλυφθεί ολόκληρος ο χώρος των λύσεων. Το κάθε άτομο του πληθυσμού ή αλλιώς

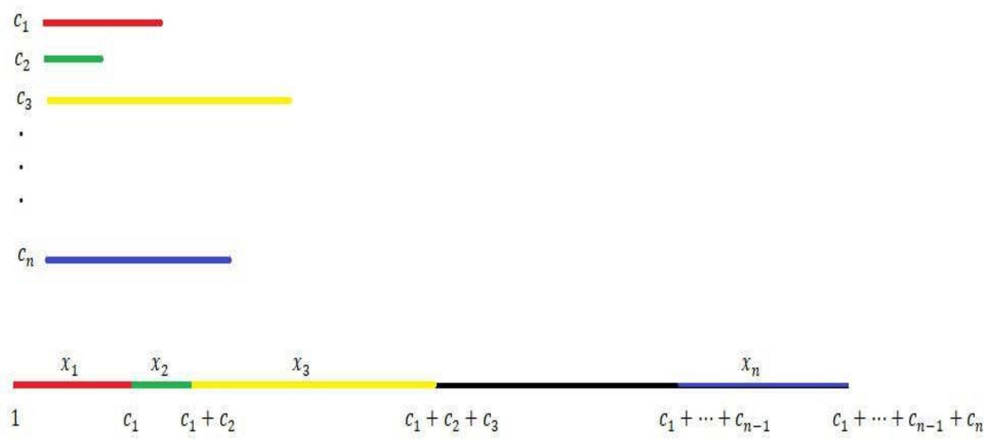
η κάθε λύση συμβολίζονται ως x_{ij} με $i = 1,2, \dots, N$ και $j = 1,2, \dots, n$, όπου n ο αριθμός των διαστάσεων και N το σύνολο του πληθυσμού. Η καταλληλότητα της κάθε λύσης θα υπολογιστεί από την συνάρτηση fitness του $2^{ου}$ σταδίου.

4. Επιλογή γονέων. Όπως συμβαίνει και στη φύση όπου ο δυνατότερος επιβιώνει έτσι και στους γενετικούς αλγορίθμους οι δυνατότεροι κάθε πληθυσμού έχουν τις μεγαλύτερες πιθανότητες να αναπαραχθούν. Δυνατότεροι θεωρούνται εκείνοι με την μεγαλύτερη αξία που δίνει η αντικειμενική συνάρτηση. Οι λιγότερο δυνατοί όμως έχουν και εκείνοι κάποια αξιόλογα χαρακτηριστικά και επίσης έχουν τη δυνατότητα να αναπαραχθούν. Οι πιο γνωστοί τρόποι επιλογής γονέων είναι οι παρακάτω:

- Ταξινόμηση κατά φθίνουσα σειρά όλων των ατόμων και ταίριασμα γονέων ανά ζεύγη ($1^{ος}$ και $2^{ος}$, $3^{ος}$ και $4^{ος}$, …. $(N\text{-}1)^{ος}$ και $N^{ος}$). Προτέρημα αυτής της μεθόδου είναι η ευκολία στον προγραμματισμό. Μειονέκτημα είναι ότι ο κάθε γονέας μπορεί να χρησιμοποιηθεί μόνο μία φορά και έτσι χάνονται αρκετοί καλοί συνδυασμοί.
- Τυχαία επιλογή γονέων(random selection). Η μέθοδος αυτή είναι επίσης απλή στην εφαρμογή της. Οι απόγονοι 2 γονέων με χαμηλή αξία είναι επίσης χαμηλής αξίας. Ζητούμενο είναι να επιβιώσουν οι δυνατότεροι άρα αυτή η μέθοδος δεν δίνει τα επιθυμητά αποτελέσματα.
- Επιλογή από ομάδες λύσεων (tournament selection). Ο πληθυσμός χωρίζεται σε ομάδες και στην κάθε ομάδα επιλέγονται τα δύο πιο ισχυρά άτομα ως γονείς.
- Ο κανόνας της ρουλέτας (roulette selection). Με αυτή τη μέθοδο η πιθανότητα επιλογής ενός ατόμου αυξάνεται ανάλογα με την αξία του. Οι αξίες των ατόμων $c_1, c_2, \dots, c_n$ αθροίζονται $\sum_{i=1}^{n} c_i = C$. Επιλέγεται μία τυχαία τιμή m από 1 μέχρι C. Το άτομο που επιλέγεται είναι:

$$\begin{cases} x_1, & \text{αν } 1 < m \leq c_1 \\ x_2, & \text{αν } c_1 + 1 < m \leq c_1 + c_2 \\ x_3, & \text{αν } c_1 + c_2 + 1 < m \leq c_1 + c_2 + c_3 \\ & \vdots \\ x_n, & \text{αν } c_1 + \cdots + c_{n-1} + 1 < m \leq c_1 + \cdots + c_{n-1} + c_n \end{cases}$$

Ένα παράδειγμα για να γίνει κατανοητή η μέθοδος παρουσιάζεται στο σχήμα 1.



Σχήμα 1. Επιλογή γονέων με τη μέθοδο της ρουλέτας.

Η πιθανότητα επιλογής του κάθε ατόμου με αυτή τη μέθοδο είναι:

$$p_i = \frac{c_i}{\Sigma_{j=1}^{n} c_j} \qquad (2.4)$$

5. Επιλογή σημείου διασταύρωσης

   Η διασταύρωση που είναι απαραίτητη για την δημιουργία των απογόνων μπορεί να γίνει σε ένα σημείο(1-point crossover), σε δύο σημεία (2-point crossover) ή περισσότερα ή ομοιόμορφα (uniform crossover).

Πίνακας 4. Διασταύρωση σε 1 σημείο(σημείο 4):						
	1ος Τελεστής	2ος Τελεστής	3ος Τελεστής	4ος Τελεστής	5ος Τελεστής	6ος Τελεστής
1ος Γονέας	0	0	1	1	0	1

2ος Γονέας	0	1	0	1	1	0
1ος Απόγονος	0	0	1	1	1	0
2ος Απόγονος	0	1	0	1	0	1

Πίνακας 5. Διασταύρωση σε 2 σημεία (σημεία 2 και 4)

	1ος Τελεστής	2ος Τελεστής	3ος Τελεστής	4ος Τελεστής	5ος Τελεστής	6ος Τελεστής
1ος Γονέας	0	0	1	1	0	1
2ος Γονέας	0	1	0	1	1	0
1ος Απόγονος	0	1	1	1	1	0
2ος Απόγονος	0	0	0	1	0	1

Πίνακας 6. Ομοιόμορφη διασταύρωση (τα γονίδια επιλέγονται τυχαία από τον ένα ή τον άλλο γονέα)

	1ος Τελεστής	2ος Τελεστής	3ος Τελεστής	4ος Τελεστής	5ος Τελεστής	6ος Τελεστής
1ος Γονέας	0	10	1	12	6	9
2ος Γονέας	2	66	0	15	71	8
1ος Απόγονος	0	66	1	15	71	9
2ος Απόγονος	2	10	0	12	6	8

6. Μετάλλαξη. Μετά τη διασταύρωση κάποιες φορές συμβαίνει και η μετάλλαξη στην δημιουργία των απογόνων. Μετάλλαξη ορίζεται η διαδικασία τυχαίας αλλαγής τις τιμής ενός γονιδίου. Με την μετάλλαξη εισάγονται νέα δεδομένα σε ένα πληθυσμό. Η μετάλλαξη θα πρέπει να δημιουργεί μία εφικτή λύση ή να αλλάζει ελάχιστα μία υπάρχουσα.
7. Επιλογή νέου πληθυσμού. Ο νέος πληθυσμός ή αλλιώς η επόμενη γενιά αποτελείται από απογόνους αλλά και γονείς. Η επιλογή των πιο αξιόλογων ατόμων του πληθυσμού γίνεται με βάση την αντικειμενική συνάρτηση.

   Παράδειγμα επιλογής νέου πληθυσμού:

   Αντικειμενική συνάρτηση max $f(x, y, z) = x + y + z$ x, y, z ακέραιες τιμές στο διάστημα [1,9]

   1η γενιά

Πίνακας 7. 1η γενιά

Γονείς	Τιμή αντικειμενικής συνάρτησης f	Απόγονοι με διασταύρωση στο 2ο σημείο	Μετάλλαξη στο 3ο σημείο	Τιμή αντικειμενικής συνάρτησης f
Α [2 5 7]	14	Ε [2 1 8]	Ε [2 1 9]	12
Β [3 1 8]	11	Ζ [3 5 7]	Ζ [3 5 3]	11
Γ [4 6 9]	19	Η [4 2 1]	Η [4 2 9]	15
Δ [2 2 1]	5	Θ [2 6 9]	Θ [2 6 1]	9

Ο πληθυσμός που θα συνεχίσει στην επόμενη γενιά αποτελείται από τα 4 άτομα με την μεγαλύτερη αξία: Α, Β, Ε, Η. Η διαδικασία θα επαναληφθεί με τα 4 άτομα ως γονείς, έως ότου τερματιστεί ο αλγόριθμος.

8. Ο αλγόριθμος τερματίζεται αν δεν υπάρχει άλλη βελτίωση, δημιουργηθούν μη αποδεκτές λύσεις ή δεν είναι δυνατή άλλη επανάληψη.

<u>Ψευδοκώδικας γενετικού αλγορίθμου</u>

INSERT N %πληθυσμός%

INSERT n %χρωμοσώματα%

INSERT X[N, n] %άτομα%

INSERT k %σημείο διασταύρωσης

INSERT g %γονίδιο μετάλλαξης

INSERT f() % συνάρτηση καταλληλότητας

WHILE (δεν ικανοποιούνται τα κριτήρια του βήματος 8)

{

Επιλογή γονέων

Διασταύρωση στο σημείο k

Μετάλλαξη του g χρωμοσώματος

Αξιολόγηση όλων με f()

Αντικατέστησε στον πίνακα X(N,n) τις N βέλτιστες λύσεις

}

RETURN X που αντιστοιχεί σε max f()

2.9. Αλγόριθμος Βελτιστοποίησης Σμήνους Σωματιδίων (Particle Swarm Optimization, PSO)

Οι Kennedy και Eberhart [13] μελετώντας την κοινωνική συμπεριφορά οργανισμών σε σμήνη δημιούργησαν τον αλγόριθμο ολικής βελτιστοποίησης σμήνους σωματιδίων. Η κίνηση των πτηνών σε σχηματισμούς, η ξαφνική αλλαγή κατεύθυνσης χωρίς καταστροφή του σχηματισμού και η διατήρηση τις απόστασης μεταξύ των πτηνών αποτελούν ένα καλά ισορροπημένο σύστημα, ευέλικτο σε αλλαγές. Οι κινήσεις ενός σωματιδίου μέσα στο σμήνος επηρεάζονται από τις κινήσεις των γειτονικών σε αυτό σωματιδίων, έτσι ο αλγόριθμος θεωρείτε συμβιωτικός και συνεργατικός.

Προτέρημα του αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων έναντι των γενετικών αλγορίθμων είναι ότι έχει μνήμη, δηλαδή οι καλές λύσεις μεταφέρονται στις επόμενες γενιές και δεν χάνονται. Επίσης υπάρχει μεγάλη συνεργασία μεταξύ των σωματιδίων προκειμένου να βρεθεί η λύση.

<u>Γενική περιγραφή αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων</u>

Τα σωματίδια βρίσκονται στο χώρο λύσεων σε συγκεκριμένη θέση $x_{i,j}$ και κινούνται με συγκεκριμένη ταχύτητα $v_{i,j}$.

Η θέση που έχει το κάθε σωματίδιο είναι μία πιθανή λύση και αναπαριστάτε με ένα διάνυσμα n-διαστάσεων στο χώρο των λύσεων $x_{i,j}$ ($i = 1,2, \ldots, N$, $j = 1,2, \ldots, n$, με N το μέγεθος του σμήνους και n το πλήθος των διαστάσεων).

Η ταχύτητα $v_{i,j}$ είναι η μετακίνηση από την μία θέση σε μία άλλη.

Η απόδοση της κάθε λύσης αξιολογείτε από μια συνάρτηση ποιότητας (fitness function) $f(x_{i,j})$.

Η κατεύθυνση του κάθε σωματιδίου εξαρτάτε από την αλληλεπίδραση του με τα γειτονικά σωματίδια και υπάρχουν 3 ενδεχόμενες κινήσεις. Το σωματίδιο μπορεί να ακολουθήσει τυχαία διαδρομή, μπορεί να κινηθεί προς την βέλτιστη θέση που είχε στη διάρκεια των επαναλήψεων $pbest_{i,j}$ (αυτό είναι δυνατό εφόσον υπάρχει μνήμη) ή να κινηθεί προς τη θέση που έχει το βέλτιστο σωματίδιο στον πληθυσμό $gbest_j$.

Οι μεταβλητές του προβλήματος υπολογίζονται ως εξής:

Ταχύτητα $v_{i,j}$

$$v_{i,j}(t+1) = v_{i,j}(t) + c_1 rand_1 \left(pbest_{i,j} - x_{i,j}(t)\right) + c_2 rand_2 \left(gbest_j - x_{i,j}(t)\right)$$
(2.5)

Θέση των σωματιδίων

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \qquad (2.6)$$

Όπου t ο αριθμός των επαναλήψεων, c_1, c_2 οι μεταβλητές επιτάχυνσης και $rand_1$, $rand_2$ τυχαίες μεταβλητές στο διάστημα [0,1]. Οι μεταβλητές επιτάχυνσης δείχνουν πόσο μακριά μπορεί να κινηθεί ένα σωματίδιο (συνήθως παίρνουν την τιμή 2).

Βέλτιστη θέση ενός σωματιδίου στο σμήνος

$$pbest_{i,j} = \begin{matrix} x_{i,j}(t+1), Εάν\, f(x_{i,j}(t+1)) < f(x_{i,j}(t)) \\ pbest_{i,j}, \end{matrix}, (YE\, rrpó\{3\}.1μα\, E\}.axt(YTorroí1(Y1c$$

$$pbest_{i,j} = \begin{matrix} x_{i,j}(t+1), Εάν\, f(x_{i,j}(t+1)) > f(x_{i,j}(t)) \\ pbest_{i,j}, \end{matrix}, (YE\, rrpó\{3\}.1μα\, μEyt(YTorroí1(Y1c$$

Βέλτιστη θέση όλου του σμήνους σε χρόνο t

$$gbest_j \in \{pbest_{1,j}, pbest_{2,j}, \dots, pbest_{N,j} | f(gbest_j)\}$$
$$= min\{f(gbest_{1j}), f(gbest_{2j}), \dots, f(gbest_{Nj})\} \qquad (2.7)$$

<u>Ψευδοκώδικας αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων</u>

INSERT N, n, $x_{i,j}(0)$, $v_{i,j}(0)$ %Εισαγωγή τιμών%

$f(x_{i,j})$ %Υπολογισμός αρχικού κόστους κάθε σωματιδίου%

FIND min $f(x_{i,j})$, i: best particle %βέλτιστο σωματίδιο, max αν μεγιστοποίηση%

FIND $pbest_{i,j}$ %βέλτιστη θέση κάθε σωματιδίου%

WHILE t< t_{max}

{

$v_{i,j}(t+1) = v_{i,j}(t) + c_1 rand_1 (pbest_{i,j} - x_{i,j}(t)) + c_2 rand_2 (gbest_j - x_{i,j}(t))$

$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1)$

FIND $f(x_{i,j})$ %νέα τιμή%

FIND $pbest_{i,j}$ %νέα τιμή%

$gbest_j = min\{f(gbest_{1j}), f(gbest_{2j}), \dots, f(gbest_{Nj})\}$

}

RETURN $gbest_j$

2.10. Αλγόριθμος πλησιέστερου γείτονα (k-nearest neighbour algorithm)

Ο αλγόριθμος παρουσιάστηκε το 1951 από τους Fix και Hodges [14] και είναι ένας από τους απλούστερους αλγορίθμους επίλυσης του προβλήματος του πλανόδιου πωλητή και παρεμφερών προβλημάτων.

