ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

**ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

**ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗ ΚΑΙ ΔΙΚΤΥΑ**

**ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

# ΜΕΛΕΤΗ ΚΑΙ ΥΠΟΛΟΓΙΣΤΙΚΗ ΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΞΕΤΑΣΕΩΝ ΣΕ ΠΑΝΕΠΙΣΤΗΜΙΑΚΑ ΙΔΡΥΜΑΤΑ ΧΩΡΙΣ ΠΕΡΙΟΡΙΣΜΟΥΣ ΧΩΡΗΤΙΚΟΤΗΤΑΣ

Τάσιος Βασίλειος

Επιβλέπων καθηγητής: Γκόγκος Χρήστος

Άρτα, Ιούνιος 2021

**STUDYING AND IMPLEMENTING A SOLUTION OF UNCAPACITED UNIVERSITY EXAMINATION TIMETABLING PROBLEM**

## Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, Πέμπτη 3/2/2022

## ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπουσα καθηγήτρια

Γκόγκος Χρήστος,

Αναπληρωτής Καθηγητής

2. Μέλος επιτροπής

Γεωργία Φουτσιτζή

Καθηγήτρια

3. Μέλος επιτροπής

Τζάλλας Αλέξανδρος

Επίκουρος Καθηγητής

Ο/Η Διευθυντής/τρια του ΠΜΣ

Γκόγκος Χρήστος,

Αναπληρωτής Καθηγητής

Υπογραφή

# Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Επίθετο, Όνομα

Τάσιος, Βασίλειος

Υπογραφή

# Περιεχόμενα

# Abstract

Η αποδοτική δημιουργία προγραμμάτων εξετάσεων είναι ένα σημαντικό και επαναλαμβανόμενο πρόβλημα το οποίο καλούνται να αντιμετωπίσουν τα εκπαιδευτικά ιδρύματα σε όλο τον κόσμο. Μια απλοποιημένη μορφή του προβλήματος έχει προταθεί το 1996 από τους Carter, Laporte και Lee οι οποίοι διέθεσαν δημόσια 13 στιγμιότυπα προβλημάτων που εν συνεχεία χρησιμοποιήθηκαν σε πληθώρα επιστημονικών εργασιών χρονοπρογραμματισμού. Στα πλαίσια της παρούσας διπλωματικής εργασίας θα δημιουργηθούν ενδεικτικές λύσεις για τα 13 στιγμιότυπα και έπειτα θα γίνει μια διαδικασία βελτιστοποίησης για αυτές τις αρχικές λύσεις.


# 1. Εισαγωγή

Ως επαναλαμβανόμενη διοικητική δραστηριότητα συνήθως στο τέλος κάθε εξαμήνου στα ακαδημαϊκά ιδρύματα, το καθήκον του χρονοδιαγράμματος των εξετάσεων απαιτεί μια μη τετριμμένη εργασία που μπορεί να διαρκέσει έως και εβδομάδες ή μήνες για να δημιουργήσει ένα ικανοποιητικό πρόγραμμα. Ως εκ τούτου, τα περισσότερα ακαδημαϊκά ιδρύματα προτείνουν να αυτοματοποιήσουν αυτήν την εργασία χρησιμοποιώντας ένα αποτελεσματικό λογισμικό ικανό να εκπληρώσει τις απαιτήσεις χρονοδιαγράμματος που φροντίζει τις προτιμήσεις των φοιτητών και των καθηγητών. Το πρόβλημα του χρονοδιαγράμματος των εξετάσεων έχει μελετηθεί έντονα επειδή είναι ένα πολύπλοκο πρόβλημα και έχει πρακτική σημασία στα εκπαιδευτικά ιδρύματα. Στο υπολογιστικό πλαίσιο, ο προγραμματισμός των εξετάσεων είναι ένα συνδυαστικό πρόβλημα βελτιστοποίησης που αντιμετωπίζεται με την ανάθεση ενός συνόλου εξετάσεων σε συγκεκριμένους πόρους, έτσι ώστε οι χρονοθυρίδες και οι αίθουσες εξετάσεων να ανταποκρίνονται σε συγκεκριμένους αυστηρούς και χαλαρούς περιορισμούς. Η εκπλήρωση σκληρών περιορισμών στο χρονοδιάγραμμα των εξετάσεων είναι υποχρεωτική για τη δημιουργία ενός εφικτού χρονοδιαγράμματος, ενώ οι ικανοποιητικοί περιορισμοί ικανοποίησης μπορούν να παραβιαστούν, αλλά η εκπλήρωσή τους βελτιώνει την ποιότητα της λύσης του χρονοδιαγράμματος των εξετάσεων. Ο βασικός στόχος για κάθε χρονοδιάγραμμα είναι η δημιουργία ενός εφικτού χρονοδιαγράμματος με καλή ποιότητα. Το πρόβλημα του χρονοδιαγράμματος των εξετάσεων είναι ένα πραγματικό πρόβλημα συνδυαστικής βελτιστοποίησης που είναι δύσκολο να επιλυθεί λόγω του ότι έχει πολλούς περιορισμούς και περιορισμένους πόρους (δηλ. χρονικά διαστήματα και αίθουσες) στην κατανομή μεγάλου αριθμού εξετάσεων. Υπάρχουν δύο τύποι προβλημάτων χρονοδιαγράμματος εξέτασης: χωρητικό και μη χωρητικό. Στην μη χωρητική έκδοση, η χωρητικότητα των αιθουσών δεν λαμβάνεται υπόψη, ενώ η χωρητική παραλλαγή θεωρεί την χωρητικότητα δωματίου ως σκληρό περιορισμό. Το χρονοδιάγραμμα των εξετάσεων είναι ένας τύπος προβλήματος προγραμματισμού που έχει πολυπλοκότητα NP-hard και έτσι πολλοί ερευνητές έχουν διερευνήσει στοχαστικές μεθόδους όπως ευρετικές και μετα-ευρετικές προσεγγίσεις για να βρουν βέλτιστες ή σχεδόν βέλτιστες λύσεις. Μέχρι πρόσφατα, στην επιστημονική βιβλιογραφία έχουν προταθεί πολλές προσεγγίσεις για την επίλυση του προβλήματος του χρονοδιαγράμματος των εξετάσεων.

## 1.1 Προβλήματα χρονοδιαγράμματος εξετάσεων

Το πρόβλημα του χρονοδιαγράμματος των εξετάσεων είναι τα πιο κοινά προβλήματα που αντιμετωπίζουν συνεχώς τα ακαδημαϊκά ιδρύματα. Το πρόβλημα ξεκινά όταν πρέπει να εκχωρήσουν ένα σύνολο εξετάσεων σε περιορισμένο αριθμό χρονικών ορίων. Οι Carter και Laporte (1996) καθόρισαν το πρόβλημα των χρονοδιαγραμμάτων των πανεπιστημιακών εξετάσεων ως:

"Η ανάθεση εξετάσεων σε περιορισμένο αριθμό διαθέσιμων χρονικών περιόδων με τέτοιο τρόπο ώστε να μην υπάρχουν συγκρούσεις ".

Το πρόβλημα του χρονοδιαγράμματος των εξετάσεων ορίζεται ουσιαστικά ως η κατανομή των εξετάσεων σε περιορισμένο αριθμό χρονικών θυρίδων, ενώ ταυτόχρονα πληροί τον μέγιστο αριθμό περιορισμών που διαφέρουν σημαντικά από ίδρυμα σε ίδρυμα. Έτσι, τα προβλήματα των χρονοδιαγραμμάτων των εξετάσεων διαφέρουν ως προς το μέγεθος, την πολυπλοκότητα και τους περιορισμούς τους . Στη βιβλιογραφία χρονοδιαγράμματος, παρουσιάστηκαν δύο κατηγορίες περιορισμών, δηλαδή οι χαλαροί περιορισμοί και οι σκληροί περιορισμοί (Alzaqebah and Abdullah, 2014; Carter and Laporte, 1996). Αυτοί οι τύποι περιορισμών εξηγούνται ως εξής:

- Σκληροί περιορισμοί είναι εκείνοι που δεν είναι σε θέση να παραβιάσουν σε καμία περίπτωση. Ως παραδείγματα:
    - Δύο εξετάσεις δεν μπορούν να προγραμματιστούν σε μία χρονοθυρίδα όταν υπάρχουν αρκετοί συνηθισμένοι μαθητές που συμμετέχουν και στις δύο εξετάσεις.
    - Ο αριθμός των μαθητών που συμμετέχουν στις εξετάσεις δεν πρέπει να υπερβαίνει τον αριθμό των διαθέσιμων θέσεων.
- Οι χαλαροί περιορισμοί είναι εκείνοι που είναι επιθυμητό να ικανοποιηθούν, αλλά δεν είναι απολύτως απαραίτητοι. Ως παραδείγματα:
    - Οι εξετάσεις που συγκρούονται είναι καλύτερα να χωριστούν καθ 'όλη τη διάρκεια της εξεταστικής περιόδου, αποφεύγοντας διαδοχικές χρονικές θέσεις των εξετάσεων ή δύο εξετάσεις την ίδια ημέρα.
    - Οι εξετάσεις που θεωρείται ότι έχουν τους περισσότερους μαθητές θα πρέπει να προγραμματιστούν το συντομότερο δυνατό ώστε να υπάρχει επαρκής χρόνος σήμανσης.
    - Απαιτείται προτεραιότητα των εξετάσεων.

Μια λύση σε ένα πρόβλημα χρονοδιαγράμματος εξέτασης που τηρεί τους σκληρούς περιορισμούς ονομάζεται εφικτό χρονοδιάγραμμα. Λόγω του βαθμού δυσκολίας στην επίλυση του προβλήματος του χρονοδιαγράμματος, ορισμένοι από τους χαλαρούς περιορισμούς μπορεί να χρειαστεί να παραμεριστούν, καθώς είναι σχεδόν αδύνατο να τηρήσει αυτούς τους περιορισμούς στη δημιουργία λύσεων. Έτσι το πρόβλημα χρονοδιαγράμματος εξετάσεων, γίνεται στην πραγματικότητα ένα πρόβλημα βελτιστοποίησης, και συνήθως ενσωματώνει έναν τεράστιο χώρο αναζήτησης λύσεων στον οποίο βρίσκονται τοπικές βέλτιστες λύσεις, τις οποίες είναι ζητούμενο να βρούμε και να συγκρίνουμε. Αυτό καθιστά δύσκολο να αντιμετωπιστεί με τις παραδοσιακές μεθόδους.

Τα προβλήματα χρονοδιαγράμματος των πανεπιστημιακών εξετάσεων χωρίζονται στα χωρητικά προβλήματα χρονοδιαγράμματος των εξετάσεων και στα μη χωρητικά προβλήματα χρονοδιαγράμματος(U-UETP).Το (U-UETP) θα ληφθεί υπόψη επειδή υπάρχουν πολλές προσεγγίσεις στη βιβλιογραφία που χρησιμοποίησαν αυτό το πρόβλημα για να αξιολογήσουν τις προσεγγίσεις τους και αυτό το πρόβλημα εξακολουθεί να είναι ενεργός ερευνητικός τομέας.

## 1.2. Μη χωρητικά προβλήματα χρονοδιαγράμματος (U-UETP)

Έχει παρατηρηθεί ότι το πρόβλημα του χρονοδιαγράμματος και η εξέτασή του ενδέχεται να μην έχουν ή να έχουν χωρητικότητα ανάλογα με τους παράγοντες της εξέτασης. Ωστόσο, σε αυτή τη μελέτη εξετάζεται μόνο το Πρόβλημα Χρονοδιαγράμματος Εξετάσεων Πανεπιστημίου χωρίς χωρητικότητα (U-UETP). Αυτό οφείλεται στο γεγονός ότι το U-UETP δεν λαμβάνει υπόψη τις χωρητικότητες των δωματίων, ενώ το

πρόβλημα του χρονοδιαγράμματος χωρητικότητας εξετάσεων πρέπει να περιλαμβάνει τους σκληρούς περιορισμούς με τρόπο που ο αριθμός των μαθητών σε ένα συγκεκριμένο δωμάτιο στους οποίους έχουν εκχωρηθεί, δεν πρέπει κατά τη διάρκεια της περιόδου ως ανά προγραμματισμένο υπερβαίνει τη χωρητικότητα δωματίου.

Τις τελευταίες δεκαετίες, μια μεγάλη ποικιλία τεχνικών προσέγγισης για την επίλυση του U-UETP έχει δημιουργηθεί από τις κοινότητες Τεχνητής Νοημοσύνης (AI) και Επιχειρησιακής Έρευνας (OR). Μια εκτενής και εξαντλητική περίληψη αυτών των τεχνικών παρέχεται από τους Qu et al. (2009α). Οι τεχνικές αυτές, με βάση τις μεθόδους χρωματισμού γραφημάτων, ανέθεταν εξετάσεις σε χρονικές θυρίδες, με βάση το επίπεδο δυσκολίας, δηλαδή με ανάλογα τον αριθμό των συγκρούσεων που μπορεί να δημιουργούν. Ως προσέγγιση ανάκαμψης του χρονοδιαγράμματος με μη προγραμματισμένες εξετάσεις, χρησιμοποιείται μια μέθοδος backtracking με αυτές τις τεχνικές. Carter et al. (1996) ξεκίνησε την κύρια έρευνα για το U-UETP ενσωματώνοντας αρκετές ευρετικές μεθόδους χρωματισμού γραφημάτων στο U-UETP (Carter et al., 1996). Άλλες μελέτες που ενσωματώνουν ευρετικές μεθόδους χρωματισμού γραφημάτων για το U-UETP έχουν χρησιμοποιηθεί επίσης (Burke et al., 2004; Asmuni et al., 2009; Aldeeb BA et al., 2015b).

Ο κύριος στόχος για οποιαδήποτε προσέγγιση αποσύνθεσης του χρονοδιαγράμματος εξέτασης είναι να παρέχει μια εφικτή λύση με τις παραβιάσεις που βασίζονται στους μαλακούς περιορισμούς με τους λιγότερους αριθμούς που μπορούν να επισημανθούν εδώ ως "βέλτιστη λύση". Η ποιότητα ενός εφικτού χρονοδιαγράμματος βελτιστοποιείται χρησιμοποιώντας διαφορετικές προσεγγίσεις βελτιστοποίησης. Σε πολλές περιπτώσεις, η ποιότητα του χρονοδιαγράμματος αξιολογείται από τη συνάρτηση με ποινή που αντικατοπτρίζει τον βαθμό κατά τον οποίο παραβιάζονται οι απαλοί περιορισμοί. Για παράδειγμα, μια λύση που αποτελεί μικρότερο αριθμό παραβιάσεων στις προτιμήσεις των εξεταζόμενων μαθητών, όπως ένα κενό χρόνου μεταξύ των εξετάσεων που θα είναι επαρκή για να επιτρέψει στον μαθητή να διαβάσει για το επόμενο μάθημα, θα αξιολογηθεί ανταγωνιστικό από αυτό με μεγαλύτερο αριθμό παραβιάσεων (Aldeeb BA et al., 2014a).

## 2. Περιγραφή Προβλήματος

Το UETP θεωρείται ως σημαντικό διοικητικό μέλημα σε πολλά εκπαιδευτικά ιδρύματα. Έχει αντιμετωπιστεί ως ένα δύσκολο πρόβλημα βελτιστοποίησης, στο οποίο ο απαιτούμενος υπολογισμός για την επίλυση του προβλήματος αυξάνεται εκθετικά με το μέγεθος του. Τα προβλήματα του χρονοδιαγράμματος των εξετάσεων θεωρούνται ως ενεργός ερευνητικός τομέας που έχει αποκτήσει την προσοχή των ερευνητών πεδίων τεχνητής νοημοσύνης και επιχειρησιακής έρευνας. Πρόκειται για ένα NP πλήρες πρόβλημα, στο οποίο οι ακριβείς προσεγγίσεις δεν εφαρμόζονται για την εξεύρεση μιας (σχεδόν) βέλτιστης λύσης λόγω του υπολογιστικού χρόνου που απαιτείται αυξάνεται εκθετικά σε σχέση με το μέγεθος του προβλήματος. Έχουν προσαρμοστεί αρκετές προσεγγίσεις για την επίλυση των προβλημάτων UETP.

Το πρόβλημα αφορά εξετάσεις, σπουδαστές και συνεχόμενες περιόδους σε κάθε μια από τις οποίες μπορούν να διεξαχθούν μια ή περισσότερες εξετάσεις. Κάθε εξέταση διαθέτει μια λίστα από σπουδαστές και κάθε σπουδαστής μπορεί να είναι εγγεγραμμένος σε μια ή περισσότερες εξετάσεις. Η λύση του προβλήματος συνίσταται στην ανάθεση εξετάσεων σε περιόδους έτσι ώστε να μην υπάρχουν συγκρούσεις, δηλαδή να μην υπάρχουν σπουδαστές που θα έπρεπε να συμμετάσχουν σε εξετάσεις σε περισσότερα του ενός μαθήματα στην ίδια περίοδο. Καθώς είναι ενδεχόμενο να υπάρχουν πολλά εναλλακτικά προγράμματα που ικανοποιούν τον ανωτέρω περιορισμό, προτιμότερο θεωρείται εκείνο το πρόγραμμα που διαθέτει επαρκή διαστήματα προετοιμασίας ανάμεσα σε διαδοχικές εξετάσεις για όλους τους φοιτητές συνολικά.

Ειδικότερα, ορίζονται τιμές ποινής που είναι 16, 8, 4, 2 ή 1 σε κάθε περίπτωση που ένας φοιτητής συμμετέχει σε δύο εξετάσεις που απέχουν 1, 2, 3, 4 ή 5 περιόδους αντίστοιχα. Η συνολική ποινή για όλους τους φοιτητές, διαιρεμένη με το πλήθος των φοιτητών αποτελεί το κόστος της λύσης.

Ακολουθεί μια επίσημη περιγραφή του προβλήματος ως πρόβλημα προγραμματισμού ακέραιου αριθμού. Σε αυτήν τη πλευρά του προβλήματος, C είναι το σύνολο των μαθημάτων, P είναι το σύνολο των περιόδων και S είναι το σύνολο των μαθητών. Κάθε μάθημα c έχει έναν αριθμό εγγραφών που καταγράφονται στο $E_c$. Παρόμοια, κάθε μαθητής έχει ένα σύνολο μαθημάτων C στα οποία είναι εγγεγραμμένος. Το σύνολο CNF περιέχει τριάδες της μορφής $c_1$, $c_2$, co που αντιπροσωπεύουν την κατάσταση στην οποία το μάθημα $c_1$ και το μάθημα $c_2$ έχουν πλήθος co κοινούς μαθητές.

Για να καθοριστεί για κάθε μάθημα η περίοδος που θα προγραμματιστεί, χρησιμοποιούνται οι μεταβλητές απόφασης $m_{cp}$ για κάθε c που ανήκει στα μαθήματα C και p που ανήκει στο διάστημα 1:: P που είναι δυαδικές μεταβλητές που υποθέτουν την τιμή 1 εάν το μάθημα c έχει προγραμματιστεί στην περίοδο p ή 0 διαφορετικά. Επιπλέον, οι δυαδικές μεταβλητές $y1_{t,p}$ ορίζονται για κάθε t που ανήκει στα CNF και p στο διάστημα 1:: |P - 1| υποθέτοντας την τιμή 1 όταν τα μαθήματα $t.c_1$ και $t.c_2$ προγραμματίζονται σε διαδοχικές περιόδους και 0 διαφορετικά. Ομοίως, οι δυαδικές μεταβλητές $y2_{t,p}$ , $y3_{t,p}$ , $y4_{t,p}$, $y5_{t,p}$ ορίζονται και λαμβάνουν την τιμή 1 όταν τα μαθήματα $t.c_1$ και $t.c_2$ έχουν προγραμματιστεί σε περιόδους με απόσταση μεταξύ 2 και 5 περιόδων αντίστοιχα.