<u>Γενική περιγραφή αλγορίθμου</u>

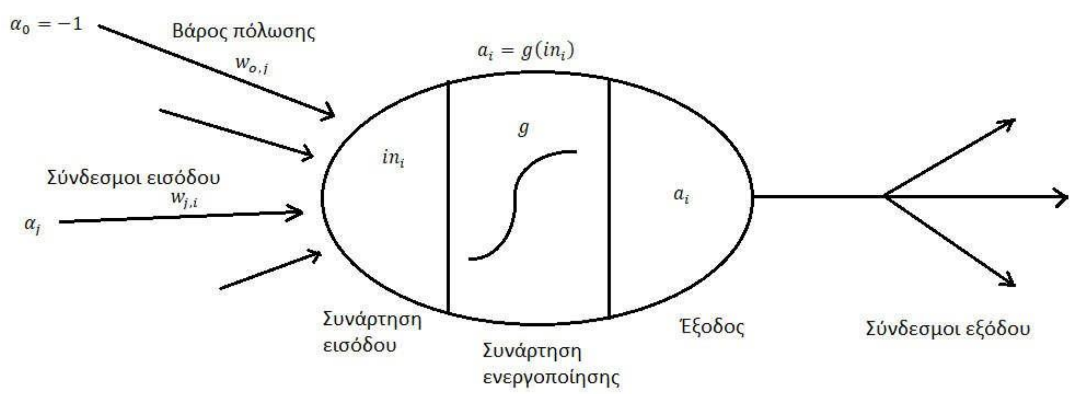
Έστω Ν οι κόμβοι του γραφήματος, ένας κενός πίνακας U[N] και η μεταβλητή C=0. Ο αλγόριθμος ξεκινά επιλέγοντας τυχαία ένα κόμβο. Ο κόμβος αυτός θα προστεθεί στην πρώτη θέση του πίνακα U. Στην πρώτη επανάληψη θα ελεγχθούν οι γειτονικοί σε αυτόν κόμβοι και θα επιλεγεί εκείνος με το μικρότερο κόστος, θα εισαχθεί στην δεύτερη θέση του πίνακα U και το κόστος της διαδρομής θα προστεθεί στην μεταβλητή C. Ο κάθε κόμβος επιτρέπεται να επιλεγεί μόνο μία φορά. Επιστροφή του αλγορίθμου θα είναι ο πίνακας U, όπου θα δηλώνει τη βέλτιστη διαδρομή και η μεταβλητή C που θα δηλώνει το συνολικό κόστος μετάβασης. Ο αλγόριθμος θα περατωθεί όταν προστεθούν όλοι οι κόμβοι στον πίνακα U.

3-Τεχνητή νοημοσύνη

Οι παρακάτω είναι διαφορετικοί τρόποι εκπαίδευσης των συστημάτων, για να αντιμετωπίσουν ένα πρόβλημα ορθολογικά.

3.1. Νευρωνικά δίκτυα

Το βασικό σύστημα επικοινωνίας του ανθρωπίνου οργανισμού ονομάζεται νευρικό σύστημα και τα κύτταρα που το αποτελούν είναι οι νευρώνες. Οι νευρώνες συγκεντρώνουν, επεξεργάζονται και διαδίδουν ηλεκτρικά σήματα. Η δυνατότητα επεξεργασίας πληροφοριών του εγκεφάλου απορρέει από δίκτυα τέτοιων νευρώνων. Στα πρώτα στάδια μελέτης της τεχνητής νοημοσύνης οι ερευνητές είχαν στόχο να δημιουργήσουν τεχνητά νευρωνικά δίκτυα. Ένα απλό μαθηματικό μοντέλο του νευρώνα που επινοήθηκε από τους McCulloch και Pitts (1943) [1] παρουσιάζεται στο σχήμα 2.



Σχήμα 2. Απλό μαθηματικό μοντέλο τεχνητού νευρωνικού δικτύου

Τα νευρωνικά δίκτυα (Neural Networks, NNs) έχουν τη δυνατότητα να επιλύουν μέσω αλγορίθμων μάθησης, εξαιρετικά πολύπλοκες μη γραμμικές εξισώσεις. Για να εκπαιδευτεί ένα νευρωνικό δίκτυο είναι απαραίτητο να εισαχθούν ζεύγη δεδομένων εισόδου (x_i) και εξόδου (y_i) με γνωστή τη σύνδεση μεταξύ τους αλλά και το βάρος που προσδιορίζει την ισχύ της σύνδεσης του ζεύγους $(W_{j,i})$.

Η σύνδεση αυτή για τα δεδομένα μάθησης είναι γνωστή οπότε στη συνέχεια το βάρος προσαρμόζεται ώστε τα δεδομένα εξόδου του συστήματος να είναι όσο πιο κοντά στα ζητούμενα.

Το σφάλμα πρόβλεψης ορίζεται ως η απόκλιση της επιθυμητής τιμής εξόδου από την τιμή εξόδου που δίνει το νευρωνικό δίκτυο (σχέση 2.7).

$$E = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 \tag{2.7}$$

Όπου x_i, y_i είναι τα δεδομένα εισόδου και εξόδου της μάθησης, $f(x_i)$ είναι το αποτέλεσμα του νευρωνικού δικτύου για το δεδομένο εισόδου x_i και N είναι το πλήθος των ζευγών δεδομένων που θα εισαχθούν για την μάθηση.

Για την ελαχιστοποίηση του σφάλματος έχουν προταθεί τρόποι μάθησης των δικτύων, με τον πιο διαδεδομένο να ονομάζεται οπισθοδιάδοση (backpropagation, BP) [15]. Η μέθοδος οπισθοδιάδοσης προκύπτει από την παραγώγιση της κλίσης του συνολικού σφάλματος. Υπολογίζει πόσο επηρεάζεται η τιμή του σφάλματος όταν τα βάρη αυξομειώνονται. Ο γενικός τύπος επαναπροσδιορισμού του βάρους στην οπισθοδιάδοσης είναι:

$$w_j = w_j - a \frac{aE}{aw_j} \tag{2.8}$$

όπου α είναι ο συντελεστής μεταβολής που ορίζει το μέγεθος της αλλαγής του βάρους, $\frac{aE}{aw_j}$ είναι η μερική παράγωγος της συνάρτησης του σφάλματος *E* ως προς το βάρος w_j. Η μερική παράγωγος $\frac{aE}{aw_j}$ δείχνει πόσο θα πρέπει να αλλάξει η τιμή του βάρους(w_j) ώστε να ελαχιστοποιηθεί η τιμή του σφάλματος *E*.

Ο αλγόριθμος οπισθοδιάδοσης εφαρμόζει την μεθόδο απότομης καθόδου στο χώρο των μεταβλητών ώστε να ελαχιστοποιήσει το σφάλμα εξόδου. Στην μέθοδο απότομης καθόδου ένα σφάλμα που συναντάται συχνά είναι ο εγκλωβισμός σε ένα τοπικό ελάχιστο (ή μέγιστο) ενώ αναζητείται ένα ολικό ελάχιστο (ή μέγιστο). Για να ξεπεραστεί αυτό το σφάλμα μπορεί να εφαρμοστεί ένας γενετικός αλγόριθμος ή ένας αλγόριθμος προσομοιωμένης ανόπτησης.

Ένας γενετικός αλγόριθμος (Genetic Algorithm, GA) έχει τη δυνατότητα να παράγει ένα ολικό ελάχιστο χωρίς να εγκλωβιστεί σε τοπικό ελάχιστο. Επιπλέον μπορεί να χρησιμοποιηθεί όταν δεν υπάρχουν πληροφορίες σχετικά με τον συντελεστή μεταβολής ή είναι πολύ δύσκολο (μεγάλο κόστος) να αποκτηθούν.

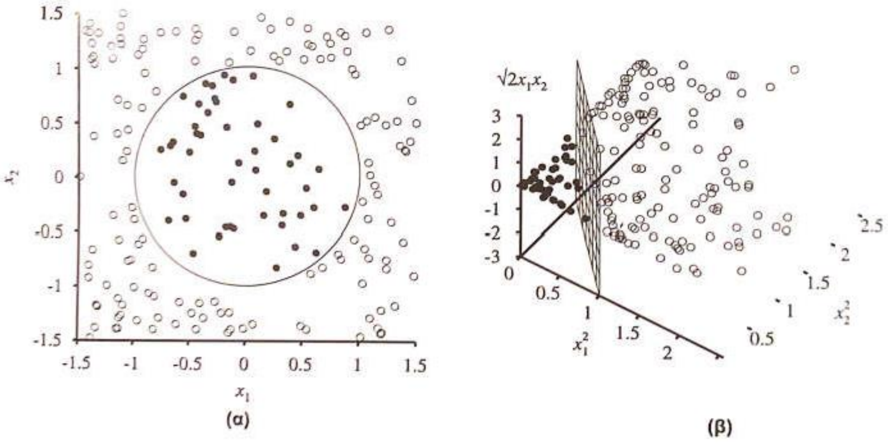
Με τον αλγόριθμο προσομοιωμένης ανόπτησης (Simulated Annealing, SA) μπορεί η αναζήτηση να ξεφύγει από ένα τοπικό βέλτιστο αλλά να παγιδευτεί σε μία γειτονιά αναζήτησης. Λύση σε αυτό το πρόβλημα δίνεται με τον αλγόριθμο περιορισμένης αναζήτησης.

Ο αλγόριθμος περιορισμένης αναζήτησης είναι αποτελεσματικός στην μάθηση των βαρών των νευρωνικών δικτύων, καθώς η αναζήτηση συνεχίζεται μετά από ένα τοπικό βέλτιστο και επιπλέον το αποτέλεσμα είναι ανθεκτικό στην επιλογή αρχικού σημείου.

3.2. Μηχανές διανυσμάτων υποστήριξης (Support Vector Machines, SVM)

Οι μηχανές διανυσμάτων υποστήριξης [16] ή πιο γενικά μηχανές πυρήνων (kernel machines) χρησιμοποιούν έναν αποδοτικό αλγόριθμο εκπαίδευσης και μπορούν να αναπαραστήσουν πολύπλοκες μη γραμμικές συναρτήσεις. Βασίζονται στη θεωρία στατιστικής μάθησης και εφαρμόζονται στην αναγνώριση εικόνας, γραφής, ταξινόμηση κειμένων και γενικά σε εφαρμογές με δεδομένα πολλών διαστάσεων.

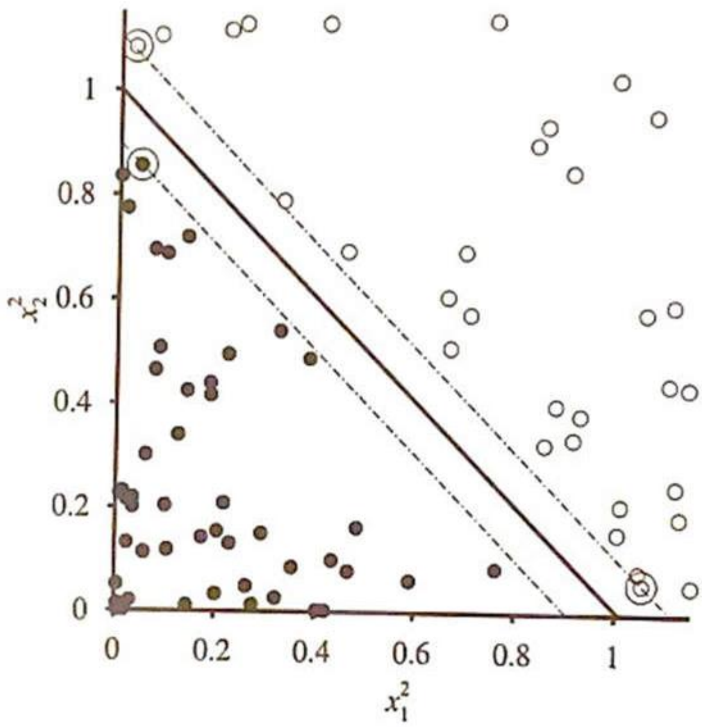
Στο σχήμα 3.α παρουσιάζεται ένας δισδιάστατος χώρος εισόδου με άξονες x_1 και x_2. Έστω οι αποδεκτές τιμές στους μαύρους κύκλους και οι μη αποδεκτές τιμές στους λευκούς κύκλους.



Σχήμα 3. (α) Δισδιάστατος χώρος εισόδου, (β) Τρισδιάστατος χώρος εισόδου

Δεν υπάρχει γραμμική σχέση που να διαχωρίζει τις αποδεκτές από τις μη αποδεκτές τιμές, υπάρχει όμως η μη γραμμική σχέση $x_1^2 + x_2^2 \leq 1$. Όταν τα ίδια δεδομένα οριστούν σε τρισδιάστατο χώρο με άξονες x_1^2, x_2^2 και $\sqrt{2}x_1x_2$ προκύπτει το σχήμα 2β. Παρατηρείται ότι τα δεδομένα σε αυτό το χώρο είναι γραμμικά διαχωρίσιμα. Σε αυτό βασίζεται ο γενικευμένος κανόνας ότι αν τα δεδομένα οριστούν σε ένα χώρο με πολλές διαστάσεις, τότε θα είναι πάντα γραμμικά διαχωρίσιμα. Αν υπάρχουν Ν δεδομένα τότε θα διαχωρίζονται πάντα σε χώρο Ν-1 διαστάσεων ή περισσοτέρων.

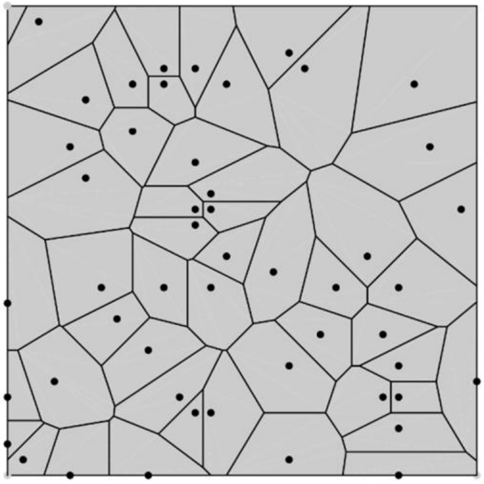
Οι μηχανές πυρήνων συνήθως βρίσκουν το βέλτιστο γραμμικό διαχωριστή , δηλαδή αυτόν που έχει το μεγαλύτερο περιθώριο (margin) μεταξύ των αποδεκτών και των μη αποδεκτών αποτελεσμάτων(εικόνα 3). Η εύρεση του γραμμικού διαχωριστή είναι πρόβλημα τετραγωνικού προγραμματισμού.