Η αντικειμενική συνάρτηση λοιπόν γίνεται ως εξής:

$$\text{minimize} \sum_{t \in CNF} t.\text{co} \left( 16 \sum_{p \in 1 \ldots |P-1|} y1_{t,p} + 8 \sum_{p \in 1 \ldots |P-2|} y2_{t,p} + 4 \sum_{p \in 1 \ldots |P-3|} y3_{t,p} + 2 \sum_{p \in 1 \ldots |P-4|} y4_{t,p} + \sum_{p \in 1 \ldots |P-5|} y5_{t,p} \right)$$

Ο πρώτος περιορισμός δηλώνει ότι κάθε μάθημα θα πρέπει να προγραμματιστεί σε μία ακριβώς περίοδο και αποτυπώνεται στην εξίσωση:

$$\sum_{p \in P} m_{c,p} = 1 \forall c \in C$$

Ο δεύτερος περιορισμός που φαίνεται στην παρακάτω εξίσωση απαγορεύει κάθε πιθανό ζεύγος μαθημάτων με κοινούς μαθητές να προγραμματίζονται την ίδια περίοδο.

$$m_{t.c_1,p} + m_{t.c_2,p} \leq 1 \forall p \in P, \forall t \in CNF$$

Το τρίτο σύνολο των εξισώσεων περιορισμών καθορίζει τις μεταβλητές $y1_{t,p}$ $y2_{t,p}$ , $y3_{t,p}$ , $y4_{t,p}, y5_{t,p}$ :

$$\left.\begin{aligned} y1_{t,p} &\geq m_{t.c_1,p} + m_{t.c_2,p+1} - 1 \\ y1_{t.p} &\geq m_{t.c_2,p} + m_{t.c_1,p+1} - 1 \end{aligned}\right\} \forall t \in CNF, \forall p \in 1 \dots |P| - 1$$

$$\left.\begin{aligned} y2_{t,p} &\geq m_{t.c_1,p} + m_{t.c_2,p+2} - 1 \\ y2_{t,p} &\geq m_{t.c_2,p} + m_{t.c_1,p+2} - 1 \end{aligned}\right\} \forall t \in CNF, \forall p \in 1 \dots |P| - 2$$

$$\left.\begin{aligned} y3_{t,p} &\geq m_{t.c_1,p} + m_{t.c_2,p+3} - 1 \\ y3_{t,p} &\geq m_{t.c_2,p} + m_{t.c_1,p+3} - 1 \end{aligned}\right\} \forall t \in CNF, \forall p \in 1 \dots |P| - 3$$

$$\left.\begin{aligned} y4_{t,p} &\geq m_{t.c_1,p} + m_{t.c_2,p+4} - 1 \\ y4_{t,p} &\geq m_{t.c_2,p} + m_{t.c_1,p+4} - 1 \end{aligned}\right\} \forall t \in CNF, \forall p \in 1 \dots |P| - 4$$

$$\left.\begin{aligned} y5_{t,p} &\geq m_{t.c_1,p} + m_{t.c_2,p+5} - 1 \\ y5_{t,p} &\geq m_{t.c_2,p} + m_{t.c_1,p+5} - 1 \end{aligned}\right\} \forall t \in CNF, \forall p \in 1 \dots |P| - 5$$

Όταν τα μαθήματα $c_1$ και $c_2$ έχουν απόσταση 1 είτε $m_{t.c_1,p}$ και $m_{t.c_2,p+1}$ είτε $m_{t.c_2,p}$ και $m_{t.c_2,p+1}$ λαμβάνουν τιμή 1. Εάν συμβαίνει αυτό, η μεταβλητή $y1_t$ θα είναι 1. Εάν τουλάχιστον ένα από τα $m_{t.c_1,p}$ ή $m_{t.c_2,p+1}$ είναι 0, η τιμή του $y1_t$ θα μπορούσε να είναι 0 ή 1, αλλά εφόσον περιλαμβάνεται στον στόχο που ελαχιστοποιείται, θα λάβει την τιμή 0.

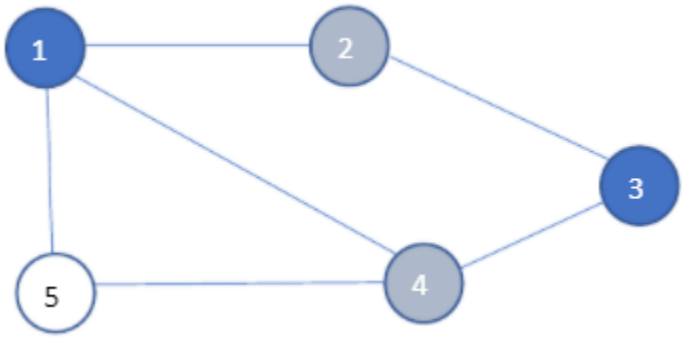
# 3. Αναπαράσταση μοντέλων για μη χωρητικά προβλήματα χρονοδιαγράμματος εξετάσεων πανεπιστημίου

Τα μοντέλα χρωματισμού γραφημάτων και οι συμμετρικοί δυαδικοί πίνακες συγκρούσεων προτείνονται για να αντιπροσωπεύουν τα προβλήματα χρονοδιαγράμματος των πανεπιστημιακών εξετάσεων. Η αρχή αυτών των μοντέλων είναι να παράγουν εφικτές λύσεις χωρίς να λαμβάνονται υπόψη οι μαλακοί περιορισμοί. Οι ακόλουθες υπό-ενότητες περιγράφουν πώς μπορεί να μοντελοποιηθεί ένα πρόβλημα χρονοδιαγράμματος χρησιμοποιώντας αυτά τα μοντέλα.

## 3.1 Χρωματισμός γραφήματος

Η δημιουργία ενός εφικτού χρονοδιαγράμματος εξετάσεων, ώστε να βρίσκονται όσο το δυνατόν λιγότερες συγκρούσεις, είναι παρόμοιο με την επίλυση του προβλήματος χρωματισμού γραφημάτων, όπου πρέπει να βρίσκονται όσο το δυνατόν λιγότερα χρώματα. Ορισμένες ερευνητικές εργασίες, τις τελευταίες δεκαετίες έχουν προτείνει διάφορα μοντέλα και διατυπώσεις για το πρόβλημα του χρονοδιαγράμματος των εξετάσεων. Για παράδειγμα, ο de Werra (1985) περιγράφει πώς ένα γράφημα μπορεί να μοντελοποιηθεί από ένα πρόβλημα χρονοδιαγράμματος. Ο κύριος στόχος του χρωματισμού γραφημάτων είναι η ελαχιστοποίηση του αριθμού των χρωμάτων που μπορούν να χρησιμοποιηθούν για τον χρωματισμό των κόμβων των γραφημάτων όσο το δυνατόν περισσότερο, λαμβάνοντας υπόψη ότι κανένας από τους γειτονικούς κόμβους δεν έχει το ίδιο χρώμα. Οι τιμές των χρωμάτων είναι γνωστές ως ο «χρωματικός αριθμός» του γραφήματος. Αυτό μπορεί να συνδεθεί με το πρόβλημα του χρονοδιαγράμματος των εξετάσεων, όπου:

- Κάθε εξέταση αντιπροσωπεύεται από έναν κόμβο.
- Η σύγκρουση μεταξύ των εξετάσεων αντιπροσωπεύεται από τις ακμές μεταξύ κόμβων, π.χ. οι γειτονικές εξετάσεις μοιράζονται μερικούς μαθητές.
- Οι χρονικές θέσεις αντιπροσωπεύονται από χρώματα κόμβων.




*Εικόνα 1. Παράδειγμα γραφήματος G*

Για παράδειγμα, στην Εικόνα 1, το UETP παρουσιάζεται χρησιμοποιώντας ένα γράφημα G = (V, E) όπου, V είναι ο συνολικός αριθμός των εξετάσεων (κόμβοι) που πρόκειται να προγραμματιστούν και το E αντιπροσωπεύει τις γειτονικές εξετάσεις μεταξύ κάθε δύο εξετασμένων εξετάσεων. Στην Εικόνα 1, οι εξετάσεις 1 και 2 χρωματίζονται διαφορετικά επειδή είναι γειτονικές και συνδέονται χρησιμοποιώντας μια ακμή. Αυτό σημαίνει ότι αυτές οι δύο εξετάσεις έχουν κοινούς μαθητές. Πρακτικά, σε αυτό το παράδειγμα, η λύση χρονοδιαγράμματος εξέτασης αναπαρίσταται ως ένα διάνυσμα: x = (1, 2, 1, 2, 3), που σημαίνει ότι

η εξέταση 1 έχει προγραμματιστεί για το χρονικό σημείο 1, για τις εξετάσεις 2 στο χρονικό σημείο 2 και για την εξέταση 3 στο χρονικό διάστημα 1.

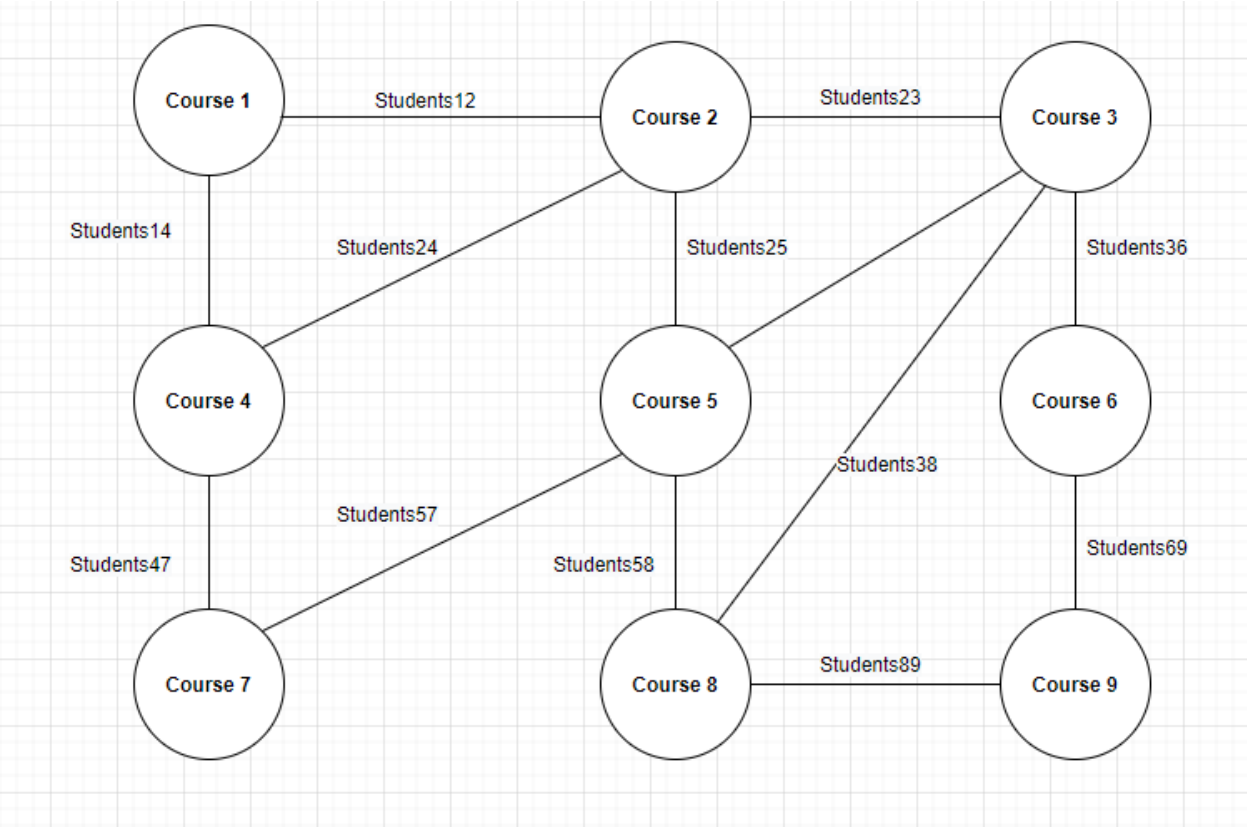
Έτσι τελικά η αντιστοίχιση από πρόβλημα χρονοδιαγράμματος σε χρωματισμό γραφήματος συνοψίζεται όπως φαίνεται στον Πίνακα 1.

*Πίνακας 1. Αντιστοίχιση*

| **Πρόβλημα Χρονοδιαγράμματος** | **Πρόβλημα Χρωματισμού Γραφήματος** |
|---|---|
| Μάθημα (course) | Κόμβος |
| Σπουδαστές (students) | Βάρος ακμής |
| Χρονοθυρίδα (timeslot) | Χρώμα |

Όταν ένα μάθημα έχει τουλάχιστον έναν κοινό σπουδαστή με ένα άλλο μάθημα, τότε υπάρχει ακμή μεταξύ των δύο αυτών κόμβων. Το βάρος που έχει αυτή η ακμή είναι και το πλήθος των σπουδαστών που έχουν κοινούς. Οπότε στη γειτονιά κάθε κόμβου, δηλαδή όλοι οι κόμβοι που υπάρχουν ακμές από έναν κόμβο, βρίσκονται όλα τα μαθήματα που έχουν κοινούς σπουδαστές και το χρώμα που θα ανατεθεί στον κόμβο είναι η χρονοθυρίδα (ημέρα) που θα εξεταστεί το μάθημα. Ένα παράδειγμα αντιστοίχισης φαίνεται στην Εικόνα 2, όπου StudentsXY είναι ο αριθμός των κοινών σπουδαστών μεταξύ του μαθήματος X και του μαθήματος Y.

*Εικόνα 2. Παράδειγμα αντιστοίχισης*



## 3.2 Συμμετρικός πίνακας σύγκρουσης δυαδικών ψηφίων

Ο συμμετρικός πίνακας σύγκρουσης δυαδικών ψηφίων είναι μια άλλη μορφή που αντιπροσωπεύει το πρόβλημα του χρονοδιαγράμματος των εξετάσεων. Είναι ένας δυαδικός πίνακας N x N, το οποίο N δηλώνει τον αριθμό των εξετάσεων, που περιέχει 1 στο κελί i, j, όπου i, j ∈ {1, 2… N}, σε περίπτωση σύγκρουσης μεταξύ των δύο εξετάσεων i και j, και 0, διαφορετικά (βλέπε Πίνακα 2). Οι αναπαραστάσεις (τόσο ο πίνακας συγκρούσεων όσο και ένα γράφημα) μπορούν εύκολα να μετασχηματιστούν μεταξύ τους. Αυτός ο πίνακας σύγκρουσης μπορεί να αποδειχθεί χρήσιμος όταν οι αλγόριθμοι δημιουργούνται για τα προβλήματα χρονοδιαγράμματος.

*Πίνακας 2. Πίνακας Σύγκρουσης*

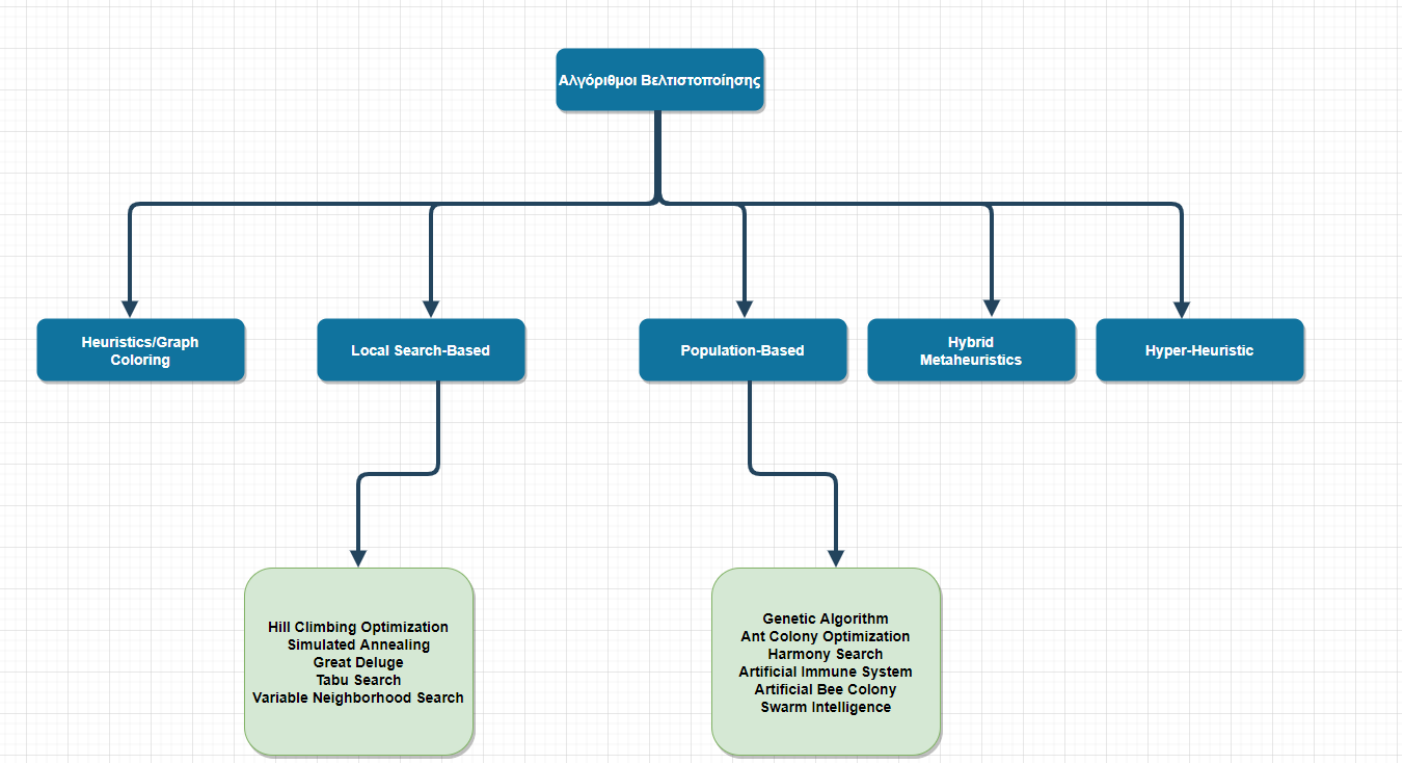
| **I/J** | **1** | **2** | **3** | **4** | **5** | **…** | **N** |
|---|---|---|---|---|---|---|---|
| **1** | - | 1 | 1 | 0 | 0 | | 0 |
| **2** | 1 | - | 0 | 1 | 0 | | 1 |
| **3** | 0 | 1 | - | 0 | 1 | | 0 |
| **4** | 1 | 0 | 1 | - | 0 | | 0 |
| **5** | 0 | 1 | 0 | 1 | - | | 1 |
| **…** | | | | | | - | |
| **N** | 0 | 1 | 1 | 0 | 0 | | - |

Σύμφωνα με τα παραπάνω, το πρόβλημα του χρονοδιαγράμματος των εξετάσεων μπορεί να θεωρηθεί ως ένα πρόβλημα χρωματισμού γραφήματος, στο οποίο ο στόχος είναι να βρεθούν οι ελάχιστες χρονικές θυρίδες που μπορούν να περιέχουν όλες τις εξετάσεις χωρίς συγκρούσεις.

# 4. Τεχνικές που εφαρμόζονται σε προβλήματα χρονοδιαγράμματος εξετάσεων χωρίς χωρητικότητα

Το πρόβλημα αναφέρεται στην εκχώρηση εξετάσεων σε χρονικά διαστήματα με τέτοιο τρόπο ώστε να είναι χωρίς σύγκρουση όσον αφορά τις εξετάσεις. Με τον όρο καμία σύγκρουση εννοείται η κατάσταση στην οποία κανένας μαθητής δεν χρειάζεται να εξεταστεί ταυτόχρονα σε δύο ή περισσότερα μαθήματα. Αυτή η ενότητα περιγράφει τις μεθόδους που χρησιμοποιούνται στη βιβλιογραφία για την αντιμετώπιση αυτού του προβλήματος. Εξετάζονται πρόσφατες τεχνικές που χρησιμοποιήθηκαν για την επίλυση του προβλήματος χρονοδιαγράμματος πανεπιστημιακών εξετάσεων που εφαρμόστηκαν από τους Carter, Laporte και Lee (1996), σύνολο δεδομένων που αποτελείται από 13 διαφορετικά προβλήματα. Προηγούμενες μελέτες αφιερωμένες στη διερεύνηση και ανάπτυξη διαφόρων αλγορίθμων βελτιστοποίησης για προβλήματα χρονοδιαγράμματος εξέτασης, έχουν πραγματοποιηθεί τις τελευταίες δεκαετίες και εφαρμόστηκαν στο παραπάνω σύνολο δεδομένων. Οι αλγόριθμοι βελτιστοποίησης που αναφέρονται σε αυτές τις μελέτες φαίνονται στην παρακάτω εικόνα (Εικόνα 3). Οι λεπτομέρειες αυτών των τεχνικών συζητούνται διεξοδικά στις επόμενες ενότητες.

*Εικόνα 3. Αλγόριθμοι Βελτιστοποίησης*




## 4.1 Heuristics/Graph Coloring Techniques

Στις μεθόδους αυτές οι εξετάσεις ταξινομούνται με βάση τη δυσκολία με την οποία θα χρωματιστούν από τις βασικές μεθόδους χρωματισμού γραφημάτων. Με τον χρωματισμό του γραφήματος, οι εξετάσεις τοποθετούνται η μια μετά την άλλη στις αντίστοιχες χρονοθυρίδες τους. Οι περισσότερες από τις ευρέως χρησιμοποιούμενες στρατηγικές βασίζονται στις στρατηγικές ταξινόμησης χρωματισμού Heuristic/Graph όπως παρουσιάζεται στον Πίνακα 4. Μία άλλη μέθοδος, που ονομάζεται μέθοδος τυχαίας ταξινόμησης χρησιμοποιήθηκε για να εισαγάγει τυχαιότητα στις υβριδικές προσεγγίσεις για χρωματισμό γραφημάτων, προκειμένου να γίνουν συγκρίσεις των αλγορίθμων, δηλαδή πως μεταβάλλονται οι λύσεις σε διαφορετικές ταξινομήσεις.

*Πίνακας 3. Μερικές στρατηγικές που βασίζονται σε χρωματισμό γραφήματος*

| **Heuristics** | **Στρατηγική ταξινόμησης** |
|---|---|
| Μεγαλύτερος βαθμός<br>(Broder, 1964) | Μείωση θέσης* σύμφωνα με τις συγκρούσεις που έχουν εξετάσεις με τις υπόλοιπες. |
| Μεγαλύτερη εγγραφή<br>(Wood, 1968) | Μείωση θέσης* σύμφωνα με τις εγγραφές για εξετάσεις. |
| Λιγότερος βαθμός κορεσμού<br>(Brélaz, 1979) | Αύξηση θέσης* σύμφωνα με τις διαθέσιμες χρονικές θέσεις για κάθε εξέταση στο χρονοδιάγραμμα κάθε φορά. |
| Μεγαλύτερος σταθμισμένος βαθμός<br>(Carter, 1996; Arani και Lotfi, 1989) | Παρόμοιο με το μεγαλύτερος βαθμό αλλά σταθμισμένο από πολλούς εμπλεκόμενους μαθητές. |
| Τυχαία ταξινόμηση<br>(Carter, 1994) | Ταξινόμηση εξετάσεων τυχαία. |
| Μεγαλύτερος χρωματισμένος βαθμός<br>(Newall, 1999) | Μείωση θέσης* σύμφωνα με τις συγκρούσεις που έχουν προγραμματισμένες εξετάσεις τη συγκεκριμένη στιγμή. |

**Η θέση ταξινόμησης προσδιορίζει τη σειρά με την οποία θα ταξινομηθούν, όσο μικρότερη η θέση τόσο πιο γρήγορα θα ταξινομηθεί.*

Το μεγάλο πλεονέκτημα που μπορούν να παρέχουν οι παραπάνω μέθοδοι είναι ότι είναι σε θέση να παρέχουν καλά λογικά αποτελέσματα μέσα σε λίγο χρόνο υπολογισμού και οι τεχνικές είναι πολύ πιο εύκολο να εφαρμοστούν. Οι αρχικές λύσεις κατασκευάζονται συχνά μέσω αυτών των γραφημάτων και ένα μεγάλο τμήμα λύσης μπορεί να κατασκευαστεί πριν από την εφαρμογή των τεχνικών βελτίωσης.

Οι παραπάνω τεχνικές εφαρμόστηκαν σε πραγματικά προβλήματα, για τη δημιουργία εξετάσεων του Πανεπιστήμιου Multimedia University and Universiti Malaysia Pahan της Μαλαισίας από τους Hussin et al., 2011 και Kahar and Kendall, 2010, και με βάση αυτά τα πειράματα από τις μελέτες, είναι κατάλληλες για το πρόβλημα που επικεντρώνεται στον σκληρό περιορισμό, δηλαδή να μην υπάρχει φοιτητής που να δίνει πάνω από μια εξέταση κάποια συγκεκριμένη στιγμή, αλλά είναι δύσκολο να επιλυθούν οι μαλακοί περιορισμοί.

## 4.2 Local Search-based Techniques

Τοπικές τεχνικές βασισμένες στην αναζήτηση είναι μια επαναληπτική διαδικασία που συχνά ονομάζεται τεχνική ενός σημείου ή τεχνική μιας λύσης. Οι τεχνικές αυτές μετακινούνται από μια λύση σε μια άλλη, εφαρμόζοντας τοπικές αλλαγές, μέχρι να βρεθεί μια λύση η οποία θεωρείται βέλτιστη είτε να ξεπεραστεί κάποιο χρονικό όριο. Για την αντιμετώπιση πολλών προβλημάτων χρονοδιαγράμματος, χρησιμοποιήθηκαν διάφορες τοπικές τεχνικές βασισμένες στην αναζήτηση, για παράδειγμα, Great Deluge, Variable Neighborhood Search, Simulated Annealing, Hill Climbing και Tabu Search.

### 4.2.1 Hill Climbing Optimization (HCO)

Η βελτιστοποίηση Hill Climbing εξερευνά ένα πολύ περιορισμένο τμήμα του χώρου αναζήτησης όπου δέχεται μόνο μια βελτιωμένη κίνηση. Αυτός ο αλγόριθμος ξεκινά με τη βοήθεια μιας αρχικής λύσης. Ο αλγόριθμος εφαρμόζει μια κίνηση για τη δημιουργία μιας νέας λύσης. Σε περίπτωση που υπάρχει βελτίωση

στην ποιότητα της λύσης, τότε η νέα λύση αντικαθιστά την τρέχουσα (Burke and Newall, 2003).Το μειονέκτημα αυτής της διαδικασίας είναι ότι, μπορεί εύκολα να παγιδευτεί στα τοπικά βέλτιστα. Προκειμένου να ξεπεραστεί αυτό το μειονέκτημα, πολλοί ερευνητές έχουν τροποποιήσει και υβριδοποιήσει τον αλγόριθμο του Hill Climbing με άλλες προσεγγίσεις.

Οι Merlot, L. T., N. Boland, B. D. Hughes & P. J. Stuckey, το 2003, εφάρμοσαν μια διαδικασία τριών φάσεων όπου στην πρώτη φάση, χρησιμοποιείται μια τεχνική περιορισμού προγραμματισμού στη δημιουργία αρχικών λύσεων. Στη δεύτερη φάση χρησιμοποιείται η τεχνική Simulated Annealing με χρήση της δομής αλυσίδας Kempe η οποία μπορεί να οριστεί ως εξής. Έστω ένα γράφημα G και ένα σύνολο ακμών Ε. Έχοντας μια διαδικασία χρωματισμού του γραφήματος αποκτούμε ένα σύνολο χρωμάτων C. Μια αλυσίδα Kempe είναι ένα σύνολο συνδεδεμένων κόμβων που έχουν χρωματιστεί με δύο μόνο χρώματα, έστω κόκκινο και μπλε. Τότε μπορούμε να αντιστρέψουμε αυτά τα δύο χρώματα και να μας δώσει τη δυνατότητα να χρωματίσουμε έναν άλλον κόμβο με χρώμα το οποίο δε μπορούσαμε να χρησιμοποιήσουμε προηγουμένως. Αργότερα, η προσέγγιση HCO εφαρμόζεται στην τρίτη φάση προκειμένου να βελτιωθεί περαιτέρω η ποιότητα της λύσης που λαμβάνεται από τη δεύτερη φάση. Αυτή η προσέγγιση είναι σε θέση να παράγει μερικά από τα ανταγωνιστικά αποτελέσματα για το σύνολο δεδομένων του Carter.

Μια άλλη τροποποίηση αυτής του αλγόριθμου Hill Climbing είναι η εκδοχή Late Acceptance Hill Climbing η οποία βελτιώνει επαναληπτικά μια συγκεκριμένη λύση. Οι τιμές (το κόστος της λύσης) αποθηκεύεται και διατηρούνται από τον αλγόριθμο σε μια λίστα συγκεκριμένου μεγέθους. Με βάση τη σύγκριση θα γίνει αποδεκτή μια νέα λύση μεταξύ προηγούμενων και νέων λύσεων στη λίστα. Τα αποτελέσματα που προκύπτουν από αυτήν την προσέγγιση είναι ανταγωνιστικά σε κόστος σε σχέση με τις υπόλοιπες προσεγγίσεις. Αυτή η προσέγγιση έχει μειονεκτήματα στα οποία η απόδοση σχετίζεται κυρίως με το μέγεθος της λίστας, διαφορετικά σύνολα δεδομένων έχουν διαφορετικές ρυθμίσεις μεγέθους. Επομένως το αποτέλεσμα της προσέγγισης αυτής εξαρτάται πλήρως από την επιλογή κατάλληλου μεγέθους λίστας.

### 4.2.2 Simulated Annealing (SA)

Ο αλγόριθμος Simulated Annealing (SA) αξιοποιεί την αναλογία της φυσικής διαδικασίας της θέρμανσης ενός μετάλλου σε υψηλή θερμοκρασία και την άμεση ψύξη του σε μια κρυσταλλική δομή ώστε να μην υπάρχει δυνατότητα να γίνουν άλλες αλλαγές έπειτα. Η SA έχει τη δυνατότητα να αποφεύγει να γίνει παγίδευση σε τοπικά βέλτιστα. Γίνεται μια τυχαία αναζήτηση καλύτερης λύσης που δέχεται όχι μόνο τις θετικές αλλά και τις αρνητικές αλλαγές αλλά με συγκεκριμένο εύρος πιθανότητας.

Η επεξεργασία της SA ξεκινά από μια αρχική λύση που δημιουργείται τυχαία. Η γειτονική ή υποψήφια λύση που παράγεται επίσης τυχαία θα αναγνωρίζεται πάντα εάν είναι καλή σε σύγκριση με την τρέχουσα, υπάρχει όμως συγκεκριμένη πιθανότητα να γίνεται αποδεκτή μια χειρότερη γειτονική λύση που καθορίζεται από την πιθανότητα Boltzmann $P = e^{-\frac{a}{t}}$, όπου α είναι η διαφορά της ποιότητας της τρέχοντος και της υποψήφιας λύσης και το t είναι μια παράμετρος ελέγχου που είναι γνωστή ως θερμοκρασία. Η θερμοκρασία μειώνεται με την πάροδο του χρόνου σύμφωνα με ένα πρόγραμμα ψύξης.

Όπως και στο HCO έτσι και ο αλγόριθμος Simulated Annealing εντάχθηκε σε διάφορες υβριδικές προσεγγίσεις. Μια από αυτές είναι η δημιουργία μιας αρχικής λύσης με τον αλγόριθμο SA χρησιμοποιώντας προγραμματισμό με περιορισμούς, δηλαδή τους περιορισμούς για τις εφικτές λύσεις του προβλήματος χρονοπρογραμματισμού. Ακολουθεί η χρήση της μεθόδου Hill Climbing για βελτίωση της λύσης με συνδυασμό των αλυσίδων Kempe. Η προσέγγιση αυτή κατέχει μερικά από τα καλύτερα αποτελέσματα αυτής της στιγμής. Βέβαια, ένα μειονέκτημα που έχει αυτή η μέθοδος είναι πως η διαδικασία ψύξης απαιτεί χρόνο, οπότε όταν παρέχεται περιορισμένος χρόνος, καθίσταται κρίσιμο και δύσκολο να

συντονισθούν οι μεταβλητές του αλγορίθμου SA σύμφωνα με τις οποίες πρέπει να επιλυθούν τα συγκεκριμένα προβλήματα.

Για να ξεπεραστεί το πρόβλημα χρόνου δημιουργήθηκε ο αλγόριθμος Fast Simulated Annealing (FastSA) από τους Leite N. Et αϊ. (2019). Η παραλλαγή αυτή χρησιμοποιεί δέκα δοχεία (θέσεις) θερμοκρασιών και για κάθε επιλεγμένη εξέταση εγκρίνεται η μετατόπιση της μόνο εάν η δηλωμένη εξέταση είχε εγκεκριμένη αλλαγή στο αμέσως προηγούμενο δοχείο θερμοκρασίας. Στα δέκα δοχεία θερμοκρασίας που δημιουργήθηκαν, ο FastSA επιβεβαίωσε ότι ένας ίσος αριθμός αξιολογήσεων πραγματοποιείται σε κάθε κάδο. Η προτεινόμενη μέθοδος παρατηρήθηκε ότι η εξέταση πιθανότατα θα έχει λίγη ή σχεδόν μηδενική αποδεκτή κίνηση στο μέλλον εάν λάβει μη αποδεκτή κίνηση στο προηγούμενο δοχείο θερμοκρασίας. Ο FastSA χρησιμοποιεί έως και 41% λιγότερες αξιολογήσεις σε συγκεκριμένο σύνολο δεδομένων και 17% λιγότερες αξιολογήσεις γενικά. Με βάση το κόστος λύσης, η προτεινόμενη μέθοδος είναι ανταγωνιστική με τον κλασσικό αλγόριθμο Simulated Annealing.

### 4.2.3 Great Deluge Algorithm (GDA)

Ο Great Deluge αλγόριθμος είναι μια ενναλακτική μορφή του αλγορίθμου Simulated Annealing η οποία είναι λιγότερο εξαρτώμενη από τις παραμέτρους που χρειάζεται. Απαιτεί μόνο δύο παραμέτρους, τον μέγιστο υπολογιστικό χρόνο και την εκτίμηση της ποιότητας της λύσης. Το όνομα προέρχεται από την αναλογία ότι σε έναν μεγάλο κατακλυσμό ένα άτομο που ανεβαίνει σε έναν λόφο θα προσπαθήσει να κινηθεί προς οποιαδήποτε κατεύθυνση που δεν βρέχει τα πόδια του με την ελπίδα να βρει έναν τρόπο να ανεβαίνει καθώς η στάθμη του νερού αυξάνεται. Ο αλγόριθμος ξεκινά με μια κακή προσέγγιση, S, της βέλτιστης λύσης. Έπειτα υπολογίζεται μια τιμή b σύμφωνα με την S που δηλώνει πόσο ανεπιθύμητη είναι αυτή η αρχική προσέγγιση, όσο υψηλότερη είναι αυτή η τιμή τόσο πιο ανεπιθύμητη είναι η λύση. Άλλη μια τιμή που υπολογίζεται με βάση έναν αριθμό παραγόντων αλλά και την προηγούμενη τιμή b, είναι η ανοχή. Μια νέα κατά προσέγγιση λύσης S ', που ονομάζεται γείτονας του S, υπολογίζεται με βάση το S. Η τιμή b' για το S', υπολογίζεται και συγκρίνεται με την ανοχή. Εάν το b' είναι καλύτερο από την ανοχή, τότε ο αλγόριθμος επανεκκινείται αναδρομικά με S: = S' και ανοχή:D(ανοχή) όπου D μια συνάρτηση για ρυθμό αποσύνθεσης (Decay rate) που μειώνει την ανοχή (που αντιπροσωπεύει αύξηση της στάθμης του νερού). Εάν το b 'είναι χειρότερο από την ανοχή, επιλέγεται ένας διαφορετικός γείτονας S * του S και η διαδικασία επαναλαμβάνεται. Εάν όλοι οι γείτονες του S παράγουν κατά προσέγγιση λύσεις πέρα από την ανοχή, τότε ο αλγόριθμος τερματίζεται και το S προβάλλεται ως η καλύτερη προσέγγιση που έχει ληφθεί.

Σε αυτόν τον αλγόριθμο έχουν εφαρμοστεί διάφορες τροποποιήσεις για το πρόβλημα μας και πολλές από αυτές έχουν καλύτερα αποτελέσματα από τους Hill Climbing και Simulated Annealing. Μερικές από αυτές:

- Εφαρμογή του GDA με προκαθορισμένο χρόνο. Ο ρυθμός αποσύνθεσης ορίζεται ως διαφορά μεταξύ της τρέχουσας ποιότητας ης λύσης και της επιθυμητής ποιότητας της λύσης με την πάροδο του χρόνου.
- Εφαρμογή του GDA με χρήση ενός ασαφούς συνόλου.
- Σε υβριδικές υλοποιήσεις γίνεται χρήση δυναμικού ρυθμού αποσύνθεσης (Dynamic Decay Rate), όπου σε κάθε πέρασμα των δεδομένων από διάφορες μεθόδους αλλάζει ο ρυθμός αποσύνθεσης αλλά και η τιμή της ποιότητας της λύσης που έχουμε σαν στόχο.

### 4.2.4 Tabu Search (TS)

Βασικά, η ιδέα αυτού του αλγορίθμου αναζήτησης tabu είναι να μελετήσει τον χώρο αναζήτησης διατηρώντας τις τελευταίες λύσεις που επισκέφτηκε σε μια λίστα tabu για να αποφύγει την επανάληψη της αναζήτησης σε αυτές τις λύσεις ενώ παράγονται νέες λύσεις. Αυτή η προσέγγιση στοχεύει να ξεφύγει από τα τοπικά βέλτιστα και να αναγκάσει τον αλγόριθμο να εξετάσει νέες περιοχές του χώρου. Το μειονέκτημα αυτής της προσέγγισης είναι ότι οι παράμετροι (δηλαδή, κριτήρια διακοπής και λίστα ταμπού) πρέπει να συντονιστούν από τον χρήστη και αυτό σχετίζεται με κάθε πρόβλημα ξεχωριστά. Ένας τρόπος με τον οποίο ξεπεράστηκε αυτό το μειονέκτημα είναι η χρήση δυναμικού μεγέθους λίστας tabu που ρυθμίζεται προσαρμοστικά κατά τη διάρκεια της αναζήτησης. Άλλη μια εκδοχή είναι η χρήση δύο λιστών tabu, μια στην οποία διατηρούνται οι τελευταίες λύσεις (Short Term Tabu List) αλλά και μια στην οποία καταχωρούνται λύσεις παλαιότερες λύσεις που δε θέλουμε να επισκεφτούμε (Long Term Tabu List).

### 4.2.5 Variable Neighborhood Search (VNS)

Ο αλγόριθμος αυτός εφαρμόζει έναν αριθμό δομών γειτονιάς, προκειμένου να ξεφύγει από τα τοπικά βέλτιστα, αλλάζοντας συστηματικά από τη μία γειτονιά στην άλλη με βάση την κατάσταση αναζήτησης. Το VNS είναι ένας meta-heuristic αλγόριθμος που εκτελεί συστηματικά τη διαδικασία αλλαγής γειτονιάς, τόσο κατά την κατάβαση στα τοπικά ελάχιστα όσο και στην απόδραση από τις κοιλάδες που τις περιέχουν. Το VNS βασίζεται στις ακόλουθες αντιλήψεις:

1. Ένα τοπικό ελάχιστο σε σχέση με μια δομή γειτονιάς δεν είναι απαραίτητα ένα τοπικό ελάχιστο για μια άλλη δομή γειτονιάς.
2. Ένα καθολικό ελάχιστο είναι ένα τοπικό ελάχιστο σε σχέση με όλες τις πιθανές δομές γειτονιάς.
3. Για πολλά προβλήματα, τα τοπικά ελάχιστα σε σχέση με μία ή περισσότερες γειτονιές είναι σχετικά κοντά μεταξύ τους.

Σε αντίθεση με πολλούς άλλους αλγόριθμους, τα βασικά σχήματα του VNS και των επεκτάσεών του είναι απλά και απαιτούν λίγες και μερικές φορές καθόλου παραμέτρους. Μια παραλλαγή του αλγορίθμου VNS χρησιμοποιώντας τον Γενετικό Αλγόριθμο με πρότυπα, τα αποτελέσματα βελτιώθηκαν περαιτέρω επιλέγοντας ένα υποσύνολο γειτονιών έξυπνα.

## 4.3 Population-Based Techniques

Αυτές οι τεχνικές θεωρούνται ως στοχαστικοί αλγόριθμοι που αρχικοποιούνται μαζί με έναν πληθυσμό λύσεων σε μια συγκεκριμένη στιγμή. Εφαρμόζεται μια στρατηγική, επαναληπτικά για να υπολογιστεί η βέλτιστη λύση από τον πληθυσμό λύσεων που διαθέτει μέχρι εκείνη την στιγμή και έπειτα δημιουργούνται νέες λύσεις που αποτελούν τον νέο πληθυσμό λύσεων. Ο γενετικός αλγόριθμος είναι η πιο προσαρμοσμένη τεχνική βάσει πληθυσμού. Άλλες τεχνικές είναι η βελτιστοποίηση Ant Colony, η Harmony Search, Artificial Immune System , Artificial Bee Colony.