Σχήμα 4. Γραμμικός διαχωριστής μεταξύ αποδεκτών και μη αποδεκτών τιμών

3.3. Ομαδοποίηση K – means

Ο αλγόριθμος k-means αναπτύχθηκε το 1967 από τον McQueen [17] και αντικείμενο του είναι να ομαδοποιήσει n δεδομένα σε K ομάδες. Τα δεδομένα κατανέμονται στις ομάδες με εκείνα που μοιάζουν περισσότερο. Πιο ποιοτική διαχώριση θεωρείται εκείνη όπου στην ομάδα υπάρχει ομοιογένεια και μεταξύ των ομάδων μικρή ομοιότητα. Στον αλγόριθμο δεν δίνονται στοιχεία για τον διαχωρισμό αλλά μόνο το πλήθος των ομάδων. Αν θεωρηθεί ότι τα δεδομένα ανήκουν σε ένα επίπεδο και οι ομάδες θεωρούνται περιοχές αυτού του επιπέδου, ο διαχωρισμός γίνεται σε κελιά Vonoroi (εικόνα 4).



Σχήμα 5. Διάγραμμα κελιών Vonoroi

Γενική περιγραφή αλγορίθμου

Έστω το σύνολο n δεδομένων $x_1, x_2, \ldots, x_n$ που θα διαχωριστούν σε K ομάδες. Επιλέγονται τυχαία K στοιχεία ως μέσοι όροι των ομάδων. Για κάθε δεδομένο x_i υπολογίζεται το κοντινότερο σε αυτό στοιχείο που έχει οριστεί ως μέσος όρος της ομάδας. Υπολογίζονται οι νέοι μέσοι όροι των δεδομένων που έχουν οριστεί σε κάθε ομάδα. Ο αλγόριθμος περατώνεται όταν δεν γίνονται άλλες μεταβολές, δηλαδή όταν όλα τα δεδομένα έχουν μεταφερθεί στη σωστή ομάδα.

4-Εφαρμογές

4.1. Αναγνώριση αντικειμένου

Από τις πρώτες εφαρμογές των SVM ήταν η αναγνώριση χειρόγραφων αριθμών. Αποτέλεσμα αυτής της εφαρμογής ήταν η αυτόματη ταξινόμηση της αλληλογραφίας αναγνωρίζοντας τον ταχυδρομικό κωδικό, αυτόματη ανάγνωση επιταγών και αυτόματη καταχώρηση στοιχείων σε υπολογιστές. Στην μηχανική μάθηση ή αλλιώς σε αυτή την περίπτωση μηχανική όραση, η λειτουργία αυτή ονομάζεται αναγνώριση αντικειμένου.

Στο Εθνικό Ινστιτούτο Επιστήμης και Τεχνολογίας των ΗΠΑ (NIST) δημιουργήθηκε μια βάση δεδομένων με 60.000 ψηφία, κάθε ένα από τα οποία έχει 400 εικονοστοιχεία (20×20 pixels) και 8-bit χρώματα σε αποχρώσεις του γκρίζου. Αυτή η βάση δεδομένων χρησιμοποιήθηκε για την εκπαίδευση του αλγορίθμου. Η πρώτη προσέγγιση είναι ο ταξινομητής των 3 πλησιέστερων σημείων (3-nearest-neighbour) ο οποίος δεν χρειάζεται χρόνο εκπαίδευσης. Μειονέκτημα αυτού του τρόπου είναι ότι λειτουργεί αποθηκεύοντας και τις 60.000 εικόνες άρα η ταχύτητα εκτέλεσης του είναι μικρή. Η μέθοδος αυτή έχει ποσοστό σφάλματος 2,4%.

Άλλη προσέγγιση είναι η χρήση νευρωνικού δικτύου με ένα κρυφό επίπεδο με 400 μεταβλητές εισόδου (μία ανά εικονοστοιχείο) και 10 μεταβλητές εξόδου (μία για κάθε αριθμό 0-9). Όλες οι διασυνδέσεις μεταξύ των επιπέδων έχουν συνολικά 123.300 βάρη. Η μέθοδος αυτού του δικτύου έχει ποσοστό σφάλματος 1,6%.

Σχήμα 6. Παραδείγματα χειρόγραφων αριθμητικών ψηφίων από τη βάση δεδομένων NIST

Πλεονέκτημα της χρήσης μηχανών διανυσμάτων υποστήριξης είναι ότι μπορούν να υποστηρίξουν διανύσματα μεγάλων διαστάσεων. Μειονέκτημα είναι ότι τα δεδομένα της εκπαίδευσης δεν μπορούν να ανανεωθούν, δηλαδή σε περίπτωση που υπάρξουν νέα δεδομένα ο αλγόριθμος πρέπει να δημιουργηθεί από την αρχή.

Ήταν εύκολο μετά τη δημιουργία αυτού του αλγορίθμου η λειτουργία να επεκταθεί στην αναγνώριση ολόκληρων χειρόγραφων κειμένου. Μετά την σάρωση του χειρόγραφου κειμένου, με τον τρόπο που προαναφέρθηκε αλλά μεγαλύτερη βάση δεδομένων, αναγνωρίζονται τα γράμματα και το χειρόγραφο παρουσιάζεται σε ηλεκτρονική μορφή.

Η πιο πρόσφατη εξέλιξη αυτής της λειτουργίας είναι η βιομετρική αναγνώριση. Η αναγνώριση των δακτυλικών αποτυπωμάτων και των προσώπων από τα smart phones, η αναγνώριση της ίριδας του οφθαλμού από συστήματα υψίστης ασφαλείας είναι εφαρμογές αυτής της λειτουργίας.

4.2. Διανομή αγαθών

Η αλματώδης αύξηση των αγορών μέσω διαδικτύου που παρατηρείται τα τελευταία χρόνια έφερε τα δίκτυα διανομής στα όρια των δυνατοτήτων τους. Ιδιαίτερα την περίοδο της πανδημίας η πλειοψηφία των καταναλωτών στα αστικά κέντρα στράφηκε στις διαδικτυακές αγορές. Αποτέλεσμα αυτής της εξέλιξης είναι η επιβάρυνση του περιβάλλοντος λόγω της ραγδαίας αύξησης των εκπομπών ρύπων αλλά και της κυκλοφοριακής συμφόρησης, καθώς τα οχήματα διανομής επαναλαμβάνουν το ίδιο δρομολόγιο πολυάριθμες φορές μέσα σε μία μέρα.

Διανομή δεν υπάρχει μόνο όταν το τελικό προϊόν είναι έτοιμο αλλά σε όλα τα στάδια παραγωγής του. Από την μεταφορά των πρώτων υλών προς την μονάδα παραγωγής, τις εσωτερικές μεταφορές, την μεταφορά στις αποθήκες και τέλος την μεταφορά στα σημεία πώλησης ή την παραλαβή από μεταφορική εταιρία και την αποστολή στον καταναλωτή. Όλες αυτές οι μεταφορές έχουν κόστος, έχουν περιορισμούς στην ποσότητα, απαιτούν ανθρώπινους πόρους και χρονικό περιορισμό. Η επίλυση αυτού του προβλήματος απαιτεί κατανόηση του προβλήματος του πλανόδιου πωλητή και του προβλήματος δρομολόγησης οχημάτων.

4.2.1. Πρόβλημα πλανόδιου πωλητή

Ένα από τα σημαντικότερα προβλήματα που αντιμετώπισε η συνδυαστική βελτιστοποίηση είναι εκείνο του πλανόδιου πωλητή, το οποίο διατυπώθηκε πρώτη φορά από τους Hamilton και Kirkman το 1800 [18]. Το πρόβλημα του πλανόδιου πωλητή αποτελεί τη βάση για την επίλυση πολλών προβλημάτων βελτιστοποίησης.

<u>Γενική περιγραφή αλγορίθμου</u>

Ορίζεται ένα γράφημα, κορυφές του οποίου είναι οι σταθμοί από τους οποίους οφείλει να περάσει ο πλανόδιος πωλητής. Κάθε μετάβαση από τον ένα σταθμό στον άλλο έχει ένα κόστος. Ζητούμενο είναι ο πωλητής να περάσει από όλους τους σταθμούς ακριβώς μία φορά, με το μικρότερο δυνατό κόστος. Το γράφημα μπορεί να είναι προσανατολισμένο ή όχι. Αν το γράφημα δεν είναι προσανατολισμένο το κόστος μεταξύ 2 κόμβων είναι το ίδιο, ανεξάρτητα από ποιον σταθμό ξεκίνησε και σε ποιον θα καταλήξει. Έστω N το σύνολο των κόμβων, αν c_{ij} το κόστος μετάβασης από τον σταθμό i στον σταθμό j τότε $c_{ij} = c_{ji}$. Ισχύει επίσης ότι το κόστος είναι θετικός αριθμός ή 0 δηλαδή $c_{ij} \geq 0, \forall\, i, j \in N$. Ορίζεται η μεταβλητή $x_{ij} = 1$ αν υπάρχει σύνδεση μεταξύ 2 κόμβων και $x_{ij} = 0$ αν δεν υπάρχει.

Επίλυση με αλγόριθμο αναζήτησης

Ένας απλός αλλά χρονοβόρος τρόπος επίλυσης αυτού του προβλήματος είναι να δημιουργηθεί μία τυχαία αρχική λύση και στη συνέχεια να συγκριθεί με άλλες τυχαίες λύσεις. Ο αλγόριθμος θα περατωθεί όταν ελεγχθούν όλοι οι δυνατοί συνδυασμοί. Είναι προφανές ότι για πολλούς κόμβους, αυτός ο τρόπος επίλυσης είναι απαγορευτικός λόγω χρόνου και μνήμης, παρόλο που είναι ο πιο αξιόπιστος όλων καθώς δεν μένει κάποια πιθανή λύση που δεν έχει ελεγχθεί.

Σε αυτή τη κατηγορία ανήκει ο αλγόριθμος διακλάδωσης και οριοθέτησης (branch and bound). Με αυτό τον αλγόριθμο το πρόβλημα διαιρείται σε υποπροβλήματα και το κάθε ένα λύνεται ατομικά. Στο τέλος οι λύσεις συγκρίνονται μεταξύ τους για να βρεθεί η βέλτιστη. Στο πρόβλημα TSP υποπροβλήματα είναι όλες οι πιθανές διαδρομές. Όταν υπολογιστούν τα κόστη τους, συγκρίνονται και βρίσκεται η βέλτιστη λύση.

Ψευδοκώδικας προβλήματος πλανόδιου πωλητή

```
INSERT N   %αριθμός κόμβων%

INSERT c[N] %κόστη μετάβασης μεταξύ όλων των κόμβων%

INSERT x[N,N] %έλεγχος διαδρομής σύνδεσης

FOR i=1; i<N; i++{

        FOR j=1; j<N;j++{
```

```
                IF x_ij = 1 AND i ≠ j THEN
                        {C=C[i,j]
                        BEST[]=[i.j] } }
}
FOR i=1; i<N; i++{
        FOR j=1; j<N;j++{
                IF x_ij = 1 AND i ≠ j THEN
                        if (C[I,j]<C )  C= C[i,j]
                        BEST[]=[i.j]
RETURN C, BEST[]
```

<u>Επίλυση με k-nearest</u>

Ένας τρόπος επίλυσης του προβλήματος του πλανόδιου πωλητή είναι η κατασκευή μίας καλής αρχικής λύσης με τον αλγόριθμο πλησιέστερου γείτονα (k-nearest).

Ο αλγόριθμος ξεκινά με την τυχαία επιλογή ενός κόμβου. Στη συνέχεια ελέγχονται οι πλησιέστεροι σε αυτόν κόμβοι, δηλαδή εκείνοι που υπάρχει σύνδεση και επιλέγεται εκείνος με το μικρότερο κόστος μετάβασης. Επαναλαμβάνεται η διαδικασία με τους γειτονικούς κόμβους εκείνου που επιλέχθηκε. Ο αλγόριθμος σταματά όταν ελεγχθούν όλοι οι κόμβοι.

<u>Ψευδοκώδικας k-nearest</u>

```
INSERT N   %αριθμός κόμβων%
INSERT c[N] %κόστη μετάβασης μεταξύ όλων των κόμβων%
FOR i=1; i<N; i++{
```

FOR j=1; j<N;j++{

IF $x_{ij} = 1$ AND i $\neq j$ THEN

IF c[i, j]<c[i-1, j-1] THEN

min=c[i,j]   %αποθήκευση κόστους%

BEST[]=[i,j] %αποθήκευση διαδρομής%

RETURN min, BEST[]

Επίλυση με αλγόριθμο προσομοιωμένης ανόπτησης

Όπως παρουσιάστηκε στην ενότητα 2.9 ο αλγόριθμος προσομοιωμένης ανόπτησης αποδέχεται κακές λύσεις με πιθανότητα, για να ξεφύγει από ένα τοπικό βέλτιστο. Στην περίπτωση του πλανόδιου πωλητή εισάγεται στον αλγόριθμο η πιθανότητα αποδοχής κακής λύσης, αυτό είναι χρήσιμο σε περίπτωση εφαρμογής αλγορίθμου τοπικής αναζήτησης.

4.2.2. Πρόβλημα δρομολόγησης οχημάτων (Vehicle Routing Problem, VRP)

Το πρόβλημα δρομολόγησης οχημάτων αποτελεί επέκταση του προβλήματος του πλανόδιου πωλητή. Είναι ένα διαχρονικό πρόβλημα που διατυπώθηκε το 1959 από τους Dantzig και Ramser [19] και είναι εξίσου επίκαιρο σήμερα. Σκοπός της επίλυσης αυτού του προβλήματος είναι το κέρδος της επιχείρησης να υπερισχύσει του κόστους μεταφοράς.

Η μαθηματική διατύπωση του προβλήματος είναι παρεμφερής με εκείνη που πλανόδιου πωλητή:

$$\min z = \sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} c_{ij} x_{ij}^{k} \qquad (1)$$

Υπό τους περιορισμούς:

$$\sum_{j=0}^{N}\sum_{k=1}^{K} x_{ij}^{k} = 1, \forall j \in \{1, \dots, N\} \quad (2)$$

$$\sum_{j=0}^{N}\sum_{k=1}^{K} x_{ij}^{k} = 1, \forall i \in \{1, \dots, N\} \quad (3)$$

$$x_{ij}^{k} \in \{0, 1\} \quad (4)$$

Οι περιορισμοί (2), (3) εκφράζουν την μοναδική επίσκεψη κάθε κόμβου από το όχημα k και ο περιορισμός (4) εκφράζει την επίσκεψη στον κόμβο ή όχι (0 δεν υπάρχει επίσκεψη, 1 υπάρχει επίσκεψη). Αυτοί οι 3 περιορισμοί είναι οι βασικοί για την διατύπωση του προβλήματος, ανάλογα όμως με την περίπτωση μπορούν να υπάρξουν και άλλοι. Στην παραπάνω διατύπωση υπάρχει ένα σημείο διάθεσης και ένα όχημα. Ένα ρεαλιστικό σενάριο θα περιείχε κάποιους επιπλέον περιορισμούς. Κάποιοι από αυτούς τους περιορισμούς έχουν δημιουργήσει υποκατηγορίες στα προβλήματα δρομολόγησης οχημάτων, όπως οι παρακάτω.

<u>Σημεία διανομής</u>

Πολλές αποθήκες και πολλά σημεία διανομής, ένα όχημα απαιτείται να κάνει παραπάνω από μία παράδοση ή και παραλαβή σε ένα δρομολόγιο.

- Πρόβλημα δρομολόγησης οχημάτων πολλαπλών αποθηκών (Multi Depot VRP)
- Ταυτόχρονη παραλαβή και διανομή (Pickup and delivery VRP)
- Παραλαβή και επιστροφή προϊόντων (Backhauls VRP)

<u>Οχήματα</u>

Πολλά οχήματα με διαφορετική χωρητικότητα, ναυλωμένα ή ιδιόκτητα, κόστος συντήρησης και επισκευών

- Όριο χωρητικότητας οχημάτων (Capacitated VRP)

<u>Χρονικά περιθώρια</u>

Κάθε παραλαβή ή παράδοση έχει συγκεκριμένο χρονικό περιθώριο, τα αγαθά πρέπει να φτάσουν στον συμφωνημένο χρόνο και ένας αριθμός διανομών πρέπει να πραγματοποιηθεί σε μία βάρδια ή πολλές.

- Εξαρτώμενη από το χρόνο δρομολόγηση οχημάτων (Time Dependant VRP)
- Περιοδική δρομολόγηση (Periodic VPR)
- Προγραμματισμός παράδοσης πολλών ημερών

Εξωτερικοί παράγοντες

Ακόμη και αν όλα τα παραπάνω δρομολογηθούν υπάρχει μια αβεβαιότητα σχετικά με μη προβλέψιμους παράγοντες όπως τις έκτακτες καιρικές συνθήκες, τις απεργίες των εργαζομένων ή τις επιστροφές ελλαττωματικών προϊόντων.

- Δρομολόγηση με διασπαρμένες παραδόσεις (Split delivery VRP)

Αλγόριθμοι επίλυσης

Το πρόβλημα δρομολόγησης οχημάτων δεν είναι δυνατόν να επιλυθεί από έναν μοναδικό αλγόριθμο αφού είναι κατηγορίας NP-hard. Επιλύεται και αυτό το πρόβλημα με μεθευρετικούς αλγορίθμους και εφαρμογές της τεχνητής νοημοσύνης.

Αλγόριθμοι αναζήτησης

Ομοίως με το πρόβλημα του πλανόδιου πωλητή (ενότητα 4.2.1) οι αλγόριθμοι αναζήτησης θα πρέπει να υπολογίσουν τα κόστη όλων των πιθανών διαδρομών, να τα συγκρίνουν και να βρουν το βέλτιστο. Όπως αναφέρθηκε και πριν αυτός ο τρόπος επίλυσης είναι ο πιο αξιόπιστος αλλά και ο πιο χρονοβόρος. Η εφαρμογή του είναι αδύνατη για μεγάλο αριθμό κόμβων (>100).

Μεθευρετικοί αλγόριθμοι

Το πρόβλημα δρομολόγησης οχημάτων μπορεί να επιλυθεί με υβριδικούς αλγορίθμους νευρωνικών δικτύων σε συνδυασμό με έναν από: περιορισμένης αναζήτησης Tabu, προσομοιωμένης ανόπτησης, γενετικούς αλγορίθμους ή πλησιέστερου γείτονα.

4.3. Πρόβλεψη τάσεων ζήτησης στην αγορά

Σε ιδανικές συνθήκες μία επιχείρηση θα παρήγαγε ακριβώς την ζητούμενη από την αγορά ποσότητα ενός προϊόντος. Πρακτικά όμως είναι αδύνατος αυτός ο υπολογισμός, καθώς υπάρχουν πολλοί παράγοντες που επηρεάζουν την ζήτηση, όπως τα ανταγωνιστικά προϊόντα, η διαφήμιση, η τιμή πώλησης κ.α. Το μέγιστο κέρδος μιας επιχείρησης θα επιτευχθεί όταν η παραγόμενη ποσότητα έχει το ίδιο ακριβώς μέγεθος με την ζητούμενη ποσότητα. Μέσω της μηχανικής μάθησης είναι δυνατή η πρόβλεψη των τάσεων της αγοράς όπως παρουσιάζεται στο μοντέλο των Knoll et. all. [20], ώστε η παραγωγή να προσαρμοστεί κατάλληλα.

Οι Hess et al. [21] συνέκριναν τα δεδομένα από κλασικές μεθόδους πρόβλεψης και από μεθόδους μηχανικής μάθησης. Για βραχυπρόθεσμες προβλέψεις (μέχρι 2 μηνών) οι μέθοδοι μηχανικής μάθησης έδειξαν καλύτερα αποτελέσματα, δηλαδή είχαν μικρότερη απόκλιση από τα πραγματικά.

Το μοντέλο μελετά την εφοδιαστική αλυσίδα σαν οντότητα και τις επιμέρους λειτουργίες τις σαν υπομοντέλα. Ένα μέρος των υπομοντέλων αναλύονται μέσω μαθηματικών μοντέλων και μεθευρετικών αλγορίθμων. Τα υπόλοιπα που δεν ακολουθούν μαθηματικούς κανόνες εξετάζονται με την εφαρμογή της μηχανικής μάθησης. Μέσω της μηχανικής μάθησης θα αντληθούν πληροφορίες από ανταγωνιστικές εφοδιαστικές αλυσίδες ή άλλες παρόμοιες καταστάσεις. Οι πληροφορίες αυτές θα αποτελέσουν δεδομένα σύγκρισης έτσι ώστε να αξιολογηθεί η κατάσταση της εφοδιαστικής αλυσίδας και θα προβλεφθούν με την μεγαλύτερη δυνατή ακρίβεια οι μελλοντικές της ανάγκες.

Λαμβάνοντας τις πληροφορίες από τα μαθηματικά μοντέλα και τα μοντέλα μηχανικής μάθησης ο υπεύθυνος λειτουργίας της εφοδιαστικής αλυσίδας θα αξιολογήσει την δεδομένη κατάσταση, θα προβεί σε αλλαγές και η διαδικασία θα επαναληφθεί.

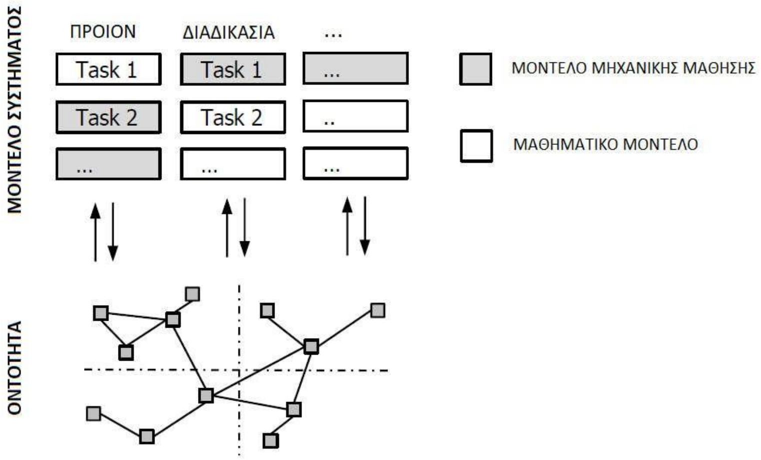
Το μοντέλο εφαρμόζεται σε 4 στάδια:

Δημιουργία της οντότητας που περιγράφει την εφοδιαστική αλυσίδα και καταγραφή των βασικών παραμέτρων λειτουργίας.

Το πρώτο βήμα σχεδίασης του μοντέλου είναι η εξακρίβωση των παραμέτρων της εφοδιαστικής αλυσίδας βάσει των οποίων θα αξιολογείται η πορεία της. Οι βασικοί σταθμοί στην παραγωγική διαδικασία είναι η προμήθεια υλών, η οργάνωση της γραμμής παραγωγής, η κατασκευή του προϊόντος, η διανομή και η πώληση. Κάθε ένας από αυτούς τους σταθμούς θα αποτελέσει υπο- μοντέλο του βασικού μοντέλου. Σε κάθε υπο-μοντέλο θα εξεταστεί ποια χαρακτηριστικά επηρεάζουν την απόδοση. Τα χαρακτηριστικά θα είναι οι παράμετροι για την αξιολόγηση της απόδοσης και ονομάζονται «Key Performance Indicators-KPI».

Μοντελοποίηση του συστήματος και αξιολόγηση, επανάληψη 1ου βήματος εφόσον κριθεί απαραίτητο.

Αφού έχουν οριστεί και κατηγοριοποιηθεί οι παράμετροι της εφοδιαστικής αλυσίδας, έχει δημιουργηθεί και η οντότητα, το επόμενο βήμα είναι να οριστεί ένα συστημικό μοντέλο πρόβλεψης των μελλοντικών αναγκών στην εφοδιαστική αλυσίδα. Οι μελλοντικές ανάγκες εξαρτώνται από πολλούς παράγοντες που η μηχανική μάθηση αδυνατεί να καλύψει. Οι παράγοντες που δεν καλύπτονται από την μηχανική μάθηση θα οριστούν από μαθηματικά μοντέλα. Το μοντέλο που χρειάζεται επίλυση θα αναλυθεί σε υπο-μοντέλα. Τα υπο-μοντέλα θα επιλυθούν με μαθηματικό τρόπο ή μέσω της μηχανικής μάθησης (ότι είναι απαραίτητο στην κάθε περίπτωση). Τα μαθηματικά μοντέλα και τα μοντέλα μηχανικής μάθησης θα συνδυαστούν με μοναδικό τρόπο για την κάθε περίπτωση όπως παρουσιάζει το σχήμα 7.



Σχήμα 7. Δημιουργία της οντότητας και των επιμέρους λειτουργιών του συστήματος

Τα μαθηματικά μοντέλα δημιουργούνται ορίζοντας μια αντικειμενική συνάρτηση και τους απαραίτητους περιορισμούς. Τα μοντέλα μηχανικής μάθησης δημιουργούνται χρησιμοποιώντας διαφορετικούς αλγορίθμους μηχανικής μάθησης όπως ο αλγόριθμος τυχαίου δάσους ή του νευρωνικού δικτύου.

<u>Δημιουργία μελλοντικού σχεδίου</u>

Η μοντελοποίηση του συστήματος όπως ορίστηκε στο προηγούμενο βήμα, δίνει στον αποφασίζοντα (logistics planner) ένα εργαλείο επιλογής των μελλοντικών του κινήσεων. Έχει δημιουργηθεί ένα πλασματικό μοντέλο σε ψηφιακό περιβάλλον προσομοίωσης της παραγωγικής διαδικασίας ευαίσθητο σε μεταβολές. Επιλέγοντας διαφορετικές κινήσεις σε οποιοδήποτε στάδιο της παραγωγής ο αποφασίζοντας μπορεί να εξετάσει την αντίδραση του συστήματος. Οι κινήσεις του αποφασίζοντα μπορεί να περιλαμβάνουν εισαγωγή νέου προϊόντος, αλλαγή του δικτύου διανομής, αύξηση ή μείωση της παραγωγής, κ.α.

<u>Πρόβλεψη μελλοντικών αλλαγών στην εφοδιαστική αλυσίδα, επανάληψη 3ου βήματος εφόσον κριθεί απαραίτητο.</u>

Η γνώση που θα προκύψει σχετικά με τις παραμέτρους αξιολόγησης απόδοσης θα υποδείξει τα σημεία που χρήζουν βελτίωσης. Όταν υπάρξει το περιθώριο βελτίωσης σε κάποιο στάδιο της παραγωγής το μοντέλο θα το υποδείξει και με αυτό τον τρόπο θα δρομολογηθούν οι αλλαγές. Προβλέποντας την μελλοντική ζήτηση, η παραγωγή θα ρυθμιστεί κατάλληλα. Επίσης θα ενημερωθούν τα δίκτυα διανομής (θα βελτιστοποιηθεί ο χώρος στα οχήματα) αλλά και τα σημεία πώλησης (θα προετοιμάσουν το απαραίτητο απόθεμα και το προσωπικό).

4.4. Πρόβλεψη και αποφυγή τροχαίων ατυχημάτων

Οι Ashikuzzaman et. al. [22] δημιούργησαν ένα μοντέλο πρόβλεψης και αποφυγής τροχαίων ατυχημάτων που βασίζεται στον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων και στα νευρωνικά δίκτυα. Τα αποτελέσματα του μοντέλου είναι ακριβέστερα από αντίστοιχα αποτελέσματα που βασίστηκαν σε άλλους αλγορίθμους (k-nearest neighbour, random forest, κ.α.).

Όταν προκύψει ατύχημα ή τραυματισμός η άμεση επέμβαση βοήθειας είναι ζωτικής σημασίας. Η επέμβαση βοήθειας είναι ένα επιχειρησιακό σχέδιο που μπορεί να δημιουργηθεί με δεδομένα που θα αντληθούν από τη μηχανική μάθηση. Μοντέλα μηχανικής μάθησης θα εκπαιδεύσουν το νευρωνικό δίκτυο με τον τρόπο που περιγράφηκε στην παράγραφο 3.1. Το προτέρημα των νευρωνικών δικτύων έναντι των άλλων μοντέλων είναι ότι αποδίδουν καλύτερα με μεγάλο όγκο δεδομένων. Ωστόσο τα μοντέλα νευρωνικών δικτύων τείνουν να εγκλωβίζονται σε τοπικά ελάχιστα. Ο υβριδικός αλγόριθμος PSO - ANN προσπερνά αυτό το πρόβλημα αφού ο PSO υπολογίζει το ολικό βέλτιστο εύκολα.

Οι σύνδεσμοι εισόδου του δικτύου αφορούν αριθμό οχημάτων, ολισθηρότητα οδοστρώματος, καιρικές συνθήκες, είδος οχήματος, ηλικία οδηγού, κ.α. Οι σύνδεσμοι εξόδου αφορούν την σοβαρότητα του τραύματος (θανατηφόρο, σοβαρό, ελαφρύ).

Μεθοδολογία

Τα δεδομένα κωδικοποιούνται και χωρίζονται σε 70% για μάθηση και 30% για έλεγχο. Υπάρχουν 12 σύνδεσμοι εισόδου, 5 κρυφά επίπεδα και 3 σύνδεσμοι εξόδου στο νευρωνικό δίκτυο με προς τα μπρος τροφοδότηση σήματος. Συνάρτηση ενεργοποίησης είναι η υπερβολική εφαπτομένη (tanh). Τα κρυφά επίπεδα δέχονται τα δεδομένα εισόδου και τα επεξεργάζονται με την συνάρτηση ενεργοποίησης. Η συνάρτηση ενεργοποίησης κάνει τη σχέση μεταξύ δεδομένων εισόδου και εξόδου μη γραμμική. Τα δεδομένα εξόδου ορίζονται ως σωματίδια και το σύνολο τους ως σμήνος. Για να βρεθεί το ολικό βέλτιστο θα εφαρμοστεί ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων.

Τα σωματίδια σε ένα σμήνος υπολογίζουν την απόσταση μεταξύ τους και κινούνται προς την βέλτιστη θέση. Η διαδικασία επαναλαμβάνεται 1000 φορές. Τα αποτελέσματα δείχνουν τη βέλτιστη θέση κάθε σωματιδίου αλλά και το βέλτιστη θέση από όλα τα σωματίδια.