### 4.3.1 Genetic Algorithm (GA)

Ο Genetic Algorithm (GA) είναι εμπνευσμένος από τη διαδικασία της φυσικής επιλογής που ανήκει στην ευρύτερη κατηγορία εξελικτικών αλγορίθμων (EA). Χρησιμοποιείται για τη δημιουργία λύσεων για προβλήματα βελτιστοποίησης και αναζήτησης που χαρακτηρίζονται ως υψηλής ποιότητας. Βασιζόμενοι

σε χειριστές όπως η μετάλλαξη, η διασταύρωση και η φυσική επιλογή εμπνευσμένους από την βιολογία. Σε έναν γενετικό αλγόριθμο (GA), ένας πληθυσμός υποψήφιων λύσεων (που ονομάζονται έιτε άτομα είτε πλάσματα) σε ένα πρόβλημα βελτιστοποίησης εξελίσσεται συνεχώς προς λύσεις υψηλότερης ποιότητας, δηλαδή καλύτερες λύσεις. Κάθε υποψήφια λύση έχει ένα σύνολο ιδιοτήτων (χρωμοσώματα) που μπορούν να μεταλλαχθούν και να τροποποιηθούν. Η εξέλιξη ξεκινά συνήθως από έναν πληθυσμό πλασμάτων (ή ατόμων) που δημιουργούνται μέσω μιας τυχαίας γεννήτριας και είναι μια επαναληπτική διαδικασία, με τον πληθυσμό σε κάθε επανάληψη να ονομάζεται γενιά. Σε κάθε γενιά αξιολογείται η καταλληλότητα κάθε ατόμου στον πληθυσμό, η οποία είναι συνήθως η αξία της αντικειμενικής συνάρτησης στο πρόβλημα βελτιστοποίησης που επιλύεται. Τα πιο κατάλληλα άτομα επιλέγονται στοχαστικά από τον τρέχοντα πληθυσμό και το γονιδίωμα κάθε ατόμου τροποποιείται (ανασυνδυάζεται και πιθανώς τυχαία μεταλλάσσεται) για να σχηματίσει μια νέα γενιά. Στη συνέχεια, η νέα γενιά υποψηφίων λύσεων χρησιμοποιείται στην επόμενη επανάληψη του αλγορίθμου. Συνήθως, ο αλγόριθμος τερματίζεται όταν είτε έχει παραχθεί ένας μέγιστος αριθμός γενεών, είτε έχει επιτευχθεί ικανοποιητικό επίπεδο καταλληλότητας για τον πληθυσμό.

Ένας τυπικός γενετικός αλγόριθμος απαιτεί:

1. μια γενετική αναπαράσταση της λύσης
2. μια συνάρτηση καταλληλότητας για την αξιολόγηση της λύσης

Η εφαρμογή του GA στο πρόβλημα μας κάνει αντιστοίχιση των ατόμων με τις αναθέσεις εξετάσεων σε χρονοθυρίδες και σαν συνάρτηση καταλληλότητας παίρνει το κόστος της τρέχουσας λύσης.

### 4.3.2 Ant Colony Optimization (ACO)

Αυτός ο αλγόριθμος μιμείται στην αποικία των μυρμηγκιών, την διαδικασία συλλογής τροφής τους. Ένα μονοπάτι που ενώνει την πηγή τροφής και τη φωλιά κατασκευάζονται από αυτήν την αποικία. Οι φερομόνες εναποτίθενται από τα μυρμήγκια στο μονοπάτι από τη φωλιά προς την πηγή τροφής. Στη συνέχεια, οι διαδρομές φερομονών κατευθύνονται προς τροφή ή φωλιά που χρησιμοποιείται από μυρμήγκια με τη βοήθεια κατατεθειμένων φερομονών κατά μήκος του μονοπατιού. Με την αύξηση της πυκνότητας της φερομόνης, τα μυρμήγκια αυξάνονται επίσης για να ακολουθήσουν το μονοπάτι. Με αυτόν τον τρόπο, τα μονοπάτια υψηλής πυκνότητας σε σύγκριση με άλλα είναι πιο ελκυστικά για τα μυρμήγκια. Όπως είναι φυσικό τα τεχνητά μυρμήγκια που χρησιμοποιούμε στον αλγόριθμο μας αντιπροσωπεύουν μεθόδους πολλαπλών παραγόντων εμπνευσμένες από τη συμπεριφορά των πραγματικών μυρμηγκιών. Οι συνδυασμοί τεχνητών μυρμηγκιών και τοπικών αλγορίθμων αναζήτησης έχουν γίνει μέθοδος επιλογής για πολλές εργασίες βελτιστοποίησης που περιλαμβάνουν κάποιο είδος γραφήματος. Τα τεχνητά μυρμήγκια εντοπίζουν βέλτιστες λύσεις μετακινώντας ένα χώρο παραμέτρων που αντιπροσωπεύει όλες τις πιθανές λύσεις. Τα πραγματικά μυρμήγκια καθορίζουν φερομόνες που κατευθύνουν ο ένας τον άλλον σε πόρους ενώ εξερευνούν το περιβάλλον τους. Τα προσομοιωμένα μυρμήγκια καταγράφουν επίσης τις θέσεις τους και την ποιότητα των λύσεων τους, έτσι ώστε σε μεταγενέστερες προσομοιώσεις που επαναλαμβάνονται, περισσότερα μυρμήγκια να εντοπίζουν καλύτερες λύσεις. Στη φύση, με την πάροδο του χρόνου το ίχνος φερομόνης αρχίζει να εξατμίζεται, μειώνοντας έτσι την ελκυστική του αντοχή. Όσο περισσότερο χρόνο χρειάζεται για ένα μυρμήγκι να ταξιδέψει κάτω από το μονοπάτι και να επιστρέψει ξανά, τόσο περισσότερο χρόνο πρέπει να εξατμιστούν οι φερομόνες. Ένα μικρό μονοπάτι, συγκριτικά, βαδίζει πιο συχνά, και έτσι η πυκνότητα φερομόνης αυξάνεται σε μικρότερα μονοπάτια από τα μεγαλύτερα. Στο πρόβλημα μας το ίχνος φερομόνης εξαρτάται άμεσα από τη συνάρτηση του κόστους με την οποία θα θέσουμε. Δηλαδή όσο μικρότερο είναι το κόστος σε μια λύση τόσο πιο πιθανό είναι να την εντοπίσουν τα τεχνητά μας μυρμήγκια.

Με την εξάτμιση της φερομόνης υπάρχει επίσης το πλεονέκτημα της αποφυγής της σύγκλισης σε τοπικά βέλτιστες λύσεις.

Στους αλγόριθμους ACP, ένα τεχνητό μυρμήγκι είναι ένας απλός υπολογιστικός παράγοντας που αναζητά καλές λύσεις σε ένα δεδομένο πρόβλημα βελτιστοποίησης. Για να εφαρμοστεί ο ACO σε πρόβλημα βελτιστοποίησης πρέπει να μετατραπεί σε πρόβλημα εύρεσης της συντομότερης διαδρομής σε ένα σταθμισμένο γράφημα. Στο πρώτο βήμα κάθε επανάληψης, κάθε μυρμήγκι στοχαστικά κατασκευάζει μια λύση, δηλαδή τη σειρά με την οποία πρέπει να ακολουθούνται τα άκρα στο γράφημα. Στο δεύτερο βήμα, συγκρίνονται οι διαδρομές που βρέθηκαν από τα διάφορα μυρμήγκια. Το τελευταίο βήμα συνίσταται στην ενημέρωση των επιπέδων φερομόνης σε κάθε άκρη.

Η διαδικασία ACO είναι:

Όσο όχι_ τερματισμός επανέλαβε:

Δημιουργία λύσεων

Σύγκριση λύσεων

Ενημέρωση φερομόνης

Τέλος_επανάληψης

## 4.3.3 Artificial Immune System (AIS)

Οι αλγόριθμοι AIS είναι μια κατηγορία υπολογιστικών ευφυών συστημάτων μηχανικής μάθησης βασισμένων σε κανόνες εμπνευσμένους από τις αρχές και τις διαδικασίες του ανοσοποιητικού συστήματος των σπονδυλωτών. Οι αλγόριθμοι συνήθως διαμορφώνονται σύμφωνα με τα χαρακτηριστικά της μάθησης και της μνήμης του ανοσοποιητικού συστήματος για χρήση στην επίλυση προβλημάτων. Τρείς από τους αλγόριθμους που εφαρμόστηκαν σε προβλήματα χρονοδιαγράμματος είναι:

- Clonal Selection Algorithm: Εφαρμόζεται συχνότερα σε τομείς βελτιστοποίησης και αναγνώρισης προτύπων.
- Negative Selection Algorithm: Χρησιμοποιείται συνήθως για προβλήματα που αφορούν την ταξινόμηση και την αναγνώριση μοτίβων.
- Immune Network Algorithms: Έχουν χρησιμοποιηθεί σε ομαδοποίηση, οπτικοποίηση δεδομένων, τομείς βελτιστοποίησης και κοινή χρήση ιδιοτήτων με τεχνητά νευρικά δίκτυα.

Τα πειραματικά αποτελέσματα, χρησιμοποιώντας σύνολα δεδομένων από το Carter αποκάλυψαν ότι όλοι οι αλγόριθμοι έχουν παράγει επιτυχώς καλά χρονοδιαγράμματα εξέτασης. Οι Clonal Selection και Negative Selection είναι πιο αποτελεσματικοί, ωστόσο ο Immune Network τρέχει γρηγορότερα.

## 4.3.4 Artificial Bee Colony (ABC)

Ο αλγόριθμος ABC αναπτύχθηκε ως καθολικός αλγόριθμος βελτιστοποίησης στον οποίο προσομοιώνεται η συμπεριφορά συλλογής τροφής των μελισσών. Στο μοντέλο ABC, η αποικία αποτελείται από τρεις ομάδες μελισσών: απασχολούμενες μέλισσες, θεατές και ανιχνευτές. Υποτίθεται ότι υπάρχει μόνο μία τεχνητή μέλισσα για κάθε πηγή τροφής. Με άλλα λόγια, ο αριθμός των απασχολούμενων μελισσών στην αποικία είναι ίσος με τον αριθμό των πηγών τροφίμων γύρω από την κυψέλη. Οι απασχολούμενες μέλισσες

πηγαίνουν στην πηγή τροφής τους και επιστρέφουν στην κυψέλη και χορεύουν σε αυτήν την περιοχή. Η απασχολούμενη μέλισσα της οποίας η πηγή τροφής έχει εγκαταλειφθεί γίνεται ανιχνευτής και αρχίζει να ψάχνει να βρει μια νέα πηγή τροφής. Οι θεατές παρακολουθούν τους χορούς των μελισσών και επιλέγουν πηγές τροφίμων ανάλογα με τους χορούς. Τα κύρια βήματα του αλγορίθμου δίνονται παρακάτω:

Οι αρχικές πηγές τροφίμων παράγονται για όλες τις απασχολούμενες μέλισσες.
Όσο δεν πληρούνται οι απαιτήσεις επανέλαβε:
Κάθε εργαζόμενη μέλισσα πηγαίνει σε μια πηγή τροφής στη μνήμη της και καθορίζει μια πλησιέστερη πηγή, στη συνέχεια αξιολογεί την ποσότητα του νέκταρ της και χορεύει στην κυψέλη.
Κάθε θεατής παρακολουθεί το χορό των μελισσών και επιλέγει μία από τις πηγές τους ανάλογα με τους χορούς και στη συνέχεια πηγαίνει σε αυτήν την πηγή. Αφού επέλεξε έναν γείτονα, αξιολογεί την ποσότητα του νέκταρ.
Οι εγκαταλελειμμένες πηγές τροφίμων προσδιορίζονται και αντικαθίστανται από τις νέες πηγές τροφίμων που ανακαλύπτονται από τους ανιχνευτές.
Η καλύτερη πηγή τροφίμων που έχει βρεθεί μέχρι στιγμής καταχωρείται.
Τέλος_Επανάληψης

Η θέση μιας πηγής τροφίμων αντιπροσωπεύει μια πιθανή λύση στο πρόβλημα βελτιστοποίησης και η ποσότητα νέκταρ μιας πηγής τροφίμων αντιστοιχεί στην ποιότητα (καταλληλότητα) της σχετικής λύσης.

## 4.4 Hybrid Metaheuristics Techniques

Ο συνδυασμός στοιχείων διαφορετικών μεθόδων σε μία είναι η κύρια ιδέα των υβριδικών μεθόδων. Διάφοροι συνδυασμοί μεθόδων από αυτές που αναφέρθηκαν ήδη αλλά και από μεθόδους και διαδικασίες που δεν είναι άμεσα συσχετιζόμενες με το πρόβλημα χρονοδιαγράμματος έχουν γίνει και έχουν παράγει μερικές από τις καλύτερες λύσεις. Κάποιες από τις υβριδικές τεχνικές αναφέρθηκαν ήδη στις μεθόδους που συμπεριλαμβάνονται σε αυτές.

Μερικές υβριδικές μέθοδοι συνδυάζουν:

- Great Deluge Algorithm και Electromagnetism-like Algorithm
- Hill Climbing και χειριστής μετάλλαξης του Genetic Algorithm
- Tabu Search και Memetic algorithm (επέκταση του GA)
- Artificial Bee Colony και Simulated Annealing
- Genetic Algorithm και Record-to-Record Travel(αλγόριθμος βελτιστοποίησης βασισμένος στην αναζήτηση).
- Artificial Bee Colony και Great Deluge.

## 4.5 Hyper-Heuristic Techniques

Η Hyper-Heuristic είναι μια ευρετική μέθοδος αναζήτησης που έχει ως στόχο να αυτοματοποιήσει, συχνά με την ενσωμάτωση μερικών τεχνικών μηχανικής μάθησης, τη διαδικασία επιλογής, συνδυασμού, δημιουργίας ή προσαρμογής πολλών απλούστερων ευρετικών (ή συστατικών τέτοιων ευρετικών) για την αποτελεσματική επίλυση προβλημάτων υπολογιστικής αναζήτησης. Ένα από τα κίνητρα για τη μελέτη υπερ-ευρετικών είναι η δημιουργία συστημάτων που μπορούν να χειριστούν τάξεις προβλημάτων και όχι να λύσουν μόνο ένα πρόβλημα. Μπορεί να υπάρχουν πολλές μέθοδοι από τις οποίες μπορεί κανείς να επιλέξει για την επίλυση ενός προβλήματος και κάθε μία έχει τη δική της δύναμη και αδυναμία. Η ιδέα

είναι αυτόματα ο αλγόριθμος να επιλέγει συνδυάζοντας τη δύναμη και αντισταθμίζοντας την αδυναμία αυτών των ευρετικών μεθόδων. Έτσι, όταν χρησιμοποιούμε Hyper-Heuristic τεχνική, προσπαθούμε να βρούμε τη σωστή μέθοδο ή συνδιασμό μεθόδων σε μια δεδομένη κατάσταση αντί να προσπαθούμε να λύσουμε ένα πρόβλημα άμεσα. Επιπλέον, αναζητούμε μια γενικά εφαρμόσιμη μεθοδολογία αντί να επιλύουμε ένα μόνο πρόβλημα.

# 5. Meta-Heuristic Optimization Algorithms

Οι meta-heuristic τεχνικές βελτιστοποίησης έχουν γίνει πολύ δημοφιλείς τις τελευταίες δύο δεκαετίες. Παραδόξως, μερικοί από αυτούς, είναι αρκετά γνωστοί μεταξύ όχι μόνο επιστημόνων πληροφορικής αλλά και επιστημόνων από διαφορετικά πεδία. Εκτός από τον τεράστιο αριθμό θεωρητικών εργασιών, τέτοιες τεχνικές βελτιστοποίησης έχουν εφαρμοστεί σε διάφορους τομείς σπουδών. Εδώ τίθεται ένα ερώτημα γιατί τα μετα-ευρετικά έχουν γίνει εξαιρετικά κοινά. Η απάντηση σε αυτό το ερώτημα μπορεί να συνοψιστεί σε τρεις κύριους λόγους: απλότητα, ευελιξία και αποφυγή τοπικών βέλτιστων.

Πρώτον, είναι αρκετά απλοί αλγόριθμοι. Έχουν εμπνευστεί κυρίως από πολύ απλές έννοιες. Οι εμπνεύσεις σχετίζονται συνήθως με φυσικά φαινόμενα, συμπεριφορές ζώων ή εξελικτικές έννοιες. Η απλότητα επιτρέπει στους επιστήμονες να προσομοιώνουν διαφορετικές φυσικές έννοιες, να προτείνουν νέους meta-heuristic αλγόριθμους, να υβριδοποιούν δύο ή περισσότερους ή να βελτιώνουν τους τρέχων. Επιπλέον, η απλότητα βοηθά άλλους επιστήμονες να μάθουν γρήγορα αυτές τις τεχνικές και να τις εφαρμόσουν στα προβλήματά τους.

Δεύτερον, η ευελιξία αναφέρεται στη δυνατότητα εφαρμογής τους σε διαφορετικά προβλήματα χωρίς ιδιαίτερες αλλαγές στη δομή του αλγορίθμου. Τα βήματα είναι εύκολα εφαρμόσιμα σε διαφορετικά προβλήματα, καθώς υποθέτουν ως επί το πλείστων προβλήματα ως μαύρα κουτιά. Με άλλα λόγια, μόνο οι είσοδοι και έξοδοι ενός συστήματος είναι σημαντικές για έναν meta-heuristic αλγόριθμο. Έτσι, το μόνο που χρειάζεται ένας σχεδιαστής είναι να ξέρει πώς να αναπαραστήσει το πρόβλημά του/της.

Τέλος, τα meta-heuristics έχουν ανώτερες ικανότητες να αποφεύγουν τα τοπικά βέλτιστα σε σύγκριση με τις συμβατικές τεχνικές βελτιστοποίησης. Αυτό οφείλεται στη στοχαστική φύση των μετα-ευρετικών που τους επιτρέπει να αποφύγουν τη στασιμότητα στις τοπικές λύσεις και να αναζητήσουν εκτενώς ολόκληρο τον χώρο αναζήτησης. Ο χώρος αναζήτησης πραγματικών προβλημάτων είναι συνήθως άγνωστος και πολύ περίπλοκος με τεράστιο αριθμό τοπικών βέλτιστων, επομένως οι αλγόριθμοι αυτοί είναι καλές επιλογές για τη βελτιστοποίηση αυτών των απαιτητικών πραγματικών προβλημάτων.

Όπως και οι heuristic έτσι και ένα σύνολο meta-heuristic αλγορίθμων βελτιστοποίησης είναι εμπνευσμένοι από τη φύση και είναι ένα σύνολο καινοτόμων μεθοδολογιών και προσεγγίσεων επίλυσης προβλημάτων που προέρχονται από φυσικές διαδικασίες. Αυτοί οι αλγόριθμοι είναι εξαιρετικά αποτελεσματικοί στην εύρεση βελτιστοποιημένων λύσεων σε πολυδιάστατα προβλήματα. Μερικοί από τους δημοφιλέστερους των τελευταίων ετών είναι:

- The Firefly Algorithm
- Bat algorithm
- Cuckoo Search
- Grey Wolf Optimizer
- Harris hawks optimization

## 5.1 The Firefly Algorithm

Ο αλγόριθμος αυτός αναπτύχθηκε από τον Xin-She Yang που εργαζόταν στο Cambridge το 2008. Προσάρμοσε τη συμπεριφορά των σμηνών πυγολαμπίδων για να αναπτύξει έναν αλγόριθμο για τη βελτιστοποίηση συναρτήσεων με πολλαπλά βέλτιστα. Συγκεκριμένα, χρησιμοποίησε την ιδέα του τρόπου με τον οποίο η φωτεινότητα των μεμονωμένων πυγολαμπίδων ελκύει τις υπόλοιπες πυγολαμπίδες και έναν παράγοντα τυχαιότητας για να ενθαρρύνει την εξερεύνηση του χώρου λύσης. Η δημοσίευση του αλγορίθμου της πυγολαμπίδας οδήγησε στη δημοσίευση πολλών εργασιών σχετικά με την ανάλυση και την τροποποίηση του αλγορίθμου και την εφαρμογή του σε πολλά προβλήματα του πραγματικού κόσμου και έχουν υπάρξει πολλές αναφορές για επιτυχημένες εφαρμογές του.