5. Έλεγχος αποτελεσματικότητας αλγορίθμων

Ένα πρόβλημα διοικητικής επιστήμης θα επιλυθεί με διαφορετικούς αλγορίθμους ώστε να εξεταστεί η αποτελεσματικότητα τους.

Ο υπεύθυνος μίας μονάδας παραγωγής καλείτε να σχεδιάσει το σύστημα διανομής, να επιλύσει δηλαδή ένα πρόβλημα δρομολόγησης οχημάτων. Το κάθε φορτηγό θα αναλάβει την διανομή σε διαφορετική περιοχή. Ζητούμενο είναι να αποφασιστεί η βέλτιστη διαδρομή που θα ακολουθήσει το κάθε φορτηγό. Θα επιλυθεί το υποπρόβλημα ενός φορτηγού που θα ξεκινήσει από την αποθήκη (A), θα διανέμει σε 4 σταθμούς (B, C, D, E) και να επιστρέψει στην αποθήκη (A). Οι αποστάσεις μεταξύ των σταθμών παρουσιάζονται στον πίνακα 5.1.

Πίνακας 5.1. Αποστάσεις μεταξύ σταθμών

	A	B	C	D	E
A	-	2	3	4	5
B	2	-	3	2	1
C	3	3	-	6	2
D	4	2	6	-	2
E	5	1	2	2	-

5.1. Επίλυση με εξαντλητική αναζήτηση

Πριν επιλυθεί το πρόβλημα με τις μεθόδους που παρουσιάστηκαν στην δεύτερη ενότητα, θα επιλυθεί με τον αλγόριθμο εξαντλητικής αναζήτησης. Δηλαδή θα βρεθούν όλες οι εφικτές λύσεις, θα υπολογιστούν τα κόστη και θα βρεθεί η βέλτιστη. Η επίλυση με εξαντλητική αναζήτηση είναι η πλέον αξιόπιστη μέθοδος επίλυσης καθώς υπάρχει βεβαιότητα για την εύρεση της βέλτιστης λύσης. Μειονέκτημα της μεθόδου είναι ότι αν υπάρχουν πολλοί σταθμοί, ο όγκος των πράξεων εξαντλεί τη μνήμη του υπολογιστή. Το παράδειγμα μπορεί να επιλυθεί με τη μέθοδο εξαντλητικής αναζήτησης καθώς για τους 5 σταθμούς υπάρχουν συνολικά (5-1)!=24 λύσεις (πίνακας 5.2).

Πίνακας 5.2. Εξαντλητική αναζήτηση

Λύσεις							Κόστος διαδρομής
1	A	D	C	E	B	A	15
2	A	B	E	D	C	A	14
3	A	E	B	C	D	A	19
4	A	E	B	D	C	A	17
5	A	B	D	C	E	A	17
6	A	B	C	D	E	A	18
7	A	E	C	D	B	A	17
8	A	D	E	C	B	A	13
9	A	C	B	E	D	A	13
10	A	E	C	B	D	A	16
11	A	E	D	C	B	A	18
12	A	D	E	B	C	A	13
13	A	B	E	C	D	A	15
14	A	D	B	E	C	A	12
15	A	C	E	B	D	A	12
16	A	C	D	B	E	A	17
17	A	E	D	B	C	A	15
18	A	C	E	D	B	A	11
19	A	B	C	E	D	A	13
20	A	D	B	C	E	A	16
21	A	C	D	E	B	A	14

22	A	D	C	B	E	A	19
23	A	B	D	E	C	A	11
24	A	C	B	D	E	A	15

Οι βέλτιστες λύσεις του προβλήματος είναι οι A-C-E-D-B-A (18) και A-B-D-E-C-A (23) με κόστος 11, καθώς το πρόβλημα είναι ελαχιστοποίησης. Το αποτέλεσμα αυτό θα χρησιμοποιηθεί ως επαλήθευση για τους αλγορίθμους που θα εξεταστούν.

5.2. Επίλυση με αλγόριθμο τοπικής αναζήτησης

Το πρόβλημα επιλύεται με τον αλγόριθμο τοπικής αναζήτησης. Ο αλγόριθμος τερματίζεται στην 5η επανάληψη και ως βέλτιστη λύση υπολογίζει στην 4η επανάληψη την A C E B D A με κόστος 12. Η λύση αυτή αν και είναι κοντά, δεν είναι το ολικό βέλτιστο. Ο αλγόριθμος δηλαδή εγκλωβίστηκε σε ένα τοπικό βέλτιστο.

Πίνακας 5.3. Αλγόριθμος τοπικής αναζήτησης

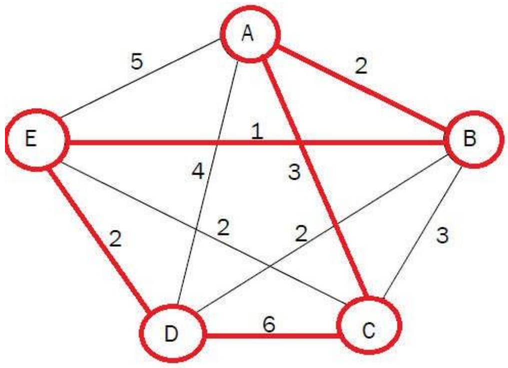
Επανάληψη	Λύση	Κόστος	Αποδοχή
Αρχικοποίηση	A B C D E A	18	Ναι
1	A C B D E A	15	Ναι
2	A B E D C A	14	Ναι
3	A D E C B A	13	Ναι
4	A C E B D A	12	Ναι
5	A D B C E A	16	Όχι – Τερματισμός αλγορίθμου και αποδοχή της προηγούμενης λύσης ως βέλτιστη

5.3. Επίλυση με αλγόριθμο προσομοιωμένης ανόπτησης

Η αποτελεσματικότητα του αλγορίθμου προσομοιωμένης ανόπτησης εξαρτάται από τον τρόπο μείωσης της θερμοκρασίας. Θα επιλυθεί το παράδειγμα με διαφορετικούς τρόπους μείωσης της θερμοκρασίας.

5.3.1. Αλγόριθμος προσομοιωμένης ανόπτησης με γραμμική μείωση θερμοκρασίας

<u>Αρχικοποίηση</u>



Αρχική λύση s_0= {A, B, E, D, C, A}

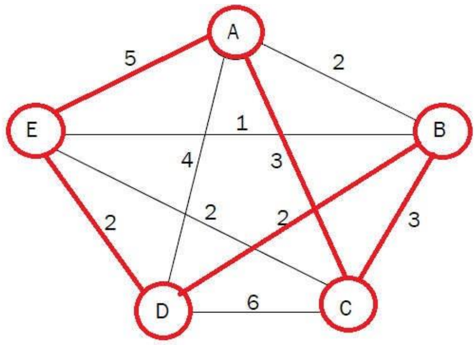
Αρχική θερμοκρασία t_0=200

Τέλος αλγορίθμου όταν t_i<1 ή max 10 επαναλήψεις

Μείωση θερμοκρασίας με κάθε επανάληψη $t_i = t_{i-1}/2$

Κόστος διαδρομής $f(s_0)$=14

<u>1η επανάληψη</u>



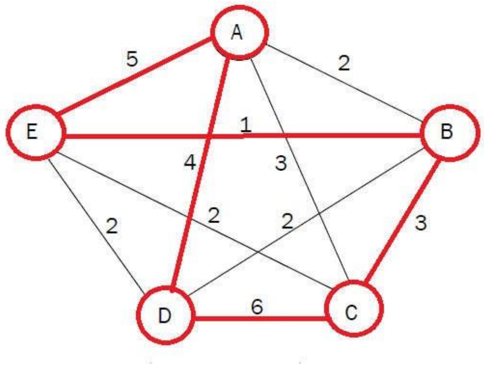
Επιλογή τυχαίας λύσης s_1= {A, C, B, D, E, A}

Κόστος διαδρομής $f(s_1)$=15

Θερμοκρασία $t_1 = t_0/2$= 100

Πιθανότητα αποδοχής P_{ac}= e^[-(f(s_1)- f(s_0))/T]=e^[-(15-14)/100]=0,99

<ins>2η επανάληψη</ins>



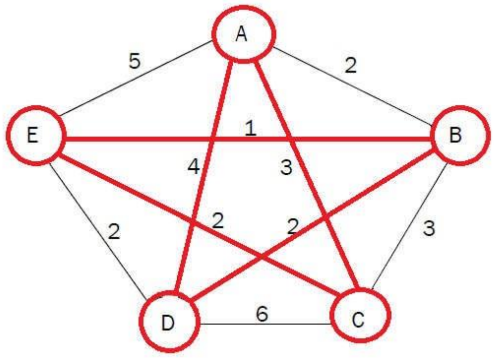
Επιλογή τυχαίας λύσης s_2= {A, E, B, C, D, A}

Κόστος διαδρομής $f(s_2)$=19

Θερμοκρασία t_2= t_1/2= 50

Πιθανότητα αποδοχής P_{ac}= e^[-($f(s_2)$- $f(s_1)$)/t_2]=e^[-(19-15)/50]=0,92

<ins>3η επανάληψη</ins>



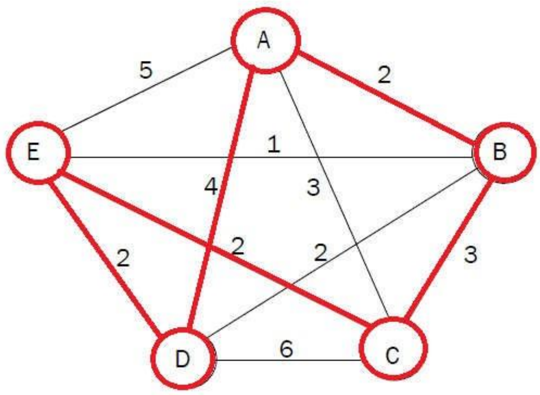
Επιλογή τυχαίας λύσης s_3= {A, C, E, B, D, A}

Κόστος διαδρομής $f(s_3)$=12

Θερμοκρασία t_3= t_2/2= 25

Πιθανότητα αποδοχής P_{ac}= 1, αφού $f(s_3)$< $f(s_2)$

<ins>4η επανάληψη</ins>

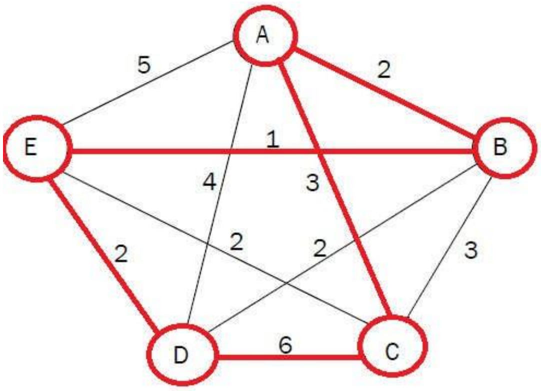


Επιλογή τυχαίας λύσης s_3= {A, D, E, C, B, A}

Κόστος διαδρομής $f(s_4)$= 13

Θερμοκρασία t_3= t_2/2= 12,5

Πιθανότητα αποδοχής P_{ac}= e^[-($f(s_4)$- $f(s_3)$)/t_4]=e^[-(13-12)/12,5]=0,92



<u>5η επανάληψη</u>

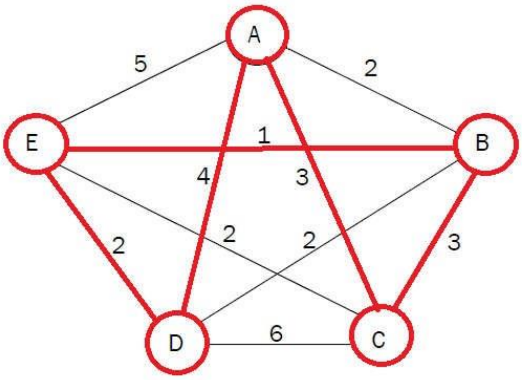
Επιλογή τυχαίας λύσης s_5= { A, B, E, D, C, A}

Κόστος διαδρομής $f(s_5)$= 14

Θερμοκρασία $t_3= t_2/2$= 6,25

Πιθανότητα αποδοχής P_{ac}= e^[-(f(s_5)- f(s_4))/t_5]=e^[-(14-13)/6,25]=0,852

**Παρατηρείται ότι η λύση της 5ης επανάληψης s_5 ταυτίζεται με την αρχική λύση s_0 και αυτό είναι το μειονέκτημα της μεθόδου. Το πρόβλημα αυτό παρακάμπτεται με τη μέθοδο περιορισμένης αναζήτησης (5.4).*



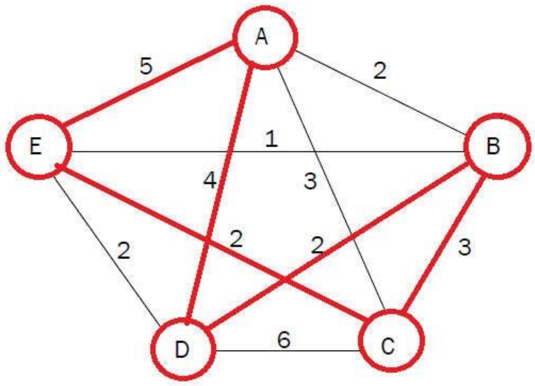
<u>6η επανάληψη</u>

Επιλογή τυχαίας λύσης s_6= { A, C, B, E, D, A}

Κόστος διαδρομής $f(s_6)$= 13

Θερμοκρασία $t_6= t_5/2$= 3,125

Πιθανότητα αποδοχής P_{ac}= 1, αφού $f(s_6)< f(s_5)$



<u>7η επανάληψη</u>

Επιλογή τυχαίας λύσης s_7= { A, D, B, C, E, A}

Κόστος διαδρομής $f(s_7)$= 16

Θερμοκρασία $t_7= t_6/2$= 1,5625

Πιθανότητα αποδοχής P_{ac}= e^[-(f(s_7)- f(s_6))/t_7]=e^[-(16-13)/1,5625]= 0,14

8η επανάληψη

Θερμοκρασία $t_8 = t_7/2 = 0{,}78125$

Τερματισμός αλγορίθμου αφού $t_8 < 1$

Πίνακας 5.4. Αποτελέσματα αλγορίθμου προσομοιωμένης ανόπτησης με γραμμική μείωση θερμοκρασίας

Επανάληψη	Θερμοκρασία	Λύση	Κόστος	Πιθανότητα αποδοχής	Τυχαίος αριθμός $\sim(0, 1)$	Αποδοχή
Αρχικοποίηση	200	A B E D C A	14			
1	100	A C B D E A	15	0,99	0,974059	Ναι
2	50	A E B C D A	19	0,92	0,769768	Ναι
3	25	A C E B D A	12	1	0,558245	Ναι
4	12,5	A D E C B A	13	0,92	0,192846	Ναι
5	6,25	A B E D C A	14	0,852	0,208289	Ναι
6	3,125	A C B E D A	13	1	0,145848	Ναι
7	1,5625	A D B C E A	16	0,14	0,656514	Όχι

8	0,78125	Τερματισμός

Ο αλγόριθμος τερματίστηκε στην 8η επανλάληψη και η βέλτιστη λύση που υπολόγισε είναι η A C B E D A με κόστος 13. Η λύση είναι τοπικό βέλτιστο και όχι ολικό.