Τα βασικά χαρακτηριστικά αυτού του αλγορίθμου βασίζονται σε ένα συγκεκριμένο σύνολο κανόνων που αναπτύχθηκε από τον Xin-She Yang. Αυτοί οι κανόνες είναι οι εξής:

1. Οι πυγολαμπίδες δεν έχουν φύλο, έτσι ώστε μια πυγολαμπίδα να έλκεται από όλες τις άλλες πυγολαμπίδες.
2. Η ελκυστικότητα είναι ανάλογη με τη φωτεινότητά τους, και για οποιεσδήποτε δύο πυγολαμπίδες, η λιγότερο φωτεινή θα έλκεται από (και επομένως θα κινηθεί προς) την πιο φωτεινή. Ωστόσο, η φαινομενική φωτεινότητα μειώνεται όσο αυξάνεται η απόσταση τους μεταξύ τους.
3. Εάν δεν υπάρχουν πυγολαμπίδες πιο φωτεινές από μια δεδομένη πυγολαμπίδα, θα κινηθεί τυχαία.

Αυτοί οι τρεις κανόνες μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός αλγόριθμου βελτιστοποίησης με το πρόσθετο χαρακτηριστικό ότι η φωτεινότητα είναι ανάλογη με την τιμή της αντικειμενικής συνάρτησης. Οι κανόνες της πυγολαμπίδας μπορούν στη συνέχεια να μετατραπούν σε βήματα στον αλγόριθμο δημιουργώντας τις θέσεις ενός αρχικού πληθυσμού πυγολαμπίδων, υπολογίζοντας την τιμή της αντικειμενικής συνάρτησης για καθεμία από αυτές τις πυγολαμπίδες και στη συνέχεια εφαρμόζοντας αυτούς τους κανόνες για πολλές γενιές.

Ο ψευδοκώδικας για τον αλγόριθμο πυγολαμπίδας είναι ο εξής:

---

Αλγόριθμος πυγολαμπίδας

---

Για σταθερό αριθμό γενεών επανάλαβε

Για i = 1 : npop επανάλαβε

Για j = 1 : npop επανάλαβε

Αν φωτεινότητα πυγολαμπίδας i < φωτεινότητα πυγολαμπίδας j τότε

μετακινήστε το firefly i προς το j. τόσο καλύτερη τιμή χρησιμοποιώντας το (5.1)

Τέλος_αν

Ενημερώστε τη νέα λύση και ενημερώστε όλες τις τιμές έντασης φωτός χρησιμοποιώντας την αντικειμενική συνάρτηση.

Τέλος_επανάληψης

Τέλος_επανάληψης

Βαθμολογήστε τις πυγολαμπίδες ανάλογα με τη φυσική κατάσταση. δηλαδή την τιμή της αντικειμενικής συνάρτησης.

Αποθηκεύστε την τρέχουσα καλύτερη τιμή.

Τέλος_επανάληψης

Όπου npop είναι ο πληθυσμός των πυγολαμπίδων.

---

Η βιβλιογραφία αναφέρει ότι αυτός ο αλγόριθμος είχε σημαντική επιτυχία για μια σειρά τυπικών παγκόσμιων προβλημάτων δοκιμής βελτιστοποίησης. Ο αλγόριθμος έχει χρησιμοποιηθεί και μελετηθεί εκτενώς και έχουν προταθεί πολλές προτεινόμενες βελτιώσεις.

## 5.2 The Bat Algorithm

Ένας αλγόριθμός εμπνευσμένος από τον τρόπο που η νυχτερίδα καταλαβαίνει τον χώρο τριγύρω της. Οι νυχτερίδες μπορούν να δουν το ίδιο καλά όπως οι άνθρωποι, αλλά έχουν εξελίξει μια εξελιγμένη μέθοδο χρήσης του ήχου που τους επιτρέπει να πλοηγούνται και να βρίσκουν τροφή στο σκοτάδι, που ονομάζεται ηχοεντοπισμός. Οι νυχτερίδες παράγουν ηχοεντοπισμό εκπέμποντας παλμούς ήχου υψηλής συχνότητας μέσω του στόματος ή της μύτης τους και ακούγοντας την ηχώ. Με αυτήν την ηχώ, η νυχτερίδα μπορεί να καθορίσει το μέγεθος, το σχήμα και την υφή των αντικειμένων στο περιβάλλον της. Ο ηχοεντοπισμός της νυχτερίδας είναι τόσο περίπλοκος που αυτά τα ζώα μπορούν να ανιχνεύσουν ένα αντικείμενο στο πλάτος μιας ανθρώπινης τρίχας.

Αν εξιδανικεύσουμε ορισμένα από τα χαρακτηριστικά ηχοεντοπισμού των νυχτερίδων, μπορούμε να αναπτύξουμε διάφορους αλγόριθμους εμπνευσμένους από νυχτερίδες. Για απλότητα, χρησιμοποιούμε τώρα τους ακόλουθους κατά προσέγγιση ή εξιδανικευμένους κανόνες:

1. Όλες οι νυχτερίδες χρησιμοποιούν ηχοεντοπισμό για να αισθανθούν την απόσταση και επίσης «γνωρίζουν» τη διαφορά μεταξύ των φραγμών τροφής/θηράματος και φόντου με κάποιο μαγικό τρόπο.
2. Οι νυχτερίδες πετούν τυχαία με ταχύτητα vi στη θέση xi με σταθερή συχνότητα fmin, μεταβαλλόμενο μήκος κύματος l και ένταση A0 για αναζήτηση θηράματος. Μπορούν να προσαρμόσουν αυτόματα το μήκος κύματος (ή τη συχνότητα) των εκπεμπόμενων παλμών τους και να προσαρμόσουν τον ρυθμό εκπομπής παλμού r στο εύρος [0, 1], ανάλογα με την εγγύτητα του στόχου τους.
3. Αν και η ένταση μπορεί να ποικίλλει με πολλούς τρόπους, υποθέτουμε ότι η ένταση ποικίλλει από ένα μεγάλο (θετικό) A0 έως μια ελάχιστη σταθερή τιμή Amin.

Στις προσομοιώσεις, χρησιμοποιούμε εικονικές νυχτερίδες φυσικά. Πρέπει να ορίσουμε τους κανόνες πώς ενημερώνονται οι θέσεις τους xi και οι ταχύτητες vi σε ένα χώρο αναζήτησης d-διαστατών. Οι νέες λύσεις $X_i^t$ και οι ταχύτητες $V_i^t$ στο χρονικό βήμα t δίνονται από:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \qquad (1)$$

$$V_i^t = V_i^{t-1} + (X_i^t - X_*)f_i \qquad (2)$$

$$X_i^t = X_i^{t-1} + V_i^t \qquad (3)$$

όπου β [ [0, 1] είναι ένα τυχαίο διάνυσμα που προέρχεται από ομοιόμορφη κατανομή. Εδώ το x* είναι η τρέχουσα παγκόσμια καλύτερη θέση (λύση) που βρίσκεται αφού συγκρίνει όλες τις λύσεις μεταξύ όλων των n νυχτερίδων. Καθώς το γινόμενο $\lambda_i$ $f_i$ είναι η αύξηση της ταχύτητας, μπορούμε να χρησιμοποιήσουμε είτε $f_i$ (ή $\lambda_i$ ) για να προσαρμόσουμε την αλλαγή ταχύτητας ενώ καθορίζουμε τον άλλο παράγοντα $\lambda_i$ (ή $f_i$), ανάλογα με τον τύπο του προβλήματος που μας ενδιαφέρει.

Οπότε ο ψευδοκώδικας του αλγορίθμου της νυχτερίδας διαμορφώνεται ως εξής:

---

Αλγόριθμος νυχτερίδας

---

Θέσε αντικειμενική συνάρτηση f(x), $x = (x_1, \dots, x_d)^T$

Αρχικοποίησή του πληθυσμού των νυχτερίδων $X_i$ (i=1,2, … , n) και τις ταχύτητες $v_i$

Ορίστε τη συχνότητα παλμού $f_i$ στο $x_i$

Αρχικοποίησε τους παλμούς $r_i$ και την ένταση $A_i$

Όσο ( t < Μέγιστο αριθμό επαναλήψεων) επανάλαβε

Δημιουργήστε νέες λύσεις προσαρμόζοντας τη συχνότητα και ενημερώνοντας τις ταχύτητες και τις τοποθεσίες/λύσεις [εξισώσεις (1) έως (3)]

Αν ( τυχαίος αριθμός < $r_i$) τότε

Διάλεξε μια λύση ανάμεσα από τις βέλτιστες λύσεις

Δημιούργησε μια τοπική λύση γύρω από την βέλτιστη λύση

Τέλος_αν

Δημιούργησε μια νέα λύση «πετώντας» τυχαία

Αν (τυχαίος αριθμός < $A_i$ & $f(x_i) < f(x^*)$) τότε

Δέξου τις νέες λύσεις

Αύξησε το $r_i$ και ελάττωσέ το $A_i$

Τέλος_αν

Βαθμολογήστε τις νυχτερίδες και βρείτε το τρέχον καλύτερο x*

Τέλος_Επανάληψης

---

Η ενημέρωση των ταχυτήτων και των θέσεων των νυχτερίδων έχει κάποια ομοιότητα με τη διαδικασία της τυπικής βελτιστοποίησης σμήνους σωματιδίων καθώς το $f_i$ ελέγχει ουσιαστικά τον ρυθμό και το εύρος της κίνησης των σωματιδίων που σωρεύονται. Ως ένα βαθμό, το ΒΑ μπορεί να θεωρηθεί ως ένας ισορροπημένος συνδυασμός της τυπικής βελτιστοποίησης του σμήνος σωματιδίων (Particle Swarm) και της εντατικής τοπικής αναζήτησης που ελέγχεται από την ένταση και τον ρυθμό παλμού.

## 5.3 Cuckoo Search

Το Cuckoo Search (CS) είναι ένας από τους πιο πρόσφατους μεταευρετικούς αλγόριθμους εμπνευσμένους από τη φύση, που αναπτύχθηκε το 2009 από τον Xin-She Yang του Πανεπιστημίου του Cambridge και τον Suash Deb του C.V. Raman College of Engineering. Το CS βασίζεται στον παρασιτισμό για την εκτροφή ορισμένων ειδών κούκου. Επιπλέον, αυτός ο αλγόριθμος ενισχύεται από τις λεγόμενες πτήσεις Lévy παρά με απλούς ισοτροπικούς τυχαίους περιπάτους. Πρόσφατες μελέτες δείχνουν ότι το CS είναι δυνητικά πολύ πιο αποτελεσματικό από τους αλγορίθμους βελτιστοποίησης σμήνους σωματιδίων , τους γενετικούς αλγόριθμους και άλλους αλγόριθμους.

Οι κούκοι είναι συναρπαστικά πουλιά, όχι μόνο λόγω των όμορφων ήχων που κάνουν αλλά και λόγω της επιθετικής στρατηγικής αναπαραγωγής τους. Μερικά είδη όπως οι κούκοι ani και Guira γεννούν τα αυγά τους σε κοινόχρηστες φωλιές, αν και μπορούν να αφαιρέσουν τα αυγά άλλων βρεφών για να αυξήσουν την

πιθανότητα εκκόλαψης των δικών τους αυγών. Αρκετός αριθμός ειδών εμπλέκεται σε υποχρεωτικό παρασιτισμό γόνου γεννώντας τα αυγά τους στις φωλιές άλλων πτηνών ξενιστών (συχνά άλλων ειδών).

Από την άλλη πλευρά, διάφορες μελέτες έχουν δείξει ότι η πτητική συμπεριφορά πολλών ζώων και εντόμων έχει δείξει τα τυπικά χαρακτηριστικά των πτήσεων Lévy με χαρακτηριστικά που μοιάζουν με τον νόμο ισχύος. Μια πρόσφατη μελέτη από τους Reynolds και Frye δείχνει ότι οι μύγες των φρούτων, ή Drosophila melanogaster, εξερευνούν το τοπίο τους χρησιμοποιώντας μια σειρά από ευθείες διαδρομές πτήσης που διαστέλλονται από ξαφνικές στροφές 90◦, οδηγώντας σε ένα μοτίβο ελεύθερης αναζήτησης διαλείπουσας κλίμακας τύπου Lévy. Μελέτες της ανθρώπινης συμπεριφοράς, όπως τα πρότυπα αναζήτησης τροφής για κυνηγούς Ju/'hoansi, δείχνουν επίσης το τυπικό χαρακτηριστικό των πτήσεων Lévy. Ακόμη και το φως μπορεί να σχετίζεται με πτήσεις Lévy. Στη συνέχεια, μια τέτοια συμπεριφορά εφαρμόστηκε στη βελτιστοποίηση και τη βέλτιστη αναζήτηση και τα αποτελέσματα δείχνουν την πολλά υποσχόμενη ικανότητά της.

Αυτές οι δύο ιδέες από την φύση είναι που έδωσαν την έμπνευση στον αλγόριθμο CS. Για απλότητα στην περιγραφή του τυπικού CS, εδώ ισχύουν οι ακόλουθοι τρεις εξιδανικευμένους κανόνες:

1. Κάθε κούκος γεννά ένα αυγό τη φορά και το ρίχνει σε μια τυχαία επιλεγμένη φωλιά.
2. Οι καλύτερες φωλιές με αυγά υψηλής ποιότητας θα μεταφερθούν στις επόμενες γενιές.
3. Ο αριθμός των διαθέσιμων φωλιών ξενιστή είναι σταθερός και το αυγό που γεννά ένας κούκος ανακαλύπτεται από το πουλί ξενιστή με πιθανότητα $p_a \in (0, 1)$. Σε αυτή την περίπτωση, το πουλί-ξενιστής μπορεί είτε να απαλλαγεί από το αυγό είτε απλά να εγκαταλείψει τη φωλιά και να φτιάξει μια εντελώς νέα φωλιά.

Ως περαιτέρω προσέγγιση, αυτή η τελευταία υπόθεση μπορεί να προσεγγιστεί αντικαθιστώντας ένα κλάσμα $p_a$ των n φωλιών ξενιστή με νέες φωλιές (με νέες τυχαίες λύσεις). Για ένα πρόβλημα μεγιστοποίησης, η ποιότητα ή η καταλληλότητα μιας λύσης μπορεί απλώς να είναι ανάλογη με την τιμή της αντικειμενικής συνάρτησης.

Από την άποψη της εφαρμογής, μπορούμε να χρησιμοποιήσουμε τις ακόλουθες απλές παραστάσεις ότι κάθε αυγό σε μια φωλιά αντιπροσωπεύει μια λύση και κάθε κούκος μπορεί να γεννήσει μόνο ένα αυγό (αντιπροσωπεύοντας έτσι μια λύση). Ο στόχος είναι να χρησιμοποιηθούν οι νέες και πιθανόν καλύτερες λύσεις (κούκους) για να αντικαταστήσουν μια όχι και τόσο καλή λύση στις φωλιές. Προφανώς, αυτός ο αλγόριθμος μπορεί να επεκταθεί στην πιο περίπλοκη περίπτωση όπου κάθε φωλιά έχει πολλά αυγά που αντιπροσωπεύουν ένα σύνολο λύσεων. Εδώ χρησιμοποιούμε την απλούστερη προσέγγιση, όπου κάθε φωλιά έχει μόνο ένα μόνο αυγό. Σε αυτή την περίπτωση, δεν υπάρχει διάκριση μεταξύ αυγού, φωλιάς ή κούκου, αφού κάθε φωλιά αντιστοιχεί σε ένα αυγό, το οποίο αντιπροσωπεύει επίσης έναν κούκο.

Αυτός ο αλγόριθμος χρησιμοποιεί έναν ισορροπημένο συνδυασμό ενός τοπικού τυχαίου περιπάτου και του καθολικού διερευνητικού τυχαίου περιπάτου, που ελέγχεται από μια παράμετρο μεταγωγής pa. Ο τοπικός τυχαίος περίπατος μπορεί να γραφτεί ως

$$x_\iota^{t+1} = x_\iota^t + \alpha s \otimes H(p_a - ) \otimes (x_j^t - x_k^t) \quad (1)$$

όπου $x_j^t$ και $x_k^t$ είναι δύο διαφορετικές λύσεις που επιλέγονται τυχαία με τυχαία μετάθεση, το H(u) είναι μια συνάρτηση Heaviside, μια συνάρτηση βήματος που πήρε το όνομά της από τον Oliver Heaviside (1850–1925), η τιμή της οποίας είναι 0 για αρνητικά ορίσματα και 1 για θετικά ορίσματα., είναι ένας τυχαίος αριθμός που προκύπτει από μια ομοιόμορφη κατανομή και s είναι το μέγεθος του βήματος. Εδώ, ⊗ σημαίνει το εισαγωγικό γινόμενο δύο διανυσμάτων.

Από την άλλη πλευρά, ο παγκόσμιος τυχαίος περίπατος πραγματοποιείται χρησιμοποιώντας πτήσεις Lévy

$x_i^{t+1} = x_i^t + \alpha L(s,\lambda)$

όπου $L(s, \lambda) = \lambda\Gamma(\lambda) \sin(\pi\lambda/2) / \pi * 1/ s^{1+\lambda}$, $(s >> s0 > 0)$

Εδώ $\alpha > 0$ είναι ο συντελεστής κλιμάκωσης μεγέθους βήματος, ο οποίος θα πρέπει να σχετίζεται με τις κλίμακες του προβλήματος που μας ενδιαφέρει.

Με βάση όλα τα παραπάνω, τα βασικά βήματα του CS μπορούν να συνοψιστούν ως ο εξής ψευδοκώδικας

---

Αλγόριθμος Cuckoo Search

---

Θέσε αντικειμενική συνάρτηση $f(x)$, $x = (x_1, \dots, x_d)^T$

Δημιουργήστε αρχικό πληθυσμό n φωλιών ξενιστή xi

Όσο ( t < Μέγιστη γενιά κούκων) ή ( κριτήριο τέλους)

Διάλεξε έναν κούκο τυχαία

Δημιούργησε λύση μέσω πτήσης Lévy

Αξιολογήστε την ποιότητα της λύσης ή την αντικειμενική τιμή $f_i$

Επιλέξτε μια φωλιά μεταξύ n (ας πούμε, j) τυχαία

Αν ($f_i < f_j$) τότε

Αντικαταστήστε το j με τη νέα λύση i

Τέλος_αν

Ένα κλάσμα ($p_a$) χειρότερων φωλιών εγκαταλείπονται

Οι νέες φωλιές/λύσεις κατασκευάζονται/δημιουργούνται από την Εξ.(1)

Κρατήστε τις καλύτερες λύσεις (ή φωλιές με ποιοτικές λύσεις)

Βαθμολογήστε τις λύσεις και βρείτε την τρέχουσα καλύτερη

Ενημέρωση $t \leftarrow t + 1$

Τέλος_επανάληψης

---

Ο αλγόριθμος Cuckoo Search είναι από τους πιο αποδοτικούς διότι έχει δύο δυνατότητες αναζήτησης: τοπική αναζήτηση και καθολική αναζήτηση, που ελέγχονται από μια πιθανότητα εναλλαγής/ανακάλυψης. Η τοπική αναζήτηση είναι πολύ εντατική, με περίπου το 1/4 του χρόνου αναζήτησης (για $p_a = 0,25$), ενώ η καθολική αναζήτηση διαρκεί περίπου τα 3/4 του συνολικού χρόνου αναζήτησης. Αυτό επιτρέπει την αποτελεσματικότερη εξερεύνηση του χώρου αναζήτησης σε καθολική κλίμακα, και κατά συνέπεια η συνολική βελτιστοποίηση μπορεί να βρεθεί με μεγαλύτερη πιθανότητα.

Ένα επιπλέον πλεονέκτημα του CS είναι ότι η καθολική αναζήτησή του χρησιμοποιεί πτήσεις ή διαδικασίες Lévy αντί για τυπικούς τυχαίους περιπάτους. Επειδή οι πτήσεις Lévy έχουν άπειρο μέσο όρο και διακύμανση, το CS μπορεί να εξερευνήσει τον χώρο αναζήτησης πιο αποτελεσματικά από τους αλγόριθμους που χρησιμοποιούν τυπικές διαδικασίες Gaussian. Αυτό το πλεονέκτημα, σε συνδυασμό με τις τοπικές δυνατότητες και τις δυνατότητες αναζήτησης και την εγγυημένη καθολική σύγκλιση, καθιστά το CS πολύ αποτελεσματικό.