5.3.2. Αλγόριθμος προσομοιωμένης ανόπτησης με εκθετική μείωση της θερμοκρασίας.

Θα επιλυθεί το πρόβλημα με τον αλγόριθμο προσομοιωμένης ανόπτησης και εκθετική μείωση θερμοκρασίας. Η θερμοκρασία σε κάθε επανάληψη θα μειώνεται ακολουθώντας τον τύπο $t_i = t_0 e^{-i\ t_0}$, όπου i ο αριθμός της επανάληψης και t_0 η αρχική θερμοκρασία.

Πίνακας 5.5. Αλγόριθμος προσομοιωμένης ανόπτησης με εκθετική μείωση θερμοκρασίας

Επανάληψη	Θερμοκρασία	Λύση	Κόστος	Πιθανότητα αποδοχής	Τυχαίος αριθμός $\sim(0, 1)$	Αποδοχή
Αρχικοποίηση	200	A B E D C A	14			
1	73,57589	A C B D E A	15	0,986500537	0,229712821	Ναι
2	27,06706	A E B C D A	19	0,862619903	0,234076968	Όχι

3	9,957414	A C E B D A	12	1	0,632770775	Ναι
4	3,663128	A D E C B A	13	0,761099828	0,581865902	Ναι
5	1,347589	A B E D C A	14	0,476129313	0,658162175	Όχι
6	0,49575	Τερματισμός				

Ο αλγόριθμος τερματίστηκε στην 6η επανάληψη και η βέλτιστη λύση που υπολόγισε είναι η A D E C B A με κόστος 12. Η λύση είναι τοπικό βέλτιστο και όχι ολικό.

5.3.3. Αλγόριθμος προσομοιωμένης ανόπτησης με λογαριθμική μείωση της θερμοκρασίας.

Θα επιλυθεί το πρόβλημα με τον αλγόριθμο προσομοιωμένης ανόπτησης και λογαριθμική μείωση θερμοκρασίας. Η θερμοκρασία σε κάθε επανάληψη θα μειώνεται ακολουθώντας τον τύπο $t_i = t_0 log(i)$, όπου i ο αριθμός της επανάληψης και t_0 η αρχική θερμοκρασία.

Πίνακας 5.6. Αλγόριθμος προσομοιωμένης ανόπτησης με λογαριθμική μείωση θερμοκρασίας

Επανάληψη	Θερμοκρασία	Λύση	Κόστος	Πιθανότητα αποδοχής	Τυχαίος αριθμός $\sim(0, 1)$	Αποδοχή

Αρχικοποίηση	200	A B E D C A	14			
1	198	A C B D E A	15	0,927040917	0,39005707	Ναι
2	196	A E B C D A	19	0,911914562	0,225928526	Ναι
3	194	A C E B D A	12	1	0,752891629	Ναι
4	192	A D E C B A	13	0,939413063	0,892178106	Ναι
5	190	A B E D C A	14	0,933569441	0,342722861	Ναι
6	188	A D E C B A	13	1	0,799401837	Ναι
7	186	A D B C E A	16	0,92252098	0,263740959	Ναι
8	184	A B C D E A	18	0,911364469	0,089754936	Ναι

9	182	A C E B D A	12	1	0,287911618	Ναι
10	180	A B E D C A	14	0,930324134	0,292050267	Όχι
Τερματισμός						

Ο αλγόριθμος τερματίστηκε στην 10η επανάληψη και η βέλτιστη λύση που υπολόγισε είναι η A C E B D A με κόστος 12. Η λύση είναι τοπικό βέλτιστο και όχι ολικό.

5.4. Επίλυση αλγορίθμου περιορισμένης αναζήτησης

Θα επιλυθεί ο αλγόριθμος περιορισμένης αναζήτησης με μνήμη 5 θέσεων.

Πίνακας 5.7. Αλγόριθμος περιορισμένης αναζήτησης με μνήμη 5 θέσεων

Επ.	Λύση	Κόστος	Λίστα Tabu	Βέλτιστη λύση
0	AEBDCA	17	AEBDCA	AEBDCA
1	ACEBDA	12	AEBDCA, ACEBDA	ACEBDA
2	ABEDCA	14	AEBDCA, ACEBDA, ABEDCA	ACEBDA
3	ADECBA	13	AEBDCA, ACEBDA, ABEDCA, ADECBA	ACEBDA
4	ADBECA	12	AEBDCA, ACEBDA, ABEDCA, ADECBA, ADBECA	ACEBDA
5	ABCEDA	13	ACEBDA, ABEDCA, ADECBA, ADBECA, ABCEDA	ACEBDA
6	ACDEBA	14	ABEDCA, ADECBA, ADBECA, ABCEDA, ACDEBA	ACEBDA
7	ACEDBA	11	ADECBA, ADBECA, ABCEDA, ACDEBA, ACEDBA	ACEDBA

8	ADCEBA	15	ADBECA, ADCEBA	ABCEDA,	ACDEBA,	ACEDBA,	ACEDBA
9	AEBCDA	19	ABCEDA, AEBCDA	ACDEBA,	ACEDBA,	ADCEBA,	ACEDBA
10	ACBDEA	15	ACDEBA, ACBDEA	ACEDBA,	ADCEBA,	AEBCDA,	ACEDBA

Ο αλγόριθμος τερματίζεται στην 10η επανάληψη και η βέλτιστη λύση που υπολόγισε είναι η ACEDBA με κόστος 11. Η βέλτιστη λύση είναι το ολικό βέλτιστο, όπως μπορεί να επαληθευτεί από το αποτέλεσμα της εξαντλητικής αναζήτησης.

5.5. Επίλυση γενετικού αλγορίθμου

Η αποτελεσματικότητα του γενετικού αλγορίθμου εξαρτάτε από τον τρόπο επιλογής των γονέων. Θα επιλυθεί ο γενετικός αλγόριθμος με διασταύρωση ενός σημείου. Στην επίλυση του προβλήματος με τη διασταύρωση ενός σημείου είναι πιθανό να προκύψει ένα σφάλμα εμφάνισης του ίδιου κόμβου 2 φορές στους απογόνους. Όταν συμβεί αυτή η περίπτωση, η μία εμφάνιση του κόμβου θα αντικατασταθεί τυχαία με τον κόμβο που δεν εμφανίστηκε καθόλου. Η επιλογή γονέων θα γίνει ανά ζεύγη και με τη μέθοδο της ρουλέτας.

5.5.1. Επίλυση γενετικού αλγορίθμου με επιλογή γονέων ανά ζεύγη

Στο πρόβλημα υπάρχουν 24 εφικτές λύσεις. Ο πιο απλός τρόπος επίλυσης είναι να οριστούν όλες οι λύσεις ως γονείς και να διασταυρωθούν ανά ζεύγη.

Πίνακας 5.8. Γενετικός αλγόριθμος με διασταύρωση ενός σημείου και επιλογή γονέων ανά ζεύγη

1η γενιά							Αξία
Γονέας Α'	A	D	C	E	B	A	15
ΓονέαςΒ'	A	B	E	D	C	A	14
Απόγονος ΑΑ'	A	D	C	D	C	A	

Απόγονος ΑΒ'	A	B	E	E	B	A	

Επιδιόρθωση του σφάλματος:

Απόγονος ΑΑ	A	D	E	B	C	A	13
Απόγονος ΑΒ	A	D	E	C	B	A	13

Γονέας Γ'	A	E	B	C	D	A	19
Γονέας Δ'	A	E	B	D	C	A	17
Απόγονος ΑΓ'	A	E	B	D	C	A	17
Απόγονος ΑΔ'	A	E	B	C	D	A	19

Γονέας Ε'	A	B	D	C	E	A	17
Γονέας Ζ'	A	B	C	D	E	A	18
Απόγονος ΑΕ'	A	B	D	D	E	A	
Απόγονος ΑΖ'	A	B	C	C	E	A	

Επιδιόρθωση του σφάλματος:

Απόγονος ΑΕ'	A	B	D	C	E	A	17
Απόγονος ΑΖ'	A	B	C	D	E	A	18

Γονέας Η'	A	E	C	D	B	A	17
Γονέας Θ'	A	D	E	C	B	A	13
Απόγονος ΑΗ'	A	E	C	C	B	A	
Απόγονος ΑΘ'	A	D	E	D	B	A	
Επιδιόρθωση του σφάλματος:							
Απόγονος ΑΗ'	A	E	D	C	B	A	18
Απόγονος ΑΘ'	A	C	E	D	B	A	11

Γονέας Ι'	A	C	B	E	D	A	13
Γονέας Κ'	A	E	C	B	D	A	16
Απόγονος ΑΙ'	A	C	B	B	D	A	
Απόγονος ΑΚ'	A	E	C	E	D	A	
Επιδιόρθωση του σφάλματος:							
Απόγονος ΑΙ'	A	C	E	B	D	A	12
Απόγονος ΑΚ'	A	B	C	E	D	A	13

Γονέας Λ'	A	E	D	C	B	A	18
Γονέας Μ'	A	D	E	B	C	A	13
Απόγονος ΑΛ'	A	E	D	B	C	A	15
Απόγονος ΑΜ'	A	D	E	C	B	A	13

Γονέας Ν'	A	B	E	C	D	A	15
Γονέας Ξ'	A	D	B	E	C	A	12
Απόγονος ΑΝ'	A	B	E	E	C	A	
Απόγονος ΑΞ'	A	D	B	C	D	A	
Επιδιόρθωση του σφάλματος:							
Απόγονος ΑΝ'	A	B	D	E	C	A	11
Απόγονος ΑΞ'	A	E	B	C	D	A	17

Γονέας Ο'	A	C	E	B	D	A	12
Γονέας Π'	A	C	D	B	E	A	17
Απόγονος ΑΟ'	A	C	E	B	E	A	
Απόγονος ΑΠ'	A	C	D	B	D	A	

Επιδιόρθωση του σφάλματος:

Απόγονος ΑΟ'	A	C	E	B	D	A	12
Απόγονος ΑΠ'	A	C	E	B	D	A	12

Γονέας Ρ'	A	E	D	B	C	A	15
Γονέας Σ'	A	C	E	D	B	A	11

Απόγονος ΑΡ'	A	E	D	D	B	A
Απόγονος ΑΣ'	A	C	E	B	C	A

Επιδιόρθωση του σφάλματος
:

Απόγονος ΑΡ'	A	E	C	D	B	A	17
Απόγονος ΑΣ'	A	D	E	B	C	A	13

Γονέας Τ'	A	B	C	E	D	A	13
Γονέας Υ'	A	D	B	C	E	A	16

Απόγονος ΑΤ'	A	B	C	C	E	A
Απόγονος ΑΥ'	A	D	B	E	D	A

Επιδιόρθωση του σφάλματος:

Απόγονος ΑΤ'	A	B	C	D	E	A	18
Απόγονος ΑΥ'	A	C	B	E	D	A	13

Γονέας Φ'	A	C	D	E	B	A	14
Γονέας Χ'	A	D	C	B	E	A	19
Απόγονος ΑΦ'	A	C	D	B	E	A	17
Απόγονος ΑΧ'	A	D	C	E	B	A	15

Γονέας Ψ'	A	B	D	E	C	A	11
Γονέας Ω'	A	C	B	D	E	A	15
Απόγονος ΑΨ'	A	B	D	D	E	A	
Απόγονος ΑΩ'	A	C	B	E	C	A	
Επιδιόρθωση του σφάλματος:							
Απόγονος ΑΨ'	A	B	C	D	E	A	18
Απόγονος ΑΩ'	A	D	B	E	C	A	12

Ο νέος πληθυσμός που θα προκύψει αποτελείται από τους 24 γονείς και από 24 απογόνους όπως παρουσιάζεται στον πίνακα 5.9.

Πίνακας 5.9. Νέος πληθυσμός

Πληθυσμός							Αξία
Γονέας Α'	A	D	C	E	B	A	15
ΓονέαςΒ'	A	B	E	D	C	A	14
Απόγονος ΑΑ	A	D	E	B	C	A	13
Απόγονος ΑΒ	A	D	E	C	B	A	13
Γονέας Γ'	A	E	B	C	D	A	19
Γονέας Δ'	A	E	B	D	C	A	17
Απόγονος ΑΓ'	A	E	B	D	C	A	17
Απόγονος ΑΔ'	A	E	B	C	D	A	19
Γονέας Ε'	A	B	D	C	E	A	17
Γονέας Ζ'	A	B	C	D	E	A	18
Απόγονος ΑΕ'	A	B	D	C	E	A	17
Απόγονος ΑΖ'	A	B	C	D	E	A	18
Γονέας Η'	A	E	C	D	B	A	17
Γονέας Θ'	A	D	E	C	B	A	13
Απόγονος ΑΗ'	A	E	D	C	B	A	18
Απόγονος ΑΘ'	A	C	E	D	B	A	11
Γονέας Ι'	A	C	B	E	D	A	13
Γονέας Κ'	A	E	C	B	D	A	16
Απόγονος ΑΙ'	A	C	E	B	D	A	12
Απόγονος ΑΚ'	A	B	C	E	D	A	13
Γονέας Λ'	A	E	D	C	B	A	18
Γονέας Μ'	A	D	E	B	C	A	13
Απόγονος ΑΛ'	A	E	D	B	C	A	15
Απόγονος ΑΜ'	A	D	E	C	B	A	13
Γονέας Ν'	A	B	E	C	D	A	15
Γονέας Ξ'	A	D	B	E	C	A	12
Απόγονος ΑΝ'	A	B	D	E	C	A	11
Απόγονος ΑΞ'	A	E	B	C	D	A	17
Γονέας Ο'	A	C	E	B	D	A	12
Γονέας Π'	A	C	D	B	E	A	17

Απόγονος ΑΟ'	A	C	E	B	D	A	12
Απόγονος ΑΠ'	A	C	E	B	D	A	12
Γονέας Ρ'	A	E	D	B	C	A	15
Γονέας Σ'	A	C	E	D	B	A	11
Απόγονος ΑΡ'	A	E	C	D	B	A	17
Απόγονος ΑΣ'	A	D	E	B	C	A	13
Γονέας Τ'	A	B	C	E	D	A	13
Γονέας Υ'	A	D	B	C	E	A	16
Απόγονος ΑΤ'	A	B	C	D	E	A	18
Απόγονος ΑΥ'	A	C	B	E	D	A	13
Γονέας Φ'	A	C	D	E	B	A	14
Γονέας Χ'	A	D	C	B	E	A	19
Απόγονος ΑΦ'	A	C	D	B	E	A	17
Απόγονος ΑΧ'	A	D	C	E	B	A	15
Γονέας Ψ'	A	B	D	E	C	A	11
Γονέας Ω'	A	C	B	D	E	A	15
Απόγονος ΑΨ'	A	B	C	D	E	A	18
Απόγονος ΑΩ'	A	D	B	E	C	A	12

Ο πίνακας 5.10 παρουσιάζει τον πληθυσμό ταξινομημένο. Η σκιαγραφημένη περιοχή περιλαμβάνει τα μέλη που θα επιβιώσουν.

Πίνακας 5.10. Ταξινόμηση του πληθυσμού							
Ταξινόμηση							
Απόγονος ΑΘ'	A	C	E	D	B	A	11
Απόγονος ΑΝ'	A	B	D	E	C	A	11
Γονέας Σ'	A	C	E	D	B	A	11
Γονέας Ψ'	A	B	D	E	C	A	11
Απόγονος ΑΙ'	A	C	E	B	D	A	12

Γονέας Ξ'	A	D	B	E	C	A	12
Γονέας Ο'	A	C	E	B	D	A	12
Απόγονος ΑΟ'	A	C	E	B	D	A	12
Απόγονος ΑΠ'	A	C	E	B	D	A	12
Απόγονος ΑΩ'	A	D	B	E	C	A	12
Απόγονος ΑΑ	A	D	E	B	C	A	13
Απόγονος ΑΒ	A	D	E	C	B	A	13
Γονέας Θ'	A	D	E	C	B	A	13
Γονέας Ι'	A	C	B	E	D	A	13
Απόγονος ΑΚ'	A	B	C	E	D	A	13
Γονέας Μ'	A	D	E	B	C	A	13
Απόγονος ΑΜ'	A	D	E	C	B	A	13
Απόγονος ΑΣ'	A	D	E	B	C	A	13
Γονέας Τ'	A	B	C	E	D	A	13
Απόγονος ΑΥ'	A	C	B	E	D	A	13
ΓονέαςΒ'	A	B	E	D	C	A	14
Γονέας Φ'	A	C	D	E	B	A	14
Γονέας Α'	A	D	C	E	B	A	15
Απόγονος ΑΛ'	A	E	D	B	C	A	15
Γονέας Ν'	A	B	E	C	D	A	15
Γονέας Ρ'	A	E	D	B	C	A	15

Απόγονος ΑΧ'	A	D	C	E	B	A	15
Γονέας Ω'	A	C	B	D	E	A	15
Γονέας Κ'	A	E	C	B	D	A	16
Γονέας Υ'	A	D	B	C	E	A	16
Γονέας Δ'	A	E	B	D	C	A	17
Απόγονος ΑΓ'	A	E	B	D	C	A	17
Γονέας Ε'	A	B	D	C	E	A	17
Απόγονος ΑΕ'	A	B	D	C	E	A	17
Γονέας Η'	A	E	C	D	B	A	17
Απόγονος ΑΞ'	A	E	B	C	D	A	17
Γονέας Π'	A	C	D	B	E	A	17
Απόγονος ΑΡ'	A	E	C	D	B	A	17
Απόγονος ΑΦ'	A	C	D	B	E	A	17
Γονέας Ζ'	A	B	C	D	E	A	18
Απόγονος ΑΖ'	A	B	C	D	E	A	18
Απόγονος ΑΗ'	A	E	D	C	B	A	18
Γονέας Λ'	A	E	D	C	B	A	18
Απόγονος ΑΤ'	A	B	C	D	E	A	18
Απόγονος ΑΨ'	A	B	C	D	E	A	18
Γονέας Γ'	A	E	B	C	D	A	19
Απόγονος ΑΔ'	A	E	B	C	D	A	19

Γονέας Χ'	A	D	C	B	E	A	19

5.5.2. Επίλυση γενετικού αλγορίθμου με επιλογή γονέων με τη μέθοδο της ρουλέτας

Το πρόβλημα δρομολόγησης οχημάτων είναι πρόβλημα ελαχιστοποίησης και η μέθοδος της ρουλέτας έρχεται σε αντίθεση με το ζητούμενο. Όσο μεγαλύτερη η τιμή της αντικειμενικής συνάρτησης μίας λύσης, τόσο περισσότερες πιθανότητες υπάρχουν να επιλεχθεί ως γονέας με τη μέθοδο της ρουλέτας. Στο πρόβλημα δρομολόγησης οχημάτων όμως, μεγάλη τιμή της αντικειμενικής συνάρτησης σημαίνει μεγάλο κόστος. Άρα η τιμές των αντικειμενικών συναρτήσεων των λύσεων πρέπει να μετατραπούν.

Μετατροπή τιμών

Βρίσκεται ο γονέας με το μεγαλύτερο κόστος (σε αυτό το παράδειγμα είναι ο γονέας 3 με κόστος 19 μονάδες). Όλα τα κόστη θα αφαιρεθούν από τις 19 μονάδες, έτσι ώστε να φανεί η διαφορά κάθε λύσης από την χειρότερη (κόστος ρουλέτας). Θα δημιουργηθεί στη συνέχεια μια γεννήτρια τυχαίων αριθμών από 0 έως 138 και ανάλογα με το διάστημα που θα ανήκει ο τυχαίος αριθμός, θα επιλεχθεί ο αντίστοιχος γονέας.

Πίνακας 5.11.Μετατροπή διαστήματος τιμών επιλογής για με τη μέθοδο της ρουλέτας

Γονείς	Κόστος	Κόστος ρουλέτας	Κόστος ρουλέτας +1	Από	Έως
1	15	4	4	0	5
2	14	5	5	6	12
3	19	0	1	13	14

4	17	2	3	15	18
5	17	2	3	19	22
6	18	1	2	23	25
7	17	2	3	26	29
8	13	6	7	30	37
9	13	6	7	38	45
10	16	3	4	46	50
11	18	1	2	51	53
12	13	6	7	54	61
13	15	4	5	62	67
14	12	7	8	68	76
15	12	7	8	77	85
16	17	2	3	86	89
17	15	4	5	90	95
18	11	8	9	96	105
19	13	6	7	106	113
20	16	3	4	114	118

21	14	5	6	119	125
22	19	0	1	126	127
23	11	8	9	128	137
24	15	4	5	138	143

Σε δοκιμή με 1000 τυχαίους αριθμούς στο διάστημα 0-143 οι φορές που επιλέχθηκε κάθε γονέας παρουσιάζεται στο διάγραμμα 5.1.

Διάγραμμα 5.1. Συχνότητα επιλογής γονέων σε 1000 τυχαίες δοκιμές

Ο σκοπός της μεθόδου της ρουλέτας είναι οι γονείς με την μεγαλύτερη αξία να έχουν μεγαλύτερες πιθανότητες επιλογής. Αν η διαδικασία λειτουργεί σωστά, οι γονείς με την μεγαλύτερη αξία θα εμφανίζονται περισσότερες φορές και ο γονείς με την μικρότερη αξία θα εμφανίζονται τις λιγότερες. Ακολουθεί ο έλεγχος της διαδικασίας στον πίνακα 5.12.

Πίνακας 5.12. Συχνότητα επιλογής λύσεων με βάση την αξία τους

Αξία	Γονείς	Επιλογές	Σύνολο	Μ.Ο.	Ποσοστό επιλογής
0	3, 22	2	16	8	0,025765
1	6, 11	2	36	18	0,057971
2	4, 5, 7, 16	4	81	20,25	0,065217
3	10, 20	2	50	25	0,080515
4	1, 13, 17, 24	4	96	24	0,077295
5	2, 21	2	91	45,5	0,146538
6	8, 9, 12, 19	4	201	50,25	0,161836
7	14, 15	2	99	49,5	0,15942
8	18, 23	2	140	70	0,225443

Παρατηρείται ότι οι λύσεις 18,23 που έχουν την μεγαλύτερη αξία αποτελούν το 22,54% των επιλογών, το μεγαλύτερο ποσοστό έναντι των υπολοίπων. Άρα η διαδικασία λειτουργεί ικανοποιητικά.

5.6. Επίλυση αλγορίθμου βελτιστοποίησης σμήνους σωματιδίων

Για να επιλυθεί το πρόβλημα με τον αλγόριθμο βελτιστοποίησης σμήνους σωματιδίων, αρχικά χρειάζεται να κωδικοποιηθούν οι λύσεις με κατάλληλο τρόπο. Τα γράμματα που αντιστοιχούν σε κάθε σταθμό θα αντικατασταθούν από αριθμούς και μάλιστα δεκαδικούς, διότι ο αλγόριθμος επιλύει συνεχή προβλήματα.

Κωδικοποίηση

Ο κάθε σταθμός θα αντιστοιχεί σε αριθμό (A-0.1, B-0.3, C-0.5, D-0.7 E-0.9). Ο τελευταίος σταθμός κάθε λύσης «A» παραλείπεται. Οι λύσεις μετατρέπονται σε θέσεις σωματιδίων (πίνακας 5.13).

Πίνακας 5.13. Μετατροπή λύσεων σε θέσεις σωματιδίων

	Λύσεις						Θέσεις σωματιδίων					Κόστος
1	A	D	C	E	B	A	0,1	0,7	0,5	0,9	0,3	15

2	A	B	E	D	C	A	0,1	0,3	0,9	0,7	0,5	14
3	A	E	B	C	D	A	0,1	0,9	0,3	0,5	0,7	19
4	A	E	B	D	C	A	0,1	0,9	0,3	0,7	0,5	17
5	A	B	D	C	E	A	0,1	0,3	0,7	0,5	0,9	17
6	A	B	C	D	E	A	0,1	0,3	0,5	0,7	0,9	18
7	A	E	C	D	B	A	0,1	0,9	0,5	0,7	0,3	17
8	A	D	E	C	B	A	0,1	0,7	0,9	0,5	0,3	13
9	A	C	B	E	D	A	0,1	0,5	0,3	0,9	0,7	13
10	A	E	C	B	D	A	0,1	0,9	0,5	0,3	0,7	16
11	A	E	D	C	B	A	0,1	0,9	0,7	0,5	0,3	18
12	A	D	E	B	C	A	0,1	0,7	0,9	0,3	0,5	13
13	A	B	E	C	D	A	0,1	0,3	0,9	0,5	0,7	15
14	A	D	B	E	C	A	0,1	0,7	0,3	0,9	0,5	12
15	A	C	E	B	D	A	0,1	0,5	0,9	0,3	0,7	12
16	A	C	D	B	E	A	0,1	0,5	0,7	0,3	0,9	17
17	A	E	D	B	C	A	0,1	0,9	0,7	0,3	0,5	15
18	A	C	E	D	B	A	0,1	0,5	0,9	0,7	0,3	11

19	A	B	C	E	D	A	0,1	0,3	0,5	0,9	0,7	13
20	A	D	B	C	E	A	0,1	0,7	0,3	0,5	0,9	16
21	A	C	D	E	B	A	0,1	0,5	0,7	0,9	0,3	14
22	A	D	C	B	E	A	0,1	0,7	0,5	0,3	0,9	19
23	A	B	D	E	C	A	0,1	0,3	0,7	0,9	0,5	11
24	A	C	B	D	E	A	0,1	0,5	0,3	0,7	0,9	15

Υπολογισμός της νέας ταχύτητας (εξίσωση 2.5)και της νέας θέσης (εξίσωση 2.6) του κάθε σωματιδίου (πίνακας 5.14).

Πίνακας 5.14. Υπολογισμός νέας ταχύτητας του κάθε σωματιδίου

	X(t)	U(t)	$c_1 rand_1(P_{best}-x(t))$	$c_2 rand_2(G_{best}-x(t))$	U(t+1)	X(t+1)
1ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	-0,8	-0,8	-0,1
	0,5	0	0	1,6	1,6	2,1
	0,9	0	0	-0,8	-0,8	0,1
	0,3	0	0	0	0	0,3
2ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1
	0,9	0	0	0	0	0,9

	0,7	0	0	0	0	0,7
	0,5	0	0	-0,8	-0,8	-0,3
3ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	-1,6	-1,6	-0,7
	0,3	0	0	2,4	2,4	2,7
	0,5	0	0	0,8	0,8	1,3
	0,7	0	0	-1,6	-1,6	-0,9
4ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	-1,6	-1,6	-0,7
	0,3	0	0	2,4	2,4	2,7
	0,7	0	0	0	0	0,7
	0,5	0	0	-0,8	-0,8	-0,3
5ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1
	0,7	0	0	0,8	0,8	1,5
	0,5	0	0	0,8	0,8	1,3
	0,9	0	0	-2,4	-2,4	-1,5
6ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1
	0,5	0	0	1,6	1,6	2,1
	0,7	0	0	0	0	0,7

	0,9	0	0	-2,4	-2,4	-1,5
7ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	1,6	1,6	2,5
	0,5	0	0	-1,6	-1,6	-1,1
	0,7	0	0	0	0	0,7
	0,3	0	0	0	0	0,3
8ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	0,8	0,8	1,5
	0,9	0	0	0	0	0,9
	0,5	0	0	-0,8	-0,8	-0,3
	0,3	0	0	0	0	0,3
9ο σωματίδιο	0,1	0	0	0	0	0,1
	0,5	0	0	0	0	0,5
	0,3	0	0	2,4	2,4	2,7
	0,9	0	0	-0,8	-0,8	0,1
	0,7	0	0	-1,6	-1,6	-0,9
10ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	-1,6	-1,6	-0,7
	0,5	0	0	1,6	1,6	2,1
	0,3	0	0	1,6	1,6	1,9
	0,7	0	0	-1,6	-1,6	-0,9

11ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	-1,6	-1,6	-0,7
	0,7	0	0	0,8	0,8	1,5
	0,5	0	0	0,8	0,8	1,3
	0,3	0	0	0	0	0,3
12ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	-0,8	-0,8	-0,1
	0,9	0	0	0	0	0,9
	0,3	0	0	1,6	1,6	1,9
	0,5	0	0	-0,8	-0,8	-0,3
13ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1
	0,9	0	0	0	0	0,9
	0,5	0	0	0,8	0,8	1,3
	0,7	0	0	-1,6	-1,6	-0,9
14ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	-0,8	-0,8	-0,1
	0,3	0	0	2,4	2,4	2,7
	0,9	0	0	-0,8	-0,8	0,1
	0,5	0	0	-0,8	-0,8	-0,3
	0,1	0	0	0	0	0,1

15ο σωματίδιο	0,5	0	0	0	0	0,5
	0,9	0	0	0	0	0,9
	0,3	0	0	1,6	1,6	1,9
	0,7	0	0	-1,6	-1,6	-0,9
16ο σωματίδιο	0,1	0	0	0	0	0,1
	0,5	0	0	0	0	0,5
	0,7	0	0	0,8	0,8	1,5
	0,3	0	0	1,6	1,6	1,9
	0,9	0	0	-2,4	-2,4	-1,5
17ο σωματίδιο	0,1	0	0	0	0	0,1
	0,9	0	0	-1,6	-1,6	-0,7
	0,7	0	0	0,8	0,8	1,5
	0,3	0	0	1,6	1,6	1,9
	0,5	0	0	-0,8	-0,8	-0,3
18ο σωματίδιο	0,1	0	0	0	0	0,1
	0,5	0	0	0	0	0,5
	0,9	0	0	0	0	0,9
	0,7	0	0	0	0	0,7
	0,3	0	0	0	0	0,3
19ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1

	0,5	0	0	1,6	1,6	2,1
	0,9	0	0	-0,8	-0,8	0,1
	0,7	0	0	-1,6	-1,6	-0,9
20ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	-0,8	-0,8	-0,1
	0,3	0	0	2,4	2,4	2,7
	0,5	0	0	0,8	0,8	1,3
	0,9	0	0	-2,4	-2,4	-1,5
21ο σωματίδιο	0,1	0	0	0	0	0,1
	0,5	0	0	0	0	0,5
	0,7	0	0	0,8	0,8	1,5
	0,9	0	0	-0,8	-0,8	0,1
	0,3	0	0	0	0	0,3
22ο σωματίδιο	0,1	0	0	0	0	0,1
	0,7	0	0	-0,8	-0,8	-0,1
	0,5	0	0	1,6	1,6	2,1
	0,3	0	0	1,6	1,6	1,9
	0,9	0	0	-2,4	-2,4	-1,5
23ο σωματίδιο	0,1	0	0	0	0	0,1
	0,3	0	0	0,8	0,8	1,1
	0,7	0	0	0,8	0,8	1,5

	0,9	0	0	-0,8	-0,8	0,1
	0,5	0	0	-0,8	-0,8	-0,3
24ο σωματίδιο	0,1	0	0	0	0	0,1
	0,5	0	0	0	0	0,5
	0,3	0	0	2,4	2,4	2,7
	0,7	0	0	0	0	0,7
	0,9	0	0	-2,4	-2,4	-1,5

Τα διανύσματα θέσης θα αντιστοιχηθούν σε κόμβους (πίνακας 5.15). Η μικρότερη τιμή του διανύσματος θα είναι ο κόμβος A, η αμέσως μεγαλύτερη ο κόμβος B, κ.τ.λ.