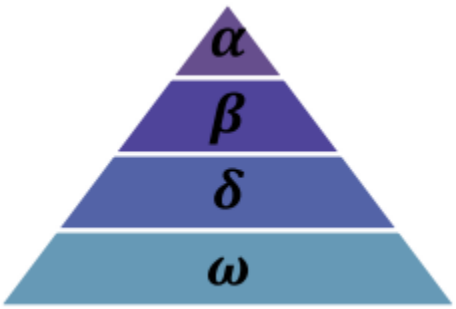
## 5.4 Grey Wolf Optimizer

Ο GWO είναι ένας αλγόριθμος βασισμένος στους γκρι λύκους που ζούνε σε αγέλες και έχουν ιεραρχία. Σε αυτό το έργο αυτή η τεχνική κυνηγιού και η κοινωνική ιεραρχία των γκρίζων λύκων μοντελοποιούνται μαθηματικά προκειμένου να σχεδιαστεί το GWO και να πραγματοποιηθεί βελτιστοποίηση.

Η ιεραρχία όπως φαίνεται στην εικόνα 4 κατατάσσει τους λύκους σε 4 επίπεδα:

- **Άλφα**: Το άλφα είναι ως επί το πλείστων υπεύθυνο για τη λήψη αποφάσεων σχετικά με το κυνήγι, τον τόπο ύπνου, την ώρα του ξυπνήματος και ούτω καθεξής. Οι αποφάσεις του άλφα πρέπει να ακολουθούνται από την αγέλη.
- **Βήτα:** Οι βήτα είναι δευτερεύοντες λύκοι που βοηθούν το άλφα στη λήψη αποφάσεων ή σε άλλες δραστηριότητες αγέλης. Ο βήτα λύκος είναι ο καλύτερος υποψήφιος για να γίνει ο άλφα σε περίπτωση που ένας από τους λύκους άλφα πεθάνει ή γίνει πολύ μεγάλος. Ο βήτα λύκος θα πρέπει να σέβεται το άλφα, αλλά να διοικεί και τους άλλους λύκους χαμηλότερου επιπέδου.
- **Δέλτα:** Εάν ένας λύκος δεν είναι άλφα, βήτα ή ωμέγα, ονομάζεται. Οι λύκοι δέλτα πρέπει να υποταχθούν στα άλφα και τα βήτα, αλλά κυριαρχούν στα ωμέγα.
- **Ωμέγα:** Ο γκρίζος λύκος με τη χαμηλότερη κατάταξη είναι το ωμέγα. Οι λύκοι ωμέγα πρέπει πάντα να υποτάσσονται σε όλους τους άλλους κυρίαρχους λύκους.

*Εικόνα 4. Ιεραρχία γκρι λύκων*




Εκτός από την κοινωνική ιεραρχία των λύκων, το ομαδικό κυνήγι είναι μια άλλη ενδιαφέρουσα κοινωνική συμπεριφορά των γκρίζων λύκων. Οι κύριες φάσεις του κυνηγιού του γκρίζου λύκου είναι οι εξής:

- Παρακολούθηση, κυνήγι και προσέγγιση του θηράματος.
- Καταδίωξη, περικύκλωση και παρενόχληση του θηράματος μέχρι να σταματήσει να κινείται.
- Επίθεση προς το θήραμα.

Προκειμένου να μοντελοποιήσουμε μαθηματικά την κοινωνική ιεραρχία των λύκων κατά το σχεδιασμό του GWO, θεωρούμε την καταλληλότερη λύση ως το άλφα (α). Κατά συνέπεια, η δεύτερη και η τρίτη καλύτερη λύση ονομάζονται βήτα (β) και δέλτα (δ) αντίστοιχα. Τα υπόλοιπα υποψήφια διαλύματα θεωρούνται ωμέγα (ω). Στον αλγόριθμο GWO το κυνήγι (βελτιστοποίηση) καθοδηγείται από τα α, β και δ. Οι ω λύκοι ακολουθούν αυτούς τους τρεις λύκους.

### 5.4.1 Περικύκλωση του θηράματος

Όπως αναφέρθηκε παραπάνω, οι γκρίζοι λύκοι περικυκλώνουν το θήραμα κατά τη διάρκεια του κυνηγιού. Προκειμένου να μοντελοποιηθεί μαθηματικά η κυκλική συμπεριφορά, προτείνονται οι ακόλουθες εξισώσεις:

$$D = | C * X_p(t) - X(t) | \qquad (1)$$

$$X(t+1) = X_p(t) - A * D \qquad (2)$$

όπου t υποδηλώνει την τρέχουσα επανάληψη, A και C είναι διανύσματα συντελεστών, $X_p$ είναι το διάνυσμα θέσης του θηράματος και X δείχνει το διάνυσμα θέσης ενός γκρίζου λύκου.

Τα διανύσματα A και C υπολογίζονται ως εξής:

$$A = 2a*r_1 - a \qquad (3)$$

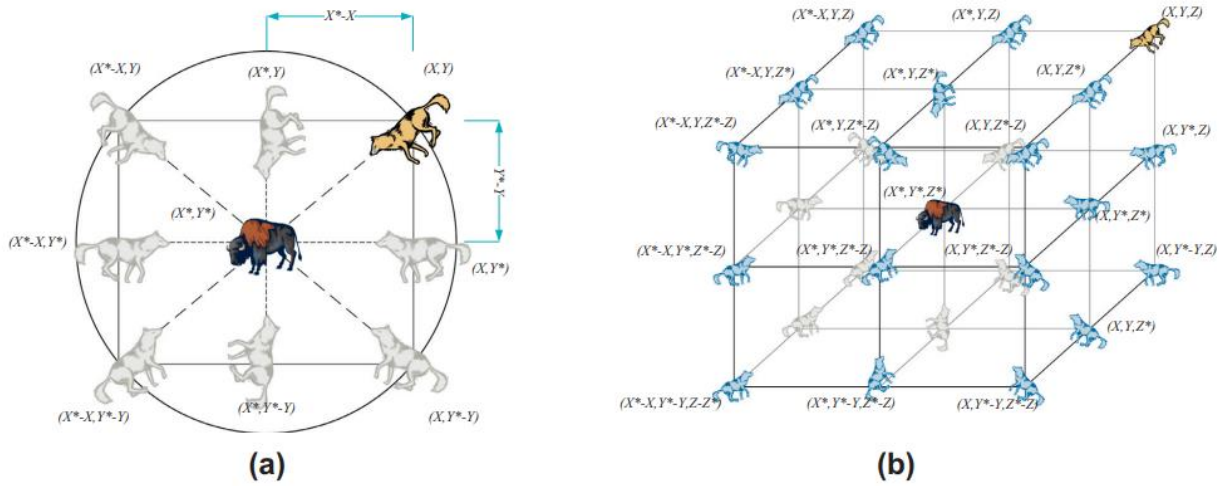
$$C = 2 * r_2 \qquad (4)$$

όπου τα συστατικά του a μειώνονται γραμμικά από 2 σε 0 κατά τη διάρκεια των επαναλήψεων και τα $r_1$, $r_2$ είναι τυχαία διανύσματα στο [0, 1].

Τα αποτελέσματα των Εξ. (1) και (2), ένα δισδιάστατο διάνυσμα θέσης και μερικοί από τους πιθανούς γείτονες απεικονίζονται στην Εικόνα 5(α). Όπως φαίνεται σε αυτό το σχήμα, ένας γκρίζος λύκος στη θέση (X, Y) μπορεί να ενημερώσει τη θέση του σύμφωνα με τη θέση του θηράματος (X* , Y*). Διαφορετικές θέσεις γύρω από τον καλύτερο παράγοντα μπορούν να προσεγγιστούν σε σχέση με την τρέχουσα θέση προσαρμόζοντας την τιμή των διανυσμάτων A και C.

Για παράδειγμα, το (X* –X, Y* ) μπορεί να επιτευχθεί ρυθμίζοντας A=(1,) και C = (1,1). Οι πιθανές ενημερωμένες θέσεις ενός γκρίζου λύκου στον τρισδιάστατο χώρο απεικονίζονται στην Εικόνα 5(β). Σημειώστε ότι τα τυχαία διανύσματα $r_1$ και $r_2$ επιτρέπουν στους λύκους να φτάσουν σε οποιαδήποτε θέση μεταξύ των σημείων που απεικονίζονται στο στην Εικόνα 5. Έτσι, ένας γκρίζος λύκος μπορεί να ενημερώσει τη θέση του μέσα στο χώρο γύρω από το θήραμα σε οποιαδήποτε τυχαία θέση χρησιμοποιώντας τις Εξ. (1) και (2).

*Εικόνα 5. Δισδιάστατη και τρισδιάστατη απεικόνιση πιθανών θέσεων*




### 5.4.2 Κυνήγι

Οι γκρίζοι λύκοι έχουν την ικανότητα να αναγνωρίζουν τη θέση του θηράματος και να τους περικυκλώνουν. Το κυνήγι συνήθως καθοδηγείται από το άλφα. Το βήτα και το δέλτα μπορεί επίσης να συμμετέχουν στο κυνήγι περιστασιακά. Ωστόσο, σε έναν αφηρημένο χώρο αναζήτησης δεν έχουμε ιδέα για τη θέση του βέλτιστου (θηράματος). Προκειμένου να προσομοιωθεί μαθηματικά η κυνηγετική συμπεριφορά των γκρίζων λύκων, υποθέτουμε ότι η άλφα (καλύτερη υποψήφια λύση) βήτα και δέλτα έχουν καλύτερη γνώση σχετικά με την πιθανή θέση του θηράματος. Επομένως, αποθηκεύουμε τις πρώτες τρεις καλύτερες λύσεις που έχουμε αποκτήσει μέχρι στιγμής και υποχρεώνουμε τους άλλους πράκτορες αναζήτησης (συμπεριλαμβανομένων των ωμέγα) να ενημερώσουν τις θέσεις τους σύμφωνα με τη θέση των καλύτερων πρακτόρων αναζήτησης. Οι ακόλουθοι τύποι προτείνονται σχετικά.

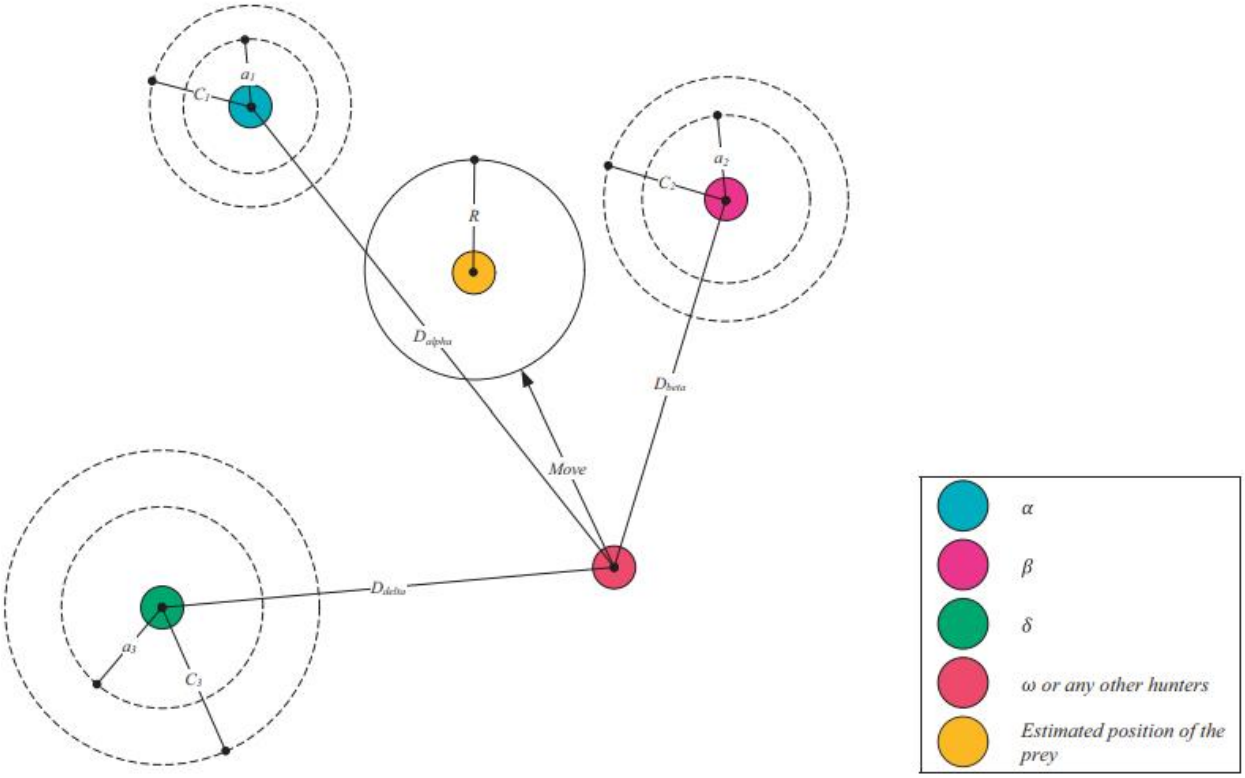
$$D_\alpha = | C_1 * X_\alpha - X | , D_\beta = | C_2 * X_\beta - X | , D_\delta = | C_3 * X_\delta - X | \quad (5)$$

$$X_1 = X_\alpha - A_1 * D_\alpha , X_2 = X_\beta - A_2 * D_\beta , X_3 = X_\delta - A_3 * D_\delta \quad (6)$$

$$X(t+1) = (X_1 + X_2 + X_3) / 3 \quad (7)$$

Η εικόνα 6 δείχνει πώς ένας πράκτορας αναζήτησης ενημερώνει τη θέση του σύμφωνα με τα άλφα, βήτα και δέλτα σε ένα χώρο αναζήτησης 2D. Μπορεί να παρατηρηθεί ότι η τελική θέση θα βρίσκεται σε μια τυχαία θέση μέσα σε έναν κύκλο που ορίζεται από τις θέσεις των άλφα, βήτα και δέλτα στον χώρο αναζήτησης. Με άλλα λόγια, το άλφα, το βήτα και το δέλτα εκτιμούν τη θέση του θηράματος και άλλοι λύκοι ενημερώνουν τυχαία τις θέσεις τους γύρω από το θήραμα.

*Εικόνα 6. Ενημέρωση θέσης στο GWO*




### 5.4.3 Επίθεση στο θήραμα (εκμετάλλευση)

Όπως αναφέρθηκε παραπάνω, οι γκρίζοι λύκοι τελειώνουν το κυνήγι επιτιθέμενοι στο θήραμα όταν αυτό σταματήσει να κινείται. Για να μοντελοποιήσουμε μαθηματικά την προσέγγιση του θηράματος μειώνουμε την τιμή του a. Σημειώστε ότι το εύρος διακύμανσης του A μειώνεται επίσης κατά a. Με άλλα λόγια το A είναι μια τυχαία τιμή στο διάστημα [2a, 2a] όπου το a μειώνεται από 2 σε 0 κατά τη διάρκεια των επαναλήψεων. Όταν οι τυχαίες τιμές του A είναι στο [ 1,1], η επόμενη θέση ενός πράκτορα αναζήτησης μπορεί να είναι σε οποιαδήποτε θέση μεταξύ της τρέχουσας θέσης του και της θέσης του θηράματος. Η εικόνα 5(α) δείχνει ότι |A| < 1 αναγκάζει τους λύκους να επιτεθούν προς το θήραμα.

Με τους τελεστές που έχουν προταθεί μέχρι στιγμής, ο αλγόριθμος GWO επιτρέπει στους πράκτορες αναζήτησής του να ενημερώνουν τη θέση τους με βάση τη θέση των άλφα, βήτα και δέλτα και να επιτίθενται προς το θήραμα. Ωστόσο, ο αλγόριθμος GWO είναι επιρρεπής σε στασιμότητα στις τοπικές λύσεις με αυτούς τους τελεστές. Είναι αλήθεια ότι ο προτεινόμενος μηχανισμός περικύκλωσης δείχνει εξερεύνηση σε κάποιο βαθμό, αλλά το GWO χρειάζεται περισσότερους χειριστές για να δώσει έμφαση στην εξερεύνηση.

### 5.4.4 Αναζήτηση για θήραμα (εξερεύνηση)

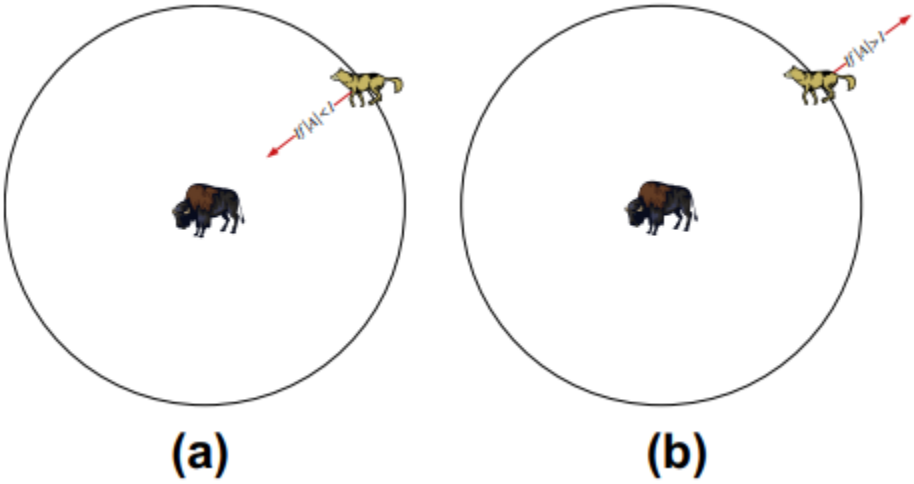
Οι γκρίζοι λύκοι αναζητούν ως επί το πλείστων σύμφωνα με τη θέση των άλφα, βήτα και δέλτα. Αποκλίνουν μεταξύ τους για να ψάξουν για θήραμα και συγκλίνουν για να επιτεθούν στο θήραμα. Για να μοντελοποιήσουμε μαθηματικά την απόκλιση, χρησιμοποιούμε το A με τυχαίες τιμές μεγαλύτερες από 1 ή μικρότερες από -1 για να υποχρεώσουμε τον πράκτορα αναζήτησης να αποκλίνει από το θήραμα. Αυτό

δίνει έμφαση στην εξερεύνηση και επιτρέπει στον αλγόριθμο GWO να κάνει παγκόσμια αναζήτηση. Η εικόνα 7(β) δείχνει επίσης ότι |A| > 1 αναγκάζει τους γκρίζους λύκους να αποκλίνουν από το θήραμα για να ελπίζουμε ότι θα βρουν ένα καλύτερο θήραμα (λύση).

Ένα άλλο συστατικό του GWO που ευνοεί την εξερεύνηση είναι το C. Όπως φαίνεται στην Εξ. (4), το διάνυσμα C περιέχει τυχαίες τιμές στο [0,2]. Αυτή η συνιστώσα παρέχει τυχαία βάρη για το θήραμα προκειμένου να τονίσει στοχαστικά (C > 1) ή να αποτονίσει (C < 1) την επίδραση του θηράματος στον καθορισμό της απόστασης στην Εξ. (1). Αυτό βοηθά το GWO να δείξει μια πιο τυχαία συμπεριφορά κατά τη διάρκεια της βελτιστοποίησης, ευνοώντας την εξερεύνηση και την τοπική αποφυγή βέλτιστων. Αξίζει να αναφέρουμε εδώ ότι το C δεν μειώνεται γραμμικά σε αντίθεση με το A. Απαιτούμε σκόπιμα το C να παρέχει τυχαίες τιμές ανά πάσα στιγμή για να τονίσουμε την εξερεύνηση όχι μόνο κατά τις αρχικές επαναλήψεις αλλά και τις τελικές επαναλήψεις. Αυτό το στοιχείο είναι πολύ χρήσιμο σε περίπτωση τοπικής στασιμότητας, ειδικά στις τελικές επαναλήψεις.

Το διάνυσμα C μπορεί επίσης να θεωρηθεί ως η επίδραση των εμποδίων στην προσέγγιση του θηράματος στη φύση. Γενικά, τα εμπόδια στη φύση εμφανίζονται στα κυνηγετικά μονοπάτια των λύκων και μάλιστα τους εμποδίζουν να πλησιάσουν γρήγορα και άνετα το θήραμα. Αυτό ακριβώς κάνει το διάνυσμα C. Ανάλογα με τη θέση του λύκου, μπορεί τυχαία να δώσει βάρος στο θήραμα και να κάνει όλο και πιο δύσκολο να φτάσει τους λύκους ή το αντίστροφο.

*Εικόνα 7. Επίθεση (a) και αναζήτηση (b)*




Συνοψίζοντας, η διαδικασία αναζήτησης ξεκινά με τη δημιουργία ενός τυχαίου πληθυσμού γκρίζων λύκων (υποψήφιες λύσεις) στον αλγόριθμο GWO. Κατά τη διάρκεια των επαναλήψεων, οι λύκοι άλφα, βήτα και δέλτα εκτιμούν την πιθανή θέση του θηράματος. Κάθε υποψήφια λύση ενημερώνει την απόστασή της από το θήραμα. Η παράμετρος a μειώνεται από 2 σε 0 για να τονιστεί η εξερεύνηση και η εκμετάλλευση, αντίστοιχα. Οι υποψήφιες λύσεις τείνουν να αποκλίνουν από το θήραμα όταν |A| > 1 και συγκλίνουν προς το θήραμα όταν |A| < 1. Τέλος, ο αλγόριθμος GWO τερματίζεται με την ικανοποίηση ενός κριτηρίου τέλους.

Οπότε ο αλγόριθμος GWO δίνεται από τον ψευδοκώδικα:

Αλγόριθμος Grey Wolf Optimizer

Δημιούργησε τον πληθυσμό τον λύκων $X_i$ (I = 1,2, … , n)

Αρχικοποίησε τα α , A και C

Υπολόγισε την καταλληλότητα κάθε πράκτορα αναζήτησης

$X_\alpha$ = ο καλύτερος πράκτορας αναζήτησης

$X_\beta$ = ο δεύτερος καλύτερος πράκτορας αναζήτησης

$X_\delta$ = ο τρίτος καλύτερος πράκτορας αναζήτησης

Όσο ( t < Μέγιστος αριθμός επαναλήψεων) επανάλαβε

Για κάθε πράκτορα αναζήτησης

Ενημέρωσε τη θέση του κάθε πράκτορα αναζήτησης Εξ(7)

Τέλος_επανάληψης

Ενημέρωσε τα α, A και C

Υπολόγισε την καταλληλότητα κάθε πράκτορα αναζήτησης

Ενημέρωσε τα $X_\alpha$ , $X_\beta$ , $X_\delta$

t = t + 1

Τέλος_επανάληψης

Επέστρεψε $X_\alpha$

---

Το GWO είναι θεωρητικά ικανό να επιλύει προβλήματα βελτιστοποίησης, μπορούν να σημειωθούν ορισμένα σημεία:

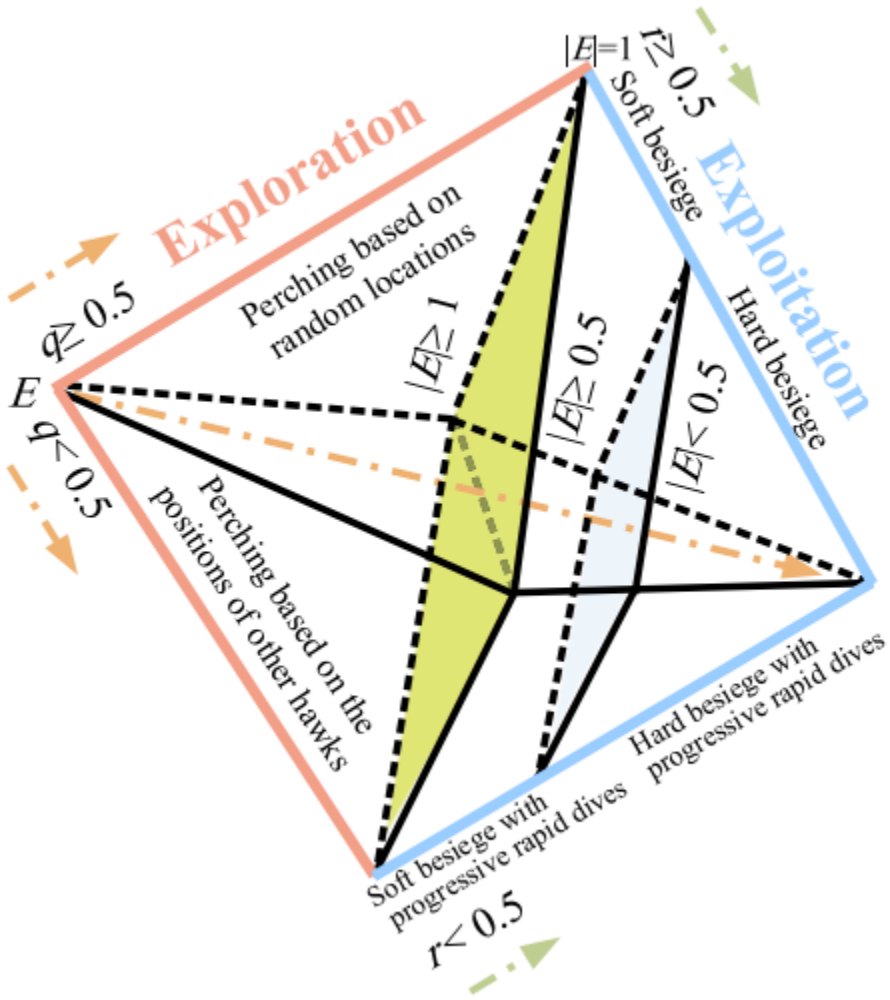
- Η προτεινόμενη κοινωνική ιεραρχία βοηθά το GWO να αποθηκεύσει τις καλύτερες λύσεις που έχουν ληφθεί μέχρι στιγμής κατά τη διάρκεια της επανάληψης.
- Ο προτεινόμενος περικυκλωτικός μηχανισμός ορίζει μια γειτονιά σε σχήμα κύκλου γύρω από τις λύσεις που μπορεί να επεκταθεί σε υψηλότερες διαστάσεις ως υπερ-σφαίρα.
- Οι τυχαίες παράμετροι A και C βοηθούν τις υποψήφιες λύσεις έχουν υπερ-σφαίρες με διαφορετικές τυχαίες ακτίνες.
- Η προτεινόμενη μέθοδος κυνηγιού επιτρέπει στις υποψήφιες λύσεις να εντοπίσουν την πιθανή θέση του θηράματος.
- Η εξερεύνηση και η εκμετάλλευση είναι εγγυημένες από τις προσαρμοστικές τιμές των a και A.
- Οι προσαρμοστικές τιμές των παραμέτρων a και A επιτρέπουν στο GWO να μεταβαίνει ομαλά μεταξύ εξερεύνησης και εκμετάλλευσης.
- Με μείωση του A, οι μισές από τις επαναλήψεις αφιερώνονται στην εξερεύνηση ($|A| >> 1$) και οι άλλες μισές αφιερώνονται στην εκμετάλλευση ($|A| <1$).
- Το GWO έχει μόνο δύο κύριες παραμέτρους προς ρύθμιση (α και C).

## 5.5 Harris Hawks Optimization

Ένας αλγόριθμος βελτιστοποίησης εμπνευσμένος από τη συμπεριφορά των γερακιών Harris κατά τη διάρκεια του κυνηγιού τους. Η κύρια τακτική των γερακιών Harris για να συλλάβουν ένα θήραμα είναι η «έκπληξη. Σε αυτή την έξυπνη στρατηγική, πολλά γεράκια προσπαθούν να επιτεθούν συνεργατικά από διαφορετικές κατευθύνσεις και ταυτόχρονα συγκλίνουν σε ένα κουνέλι που διαφεύγει που έχει εντοπιστεί έξω από το κάλυμμα. Η επίθεση μπορεί να ολοκληρωθεί γρήγορα με τη σύλληψη του αιφνιδιασμένου θηράματος σε λίγα δευτερόλεπτα, αλλά περιστασιακά, όσον αφορά τις δυνατότητες διαφυγής και τις συμπεριφορές του θηράματος, η επίθεση μπορεί να περιλαμβάνουν πολλαπλές, μικρού μήκους, γρήγορες βουτιές κοντά στο θήραμα για αρκετά λεπτά. Τα γεράκια Harris μπορούν να επιδείξουν μια ποικιλία από στυλ κυνηγιού που εξαρτώνται από τη δυναμική φύση των περιστάσεων και τα μοτίβα διαφυγής ενός θηράματος. Μια τακτική αλλαγής εμφανίζεται όταν το καλύτερο γεράκι (αρχηγός) σκύβει στο θήραμα και χάνεται, και το κυνηγητό θα συνεχιστεί από ένα από τα μέλη του κόμματος. Αυτές οι δραστηριότητες εναλλαγής μπορούν να παρατηρηθούν σε διαφορετικές καταστάσεις επειδή είναι ευεργετικές για τη σύγχυση του κουνελιού που δραπετεύει. Το κύριο πλεονέκτημα αυτών των συνεργατικών τακτικών είναι ότι τα γεράκια Harris μπορούν να κυνηγήσουν το κουνέλι που έχει εντοπιστεί μέχρι εξάντλησης, γεγονός που αυξάνει την ευπάθειά του.

Μοντελοποιήθηκαν, έτσι, οι εξερευνητικές και εκμεταλλευτικές φάσεις του προτεινόμενου HHO, εμπνευσμένες από την εξερεύνηση ενός θηράματος, την έκπληξη και τις διαφορετικές στρατηγικές επίθεσης των γερακιών Harris. Το HHO είναι μια τεχνική βελτιστοποίησης που βασίζεται στον πληθυσμό, χωρίς βαθμούς. Ως εκ τούτου, μπορεί να εφαρμοστεί σε οποιοδήποτε πρόβλημα βελτιστοποίησης με την επιφύλαξη μιας σωστής διατύπωσης. Η εικόνα 8 δείχνει όλες τις φάσεις του HHO, οι οποίες περιγράφονται στις επόμενες υποενότητες.

*Εικόνα 8. Διαφορετικές Φάσεις HHO*




### 5.5.1. Φάση εξερεύνησης

Σε αυτό το μέρος προτείνεται ο μηχανισμός εξερεύνησης του HHO. Αν λάβουμε υπόψη τη φύση των γερακιών Harris, μπορούν να παρακολουθήσουν και να ανιχνεύσουν το θήραμα από τα δυνατά τους μάτια, αλλά μερικές φορές το θήραμα δεν μπορεί να δει εύκολα. Ως εκ τούτου, τα γεράκια περιμένουν, παρατηρούν και παρακολουθούν την τοποθεσία της ερήμου για να εντοπίσουν ένα θήραμα ίσως μετά από αρκετές ώρες. Στο HHO, τα γεράκια Harris είναι οι υποψήφιες λύσεις και η καλύτερη υποψήφια λύση σε κάθε βήμα θεωρείται ως το επιδιωκόμενο θήραμα ή σχεδόν η βέλτιστη. Στο HHO, τα γεράκια Harris κουρνιάζουν τυχαία σε ορισμένες τοποθεσίες και περιμένουν να εντοπίσουν ένα θήραμα με βάση δύο στρατηγικές. Αν λάβουμε υπόψη μια ίση πιθανότητα q για κάθε στρατηγική κούρνιασης, κουρνιάζουν με βάση τις θέσεις των άλλων μελών της οικογένειας (για να είναι αρκετά κοντά τους όταν επιτίθενται) και του κουνελιού, το οποίο διαμορφώνεται στην Εξ. (1) για την συνθήκη q < 0,5, ή κουρνιάζει σε τυχαία ψηλά δέντρα (τυχαίες τοποθεσίες εντός της περιοχής κατοικίας της ομάδας), η οποία μοντελοποιείται στην Εξ. (1) για συνθήκη q ≥ 0,5.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 |X_{rand}(t) - 2r_2 X(t)| , & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) , & q < 0.5 \end{cases} \quad (1)$$

όπου X(t + 1) είναι το διάνυσμα θέσης των γερακιών στην επόμενη επανάληψη t, $X_{rabbit}(t)$ είναι η θέση του κουνελιού, X(t) είναι το διάνυσμα της τρέχουσας θέσης των γερακιών, $r_1$, $r_2$, $r_3$, $r_4$ και q είναι τυχαίοι αριθμοί μέσα στο (0,1), οι οποίοι ενημερώνονται σε κάθε επανάληψη, τα LB και UB δείχνουν τα άνω και κάτω όρια των μεταβλητών, το $X_{rand}(t)$ είναι ένα τυχαία επιλεγμένο γεράκι από τον τρέχοντα πληθυσμό και το $X_m$ είναι η μέση θέση του σημερινού πληθυσμού των γερακιών.

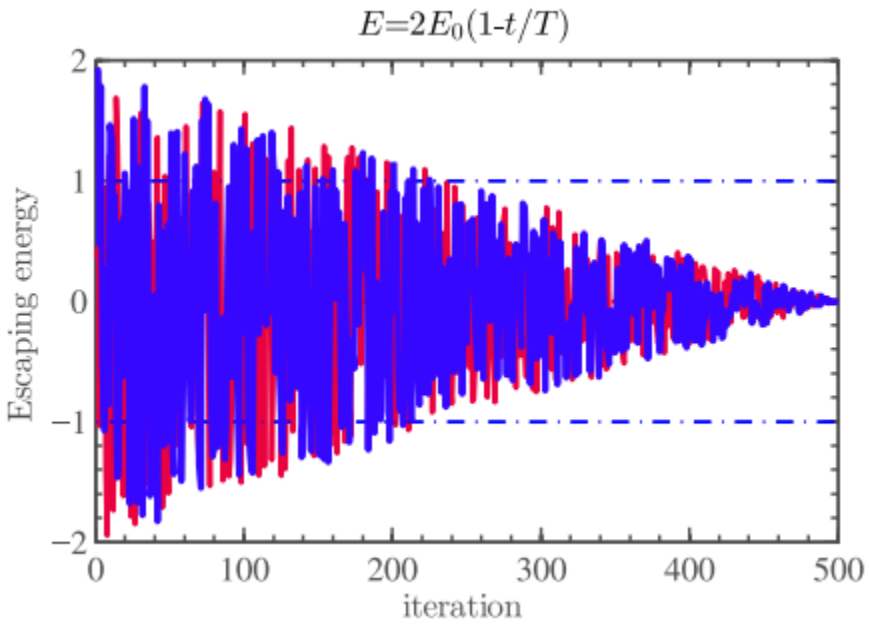
### 5.5.2. Μετάβαση από την εξερεύνηση στην εκμετάλλευση

Ο αλγόριθμος HHO μπορεί να μεταφερθεί από την εξερεύνηση στην εκμετάλλευση και, στη συνέχεια, να αλλάξει μεταξύ διαφορετικών συμπεριφορών εκμετάλλευσης με βάση την ενέργεια διαφυγής του θηράματος. Η ενέργεια ενός θηράματος μειώνεται σημαντικά κατά τη συμπεριφορά διαφυγής. Για να μοντελοποιήσουμε αυτό το γεγονός, η ενέργεια ενός θηράματος μοντελοποιείται ως εξής:

$$E = 2E_0(1 - t / T)$$

όπου το E δείχνει την ενέργεια διαφυγής του θηράματος, το T είναι ο μέγιστος αριθμός επαναλήψεων και το $E_0$ είναι η αρχική κατάσταση της ενέργειάς του. Στο HHO, το $E_0$ αλλάζει τυχαία μέσα στο διάστημα (−1, 1) σε κάθε επανάληψη. Όταν η τιμή του $E_0$ μειώνεται από 0 σε −1, το κουνέλι σημαδεύεται φυσικά, ενώ όταν η τιμή του $E_0$ αυξάνεται από 0 σε 1, σημαίνει ότι το κουνέλι δυναμώνει. Η δυναμική ενέργεια διαφυγής E έχει πτωτική τάση κατά τις επαναλήψεις. Όταν η ενέργεια που διαφεύγει |E| ≥1, τα γεράκια αναζητούν διαφορετικές περιοχές για να εξερευνήσουν μια τοποθεσία κουνελιού, επομένως, το HHO εκτελεί τη φάση εξερεύνησης και όταν |E| <1, ο αλγόριθμος προσπαθεί να εκμεταλλευτεί τη γειτονιά των λύσεων κατά τη διάρκεια των βημάτων εκμετάλλευσης. Εν ολίγοις, η εξερεύνηση γίνεται όταν |E| ≥1, ενώ η εκμετάλλευση γίνεται σε μεταγενέστερα βήματα όταν |E| <1. Η εξαρτώμενη από το χρόνο συμπεριφορά του E παρουσιάζεται επίσης στην εικόνα 9.

*Εικόνα 9. Συμπεριφορά για 500 επαναλήψεις*



### 5.5.3. Φάση εκμετάλλευσης

Σε αυτή τη φάση, τα γεράκια του Χάρις εκτελούν την αιφνιδιαστική επίθεση επιτιθέμενοι στο επιδιωκόμενο θήραμα που εντοπίστηκε στην προηγούμενη φάση. Ωστόσο, τα θηράματα συχνά προσπαθούν να ξεφύγουν από επικίνδυνες καταστάσεις. Ως εκ τούτου, διαφορετικά στυλ κυνηγήματος εμφανίζονται σε πραγματικές καταστάσεις. Σύμφωνα με τις συμπεριφορές διαφυγής του θηράματος και τις στρατηγικές καταδίωξης των γερακιών του Χάρις, προτείνονται τέσσερις πιθανές στρατηγικές στο HHO για να μοντελοποιήσουν το στάδιο της επίθεσης. Τα θηράματα προσπαθούν πάντα να ξεφύγουν από απειλητικές καταστάσεις. Ας υποθέσουμε ότι το r είναι η πιθανότητα ενός θηράματος να δραπετεύσει επιτυχώς ($r < 0,5$) ή να μην δραπετεύσει επιτυχώς ($r \geq 0,5$) πριν από την έκπληξη. Ό,τι κι αν κάνει το θήραμα, τα γεράκια θα εκτελέσουν μια σκληρή ή ήπια πολιορκία για να πιάσουν το θήραμα. Σημαίνει ότι θα περικυκλώσουν το θήραμα από διαφορετικές κατευθύνσεις απαλά ή σκληρά ανάλογα με τη διατηρούμενη ενέργεια του θηράματος. Σε πραγματικές καταστάσεις, τα γεράκια πλησιάζουν όλο και πιο κοντά στο επιδιωκόμενο θήραμα για να αυξήσουν τις πιθανότητές τους να σκοτώσουν συνεργατικά το κουνέλι εκτελώντας την έκπληξη. Μετά από αρκετά λεπτά, το θήραμα που δραπετεύει θα χάνει όλο και περισσότερη ενέργεια. τότε, τα γεράκια εντείνουν τη διαδικασία της πολιορκίας για να πιάσουν αβίαστα το εξαντλημένο θήραμα. Για να μοντελοποιηθεί αυτή η στρατηγική και να επιτραπεί στο HHO να εναλλάσσεται μεταξύ ήπιας και σκληρής διαδικασίας πολιορκίας, χρησιμοποιείται η παράμετρος E. Από την άποψη αυτή, όταν $|E| \geq 0,5$, συμβαίνει η ήπια πολιορκία και όταν $|E| < 0,5$, γίνεται η σκληρή πολιορκία.

#### *5.5.3.1. Ήπια πολιορκία*

Όταν $r \geq 0,5$ και $|E| \geq 0,5$, το κουνέλι έχει ακόμα αρκετή ενέργεια και προσπαθεί να ξεφύγει με κάποια τυχαία παραπλανητικά άλματα αλλά τελικά δεν μπορεί. Κατά τη διάρκεια αυτών των προσπαθειών, τα γεράκια Harris το περικυκλώνουν ήπια για να κάνουν το κουνέλι πιο εξουθενωμένο και μετά εκτελούν την έκπληξη. Αυτή η συμπεριφορά διαμορφώνεται από τους ακόλουθους κανόνες:

$$X(t + 1) = \Delta X(t) - E\,|JX_{rabbit}(t) - X(t)|$$

$$\Delta X(t) = X_{rabbit}(t) - X(t)$$

όπου $\Delta X(t)$ είναι η διαφορά μεταξύ του διανύσματος θέσης του κουνελιού και της τρέχουσας θέσης στην επανάληψη t, ο r5 είναι ένας τυχαίος αριθμός μέσα στο (0,1) και το $J = 2(1 - r_5)$ αντιπροσωπεύει την ισχύ του τυχαίου άλματος του κουνελιού καθ' όλη τη διάρκεια της διαδικασίας διαφυγής. Η τιμή J αλλάζει τυχαία σε κάθε επανάληψη για να προσομοιώσει τη φύση των κινήσεων του κουνελιού.