Πίνακας 5.15. Μετατροπή διανύσματος θέσης σε κόμβους

Σωματίδια	X(t+1)					Αντιστοίχιση σε κόμβους				
1	0,1	-0,1	2,1	0,1	0,3	B	A	E	C	D
2	0,1	1,1	0,9	0,7	-0,3	B	E	D	C	A
3	0,1	-0,7	2,7	1,3	-0,9	C	B	E	D	A
4	0,1	-0,7	2,7	0,7	-0,3	C	A	E	D	B
5	0,1	1,1	1,5	1,3	-1,5	B	C	D	E	A
6	0,1	1,1	2,1	0,7	-1,5	B	D	E	C	A
7	0,1	2,5	-1,1	0,7	0,3	B	E	A	D	C

8	0,1	1,5	0,9	-0,3	0,3	B	E	D	A	C
9	0,1	0,5	2,7	0,1	-0,9	C	D	E	B	A
10	0,1	-0,7	2,1	1,9	-0,9	C	B	E	D	A
11	0,1	-0,7	1,5	1,3	0,3	B	A	E	D	C
12	0,1	-0,1	0,9	1,9	-0,3	C	B	D	E	A
13	0,1	1,1	0,9	1,3	-0,9	B	D	C	E	A
14	0,1	-0,1	2,7	0,1	-0,3	C	B	E	D	A
15	0,1	0,5	0,9	1,9	-0,9	B	C	D	E	A
16	0,1	0,5	1,5	1,9	-1,5	B	C	D	E	A
17	0,1	-0,7	1,5	1,9	-0,3	C	A	D	E	B
18	0,1	0,5	0,9	0,7	0,3	A	C	E	D	B
19	0,1	1,1	2,1	0,1	-0,9	C	D	E	B	A
20	0,1	-0,1	2,7	1,3	-1,5	C	B	E	D	A
21	0,1	0,5	1,5	0,1	0,3	A	D	E	B	C
22	0,1	-0,1	2,1	1,9	-1,5	C	B	E	D	A
23	0,1	1,1	1,5	0,1	-0,3	C	D	E	B	A
24	0,1	0,5	2,7	0,7	-1,5	B	C	E	D	A

Θα υπολογιστεί το κόστος των νέων θέσεων (πίνακας 5.16).

Πίνακας 5.16. Υπολογισμός κόστους νέων θέσεων

	Αρχική μορφή						Προηγούμενα κόστη	Νέα κόστη
1	A	E	C	D	B	A	15	17
2	A	B	E	D	C	A	14	14
3	A	C	B	E	D	A	19	13
4	A	E	D	B	C	A	17	15
5	A	B	C	D	E	A	17	18
6	A	B	D	E	C	A	18	11
7	A	D	C	B	E	A	17	19
8	A	C	B	E	D	A	13	13
9	A	C	D	E	B	A	13	14
10	A	C	B	E	D	A	16	13
11	A	E	D	C	B	A	18	18
12	A	C	B	D	E	A	13	15
13	A	B	D	C	E	A	15	17
14	A	C	B	E	D	A	12	13

15	A	B	C	D	E	A	12	18
16	A	B	C	D	E	A	17	18
17	A	D	E	B	C	A	15	13
18	A	C	E	D	B	A	11	11
19	A	C	D	E	B	A	13	14
20	A	C	B	E	D	A	16	13
21	A	D	E	B	C	A	14	13
22	A	C	B	E	D	A	19	13
23	A	C	D	E	B	A	11	14
24	A	B	C	E	D	A	15	13

Τα νέα κόστη θα συγκριθούν με τα αρχικά (πίνακας 5.17).

Πίνακας 5.17. Σύγκριση αρχικούς κόστους και τελικού για την κάθε λύση

	Προηγούμενα κόστη	Νέα κόστη	
1	15	17	Αύξηση
2	14	14	Σταθερό/μείωση
3	19	13	Σταθερό/μείωση

4	17	15	Σταθερό/μείωση
5	17	18	Αύξηση
6	18	11	Σταθερό/μείωση
7	17	19	Αύξηση
8	13	13	Σταθερό/μείωση
9	13	14	Αύξηση
10	16	13	Σταθερό/μείωση
11	18	18	Σταθερό/μείωση
12	13	15	Αύξηση
13	15	17	Αύξηση
14	12	13	Αύξηση
15	12	18	Αύξηση
16	17	18	Αύξηση
17	15	13	Σταθερό/μείωση
18	11	11	Σταθερό/μείωση
19	13	14	Αύξηση
20	16	13	Σταθερό/μείωση

21	14	13	Σταθερό/μείωση
22	19	13	Σταθερό/μείωση
23	11	14	Αύξηση
24	15	13	Σταθερό/μείωση

Παρατηρείται ότι 14 τιμές μειώθηκαν ή έμειναν σταθερές και 10 αυξήθηκαν, δηλαδή υπάρχει βελτίωση στην πλειοψηφία των λύσεων.

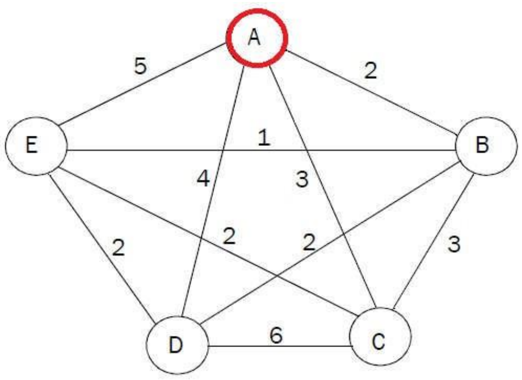
5.7. Επίλυση αλγορίθμου πλησιέστερου γείτονα

Επίλυση του προβλήματος με τον αλγόριθμο πλησιέστερου γείτονα ή αλλιώς k-means.

<ins>Αρχικοποίηση</ins>

U=[A]

C=0

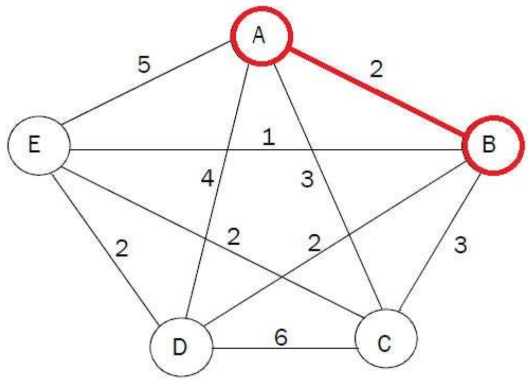


<ins>1η επανάληψη</ins>

min{AB, AC, AD, AE}=min{2, 3, 4, 5}=2 = (AB)

U=[A, B]

C=2

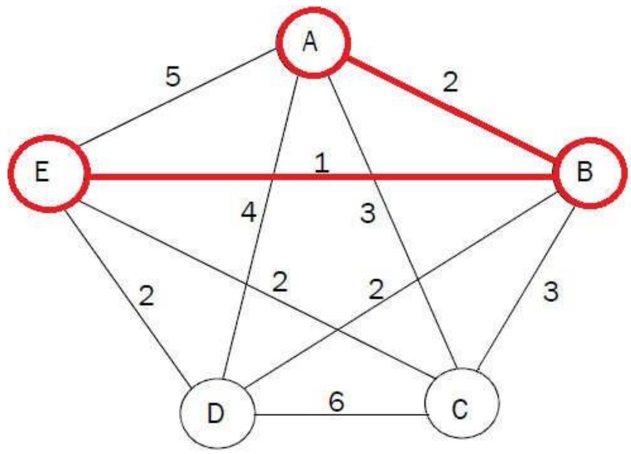


2^{η} επανάληψη

min{BE, BD, BC}=min{1, 2, 3}=1 = (BE)

U=[A, B, E]

C=3

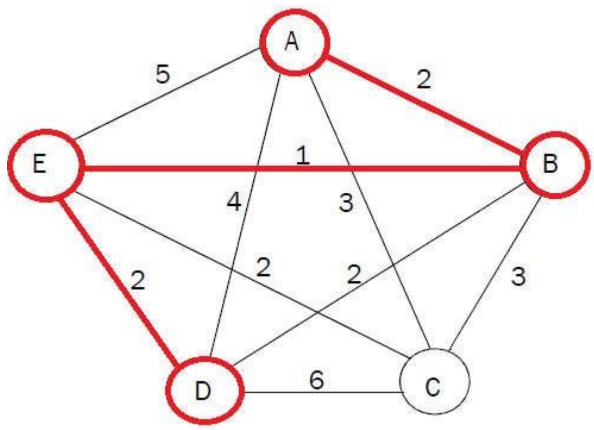


3^{η} επανάληψη

min{EC, ED }=min{2, 2}=2 = (AD)

U=[A, B, E, D]

C=5

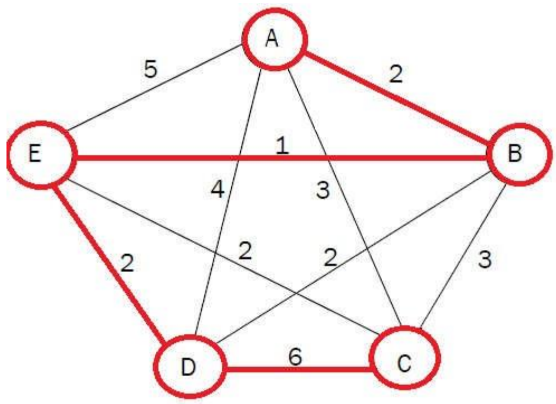


4^{η} επανάληψη

min{DC}=min{6}=6 = (DC)

U=[A, B, E, D, C]

C=11

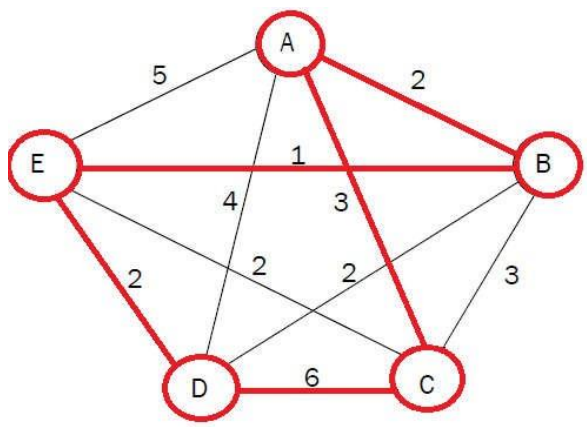


5^{η} επανάληψη

min{CA}=min{3}=3 = (CA)

U=[A, B, E, D, C, A]

C=14



Πίνακας 5.18. Αποτελέσματα αλγορίθμου πλησιέστερου γείτονα

Επανάληψη	Κόμβοι	Κόστος	Συνολικό κόστος
Αρχικοποίηση	A	-	-
1	A, B	2	2
2	A, B, E	1	3
3	A, B, E, D	2	5
4	A, B, E, D, C	6	11
5	A, B, E, D, C, A	3	14

Ο αλγόριθμος τερματίζεται στην 5η επανάληψη και ως βέλτιστη λύση έχει υπολογίσει την διαδρομή A, B, E, D, C, A με κόστος 14. Η λύση αυτή όπως και κάποιων προηγούμενων μεθόδων δεν είναι το ολικό βέλτιστο.

Βιβλιογραφία

[1] McCulloch, W. Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics* 5, 115-113.

[2] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs, *Numerische Mathematik* 1, 269–271.

[3] Bellman, Richard (1958). On a routing problem, *Quarterly of Applied Mathematics*. 16: 87–90.

[4]Ford, Lester R. Jr. (1956). Network Flow Theory. Paper P-923.

[5] Shimbel, A. (1955). Structure in communication nets. Proceedings of the Symposium on Information Networks, *Polytechnic Press of the Polytechnic Institute of Brooklyn*, 199–203.

[6] Ford, L. Fulkerson, D. (1962). *Flows in Networks*, The Rand Corporate Publisher.

[7] Kruskal, J. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem, *Proceedings of the American Machematical Society* 7, 48-50

[8] Prim, R. (1957). Shortest connection networks and some generalizations, *Bell System Technical Journal* 36, 1389-1401.

[9] Land, A. Doig, G. (1960). An automatic method of solving discrete programming problems, *Ecometrica,* Vol. 28, no. 3. pp. 487-520.

[10] Kirkpatrick, S. Gelatt, C. Vecchi, M. (1982). Optimization by Simulated Annealing, *Science*, Vol. 220, 671-680.

[11] Glover, F. (1989). Tabu Search I, *ORSA Journal of computing*, 1(3), 190-206.

[12] Holland, J. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI.

[13] Kennedy, J. Eberhart, R. (1995). Particle Swarm Optimization, Proceedings of 1995 IEEE International Conference on Neural Networks, 4, 1942-1948

[14] Fix, E. Hodges, J. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties

[15] Rumelhart, D. Hinton, G. Williams, R. (1986). Learning Representations by back-propagating errors. *Nature*, 323 (6088): 533-536.

[16] Cortes, C. Vapnik, V. (1995). Support Vector networks, *Machine Learning*, 20 (3): 273-297.

[17] MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. Vol. 1. University of California Press. pp. 281-297.

[18] Biggs, Lloyd, Wilson, (1986). A discussion of the early work of Hamilton and Kirkman. *Graph Theory*.

[19] Dantzig, G. Ramser, J. (1959). The truck Dispatching Problem. Management Science, 6, 80-91

[20] Knoll, D. Pruglmeier, M. Reinhart, G. (2016). Predicting Future Inbound Logistics Processes using Machine Learning. Procedia CIRP, Vol. 52, pp. 145-150

[21] Hess, A. Spinler, S. Winkenbach, M. (2021). Real-time demand forecasting for an urban delivery platform. Transp. Res. Part E Logist.

Transp. Rev. 2021, 145, 102147.

[22] Ashikuzzaman, M. Akram, W. (2021). PSO-ANN in preventing traffic collisions: a comparative study. *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 24, No. 3, pp. 1796-1803.

[23] Μαρινάκης, Ι, Μαρινάκη, Μ. Ματσατσίνης, Ν. Ζοπουνίδης, Κ. (2011). Μεθευρετικοί και Εξελικτικοί Αλγόριθμοι σε Προβλήματα Διοικητικής Επιστήμης, Μέρος Ι, pp. 30-33