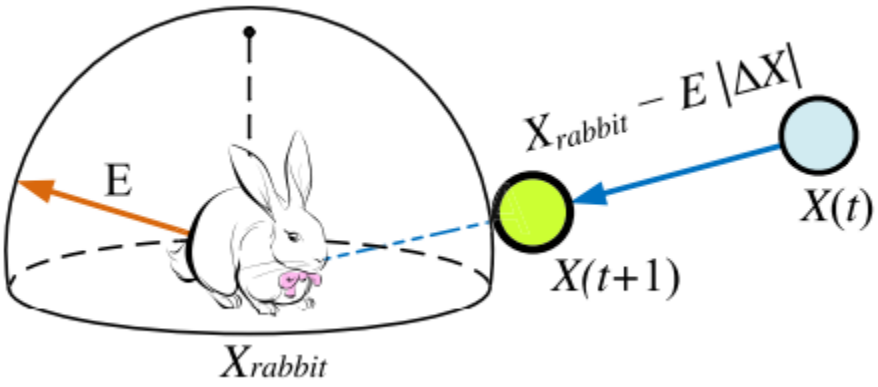
#### *5.5.3.2. Σκληρή πολιορκία*

Όταν $r \geq 0,5$ και $|E| < 0,5$, το θήραμα είναι τόσο εξαντλημένο και έχει χαμηλή ενέργεια διαφυγής. Επιπλέον, τα γεράκια Harris δύσκολα περικυκλώνουν το επιδιωκόμενο θήραμα για να εκτελέσουν τελικά την έκπληξη. Σε αυτήν την περίπτωση, οι τρέχουσες θέσεις ενημερώνονται χρησιμοποιώντας την Εξ. (2):

$$X(t + 1) = X_{rabbit}(t) - E\,|\Delta X(t)| \qquad (2)$$

Ένα απλό παράδειγμα αυτού του βήματος με ένα γεράκι απεικονίζεται στην εικόνα 10.

*Εικόνα 10. Παράδειγμα σκληρής πολιορκίας*




### *5.5.3.3. Ήπια πολιορκία με προοδευτικές γρήγορες καταδύσεις*

Όταν ακόμα |E| ≥0,5 αλλά r < 0,5, το κουνέλι έχει αρκετή ενέργεια για να δραπετεύσει με επιτυχία και εξακολουθεί να κατασκευάζεται μια ήπια πολιορκία πριν από την έκπληξη. Αυτή η διαδικασία είναι πιο έξυπνη από την προηγούμενη περίπτωση.

Εμπνευσμένος από πραγματικές συμπεριφορές γερακιών, ο αλγόριθμος HHO υποθέτει ότι τα γεράκια μπορούν σταδιακά να επιλέξουν την καλύτερη δυνατή κατάδυση προς το θήραμα, όταν θέλουν να πιάσουν το θήραμα στις ανταγωνιστικές καταστάσεις. Επομένως, για να εκτελεστεί μια ήπια πολιορκία, τα γεράκια μπορούν να αξιολογήσουν (αποφασίσουν) την επόμενη κίνησή τους με βάση τον ακόλουθο κανόνα στην Εξ. (3):

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X(t)| \qquad (3)$$

Στη συνέχεια, συγκρίνουν το πιθανό αποτέλεσμα μιας τέτοιας κίνησης με την προηγούμενη κατάδυση για να εντοπίσουν ότι θα είναι καλή κατάδυση ή όχι. Αν δεν ήταν λογικό (όταν βλέπουν ότι το θήραμα εκτελεί πιο παραπλανητικές κινήσεις), αρχίζουν επίσης να κάνουν ακανόνιστες, απότομες και γρήγορες βουτιές όταν πλησιάζουν το κουνέλι. Ο κανόνας με τον οποίο κάνουν βουτιές είναι ο εξής:

$$Z = Y + S \times LF(D) \qquad (4)$$

όπου D είναι η διάσταση του προβλήματος και S είναι ένα τυχαίο διάνυσμα κατά μέγεθος 1 × D και LF είναι η συνάρτηση Levy flight.

Ως εκ τούτου, η τελική στρατηγική για την ενημέρωση των θέσεων των γερακιών στη φάση ήπιας πολιορκίας μπορεί να υπολογιστεί από την Εξ. (5):

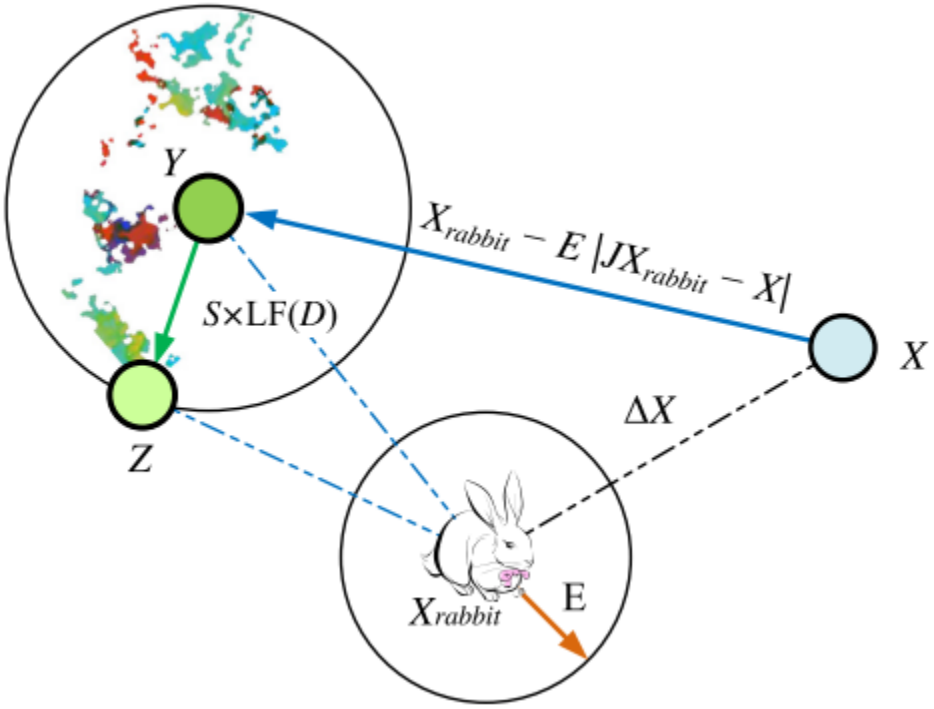
$$X(t+1) = \begin{cases} Y \text{ if } F(Y) < F(X(t)) \\ Z \text{ if } F(Z) < F(X(t)) \end{cases} \qquad (5)$$

όπου τα Y και Z λαμβάνονται χρησιμοποιώντας τις Εξ. (3) και (4).

Μια απλή απεικόνιση αυτού του βήματος για ένα γεράκι παρουσιάζεται στην εικόνα 11. Σημειώστε ότι το ιστορικό θέσης των μοτίβων κίνησης άλματος με βάση το LF κατά τη διάρκεια ορισμένων επαναλήψεων καταγράφεται και φαίνεται σε αυτήν την εικόνα. Οι έγχρωμες κουκκίδες είναι τα αποτυπώματα θέσης των

μοτίβων που βασίζονται σε LF σε μία δοκιμή και, στη συνέχεια, το HHO φτάνει στη θέση Z. Σε κάθε βήμα, μόνο η καλύτερη θέση Y ή Z θα επιλέγεται ως η επόμενη θέση. Αυτή η στρατηγική εφαρμόζεται σε όλους τους πράκτορες αναζήτησης.

*Εικόνα 11. Παράδειγμα ήπιας πολιορκίας με προοδευτικές γρήγορες καταδύσεις*




### *5.5.3.4. Σκληρή πολιορκία με προοδευτικές γρήγορες καταδύσεις*

Όταν |E| <0,5 και r < 0,5, το κουνέλι δεν έχει αρκετή ενέργεια για να ξεφύγει και μια σκληρή πολιορκία κατασκευάζεται πριν από την έκπληξη για να πιάσει και να σκοτώσει το θήραμα. Η κατάσταση αυτού του βήματος στην πλευρά του θηράματος είναι παρόμοια με εκείνη στην ήπια πολιορκία, αλλά αυτή τη φορά, τα γεράκια προσπαθούν να μειώσουν την απόσταση της μέσης θέσης τους με το θήραμα που διαφεύγει. Επομένως, ο ακόλουθος κανόνας εκτελείται σε κατάσταση σκληρής πολιορκίας:

$$X(t+1) = \begin{cases} Y \text{ if } F(Y) < F(X(t)) \\ Z \text{ if } F(Z) < F(X(t)) \end{cases}$$

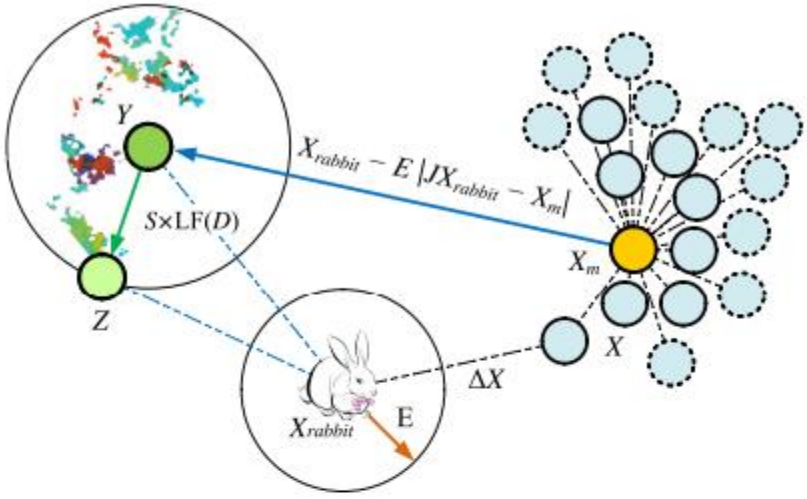
όπου το Y και το Z λαμβάνονται χρησιμοποιώντας νέους κανόνες:

$$Y = X_{rabbit}(t) - E\,|JX_{rabbit}(t) - X_m(t)|$$

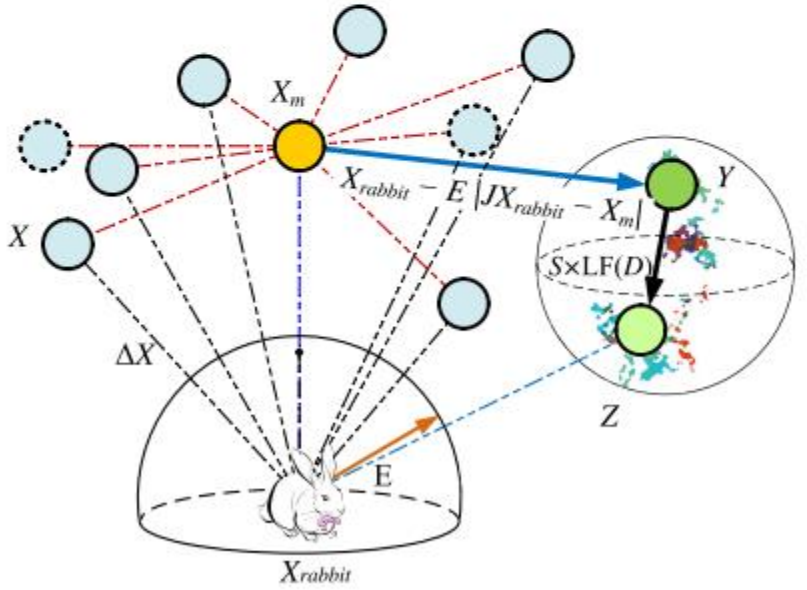
$$Z = Y + S \times LF(D)$$

Ένα απλό παράδειγμα αυτού του βήματος παρουσιάζεται στην εικόνα 12. Σημειώστε ότι οι έγχρωμες κουκκίδες είναι τα αποτυπώματα θέσης των μοτίβων που βασίζονται σε LF σε μία δοκιμή και μόνο το Y ή το Z θα είναι η επόμενη θέση για τη νέα επανάληψη.

*Εικόνα 12.Παράδειγμα συνολικών διανυσμάτων στην περίπτωση σκληρής πολιορκίας με προοδευτικές γρήγορες καταδύσεις σε 2D και 3D χώρο.*




(a) The process in 2D space




(b) The process in 3D space

Οπότε ο ψευδοκώδικας για τον αλγόριθμο Harris Hawks Optimization γίνεται ως εξής:

Αλγόριθμος HHO

Είσοδοι: Το μέγεθος του πληθυσμού N και ο μέγιστος αριθμός επαναλήψεων

Αποτελέσματα: Η θέση του κουνελιού(λύση) και η καταλληλότητα του

Αρχικοποίηση του τυχαίου πληθυσμού Xi(i = 1, 2, . . . , N)

Όσο (δεν ισχύει η συνθήκη διακοπής) επανάλαβε

Υπολόγισε τις τιμές καταλληλότητας των γερακιών

Όρισε το $X_{rabbit}$ ως τη θέση του κουνελιού (καλύτερη θέση)

Για (κάθε γεράκι ($X_i$)) επανάλαβε

Ενημέρωσε την αρχική ενέργεια $E_0$ και δύναμη άλματος J

Ενημέρωσε το E

Άν (|E|≥ 1) τότε ! Φάση εξερεύνησης

Ενημέρωσε το διάνυσμα τοποθεσίας

Τέλος_αν

Άν (|E|< 1) τότε ! Φάση εκμετάλλευσης

Άν (r ≥0,5 και |E|≥ 0,5 ) τότε ! Ήπια πολιορκία

Ενημέρωσε το διάνυσμα τοποθεσίας

Αλλιώς_αν (r ≥0,5 και |E|< 0,5 ) τότε ! Σκληρή πολιορκία

Ενημέρωσε το διάνυσμα τοποθεσίας

Αλλιώς_αν (r < 0,5 και |E|≥ 0,5 ) τότε ! Ήπια πολιορκία με προοδευτικές γρήγορες καταδύσεις

Ενημέρωσε το διάνυσμα τοποθεσίας

Αλλιώς_αν (r < 0,5 και |E|< 0,5 ) τότε ! Σκληρή πολιορκία με προοδευτικές γρήγορες καταδύσεις

Ενημέρωσε το διάνυσμα τοποθεσίας

Επιστροφή $X_{rabbit}$

---

Στη βιβλιογραφία φαίνεται πως ο αλγόριθμος HHO ήταν σε θέση να βρίσκει εξαιρετικές λύσεις σε σύγκριση με άλλους βελτιστοποιητές που θεωρούνται εδραιωμένοι στην εύρεση λύσεων.

# 6. Υλοποίηση

Στα πλαίσια αυτής της διπλωματικής εργασίας, υλοποιήθηκε ένα πρόγραμμα στη γλώσσα προγραμματισμού C++ για να λύσει το πρόβλημα του χρονοπρογραμματισμού εξετάσεων σε χρονοθυρίδες χωρίς χωρικούς περιορισμούς (U-UETP).

## 6.1 Δεδομένα Προβλήματος

Τα δεδομένα του προβλήματος που θα χρησιμοποιηθούν στα πειράματα είναι το σύνολο των 13 στιγμιότυπων προβλημάτων χρονοδιαγράμματος εξετάσεων πραγματικού κόσμου που οι Carter, Laporte

και Lee το 1996 παρουσίασαν και συγκέντρωσαν από 3 καναδικά γυμνάσια, 5 καναδικά, 1 της Αμερικής, 1 του Ηνωμένου Βασιλείου και 1 από πανεπιστήμιο της Μέσης Ανατολής.

Συγκεντρωτικά στοιχεία των προβλημάτων παρουσιάζονται στον Πίνακα 1 όπου εμφανίζεται και το πλήθος των διαθέσιμων περιόδων για κάθε πρόβλημα. Τα αρχεία δεδομένων (κατάληξη .stu) διαθέτουν για κάθε σπουδαστή μια γραμμή που περιέχει τους αριθμούς των μαθημάτων στα οποία είναι εγγεγραμμένος χωρισμένους μεταξύ τους με κενά. Η πρώτη γραμμή του αρχείου αντιστοιχεί στον πρώτο σπουδαστή, η δεύτερη γραμμή στο δεύτερο σπουδαστή κ.ο.κ.

Για παράδειγμα το αρχείο car-f-92.stu περιέχει 18419 σειρές δεδομένων και ξεκινά με τις ακόλουθες σειρές:

0170
0156
0281
0006
0154
0156
0383
0534
0535
0536
...

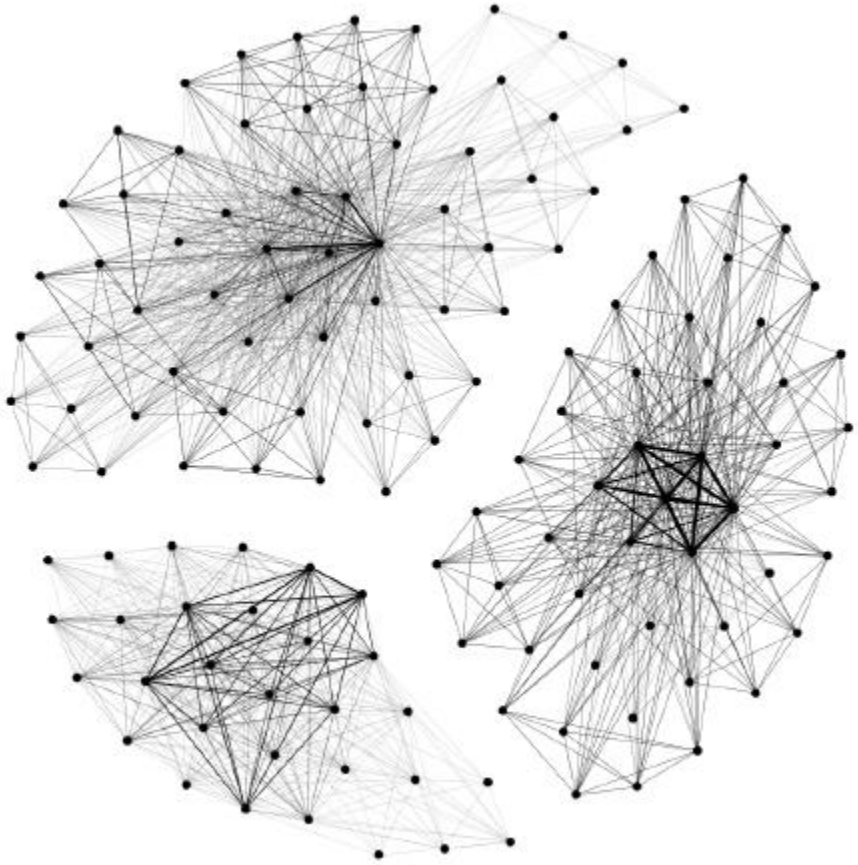
που σημαίνουν ότι ο φοιτητής 1 έχει εγγραφεί στο μάθημα 0170, ο φοιτητής 2 έχει εγγραφεί στο μάθημα 0156, ο φοιτητής 3 έχει εγγραφεί στο μάθημα 0281, ο φοιτητής 4 έχει εγγραφεί στο μάθημα 0006, ο φοιτητής 5 στα μαθήματα 0154 0156 κ.ο.κ.

*Πίνακας 4: Δεδομένα προβλημάτων*

| **Πρόβλημα** | **Αρχείο Δεδομένων** | **Εξετάσεις** | **Φοιτητές** | **Εγγραφές** | **Περίοδοι** | **Πυκνότητα** |
|---|---|---|---|---|---|---|
| car-f-92 | car-f-92.stu | 543 | 18419 | 55522 | 32 | 0.14 |
| car-s-91 | car-s-91.stu | 682 | 16925 | 56877 | 35 | 0.13 |
| ear-f-83 | ear-f-83.stu | 190 | 1125 | 8109 | 24 | 0.27 |
| hec-s-92 | hec-s-92.stu | 81 | 2823 | 10632 | 18 | 0.42 |
| kfu-s-93 | kfu-s-93.stu | 461 | 5349 | 25113 | 20 | 0.06 |
| lse-f-91 | lse-f-91.stu | 381 | 2726 | 10918 | 18 | 0.06 |
| pur-s-93 | pur-s-93.stu | 2419 | 30029 | 120681 | 42 | 0.03 |
| rye-s-93 | rye-s-93.stu | 486 | 11483 | 45051 | 23 | 0.07 |
| sta-f-83 | sta-f-83.stu | 139 | 611 | 5751 | 13 | 0.14 |
| tre-s-92 | tre-s-92.stu | 261 | 4360 | 14901 | 23 | 0.18 |
| uta-s-92 | uta-s-92.stu | 622 | 21266 | 58979 | 35 | 0.13 |
| ute-s-92 | ute-s-92.stu | 184 | 2749 | 11793 | 10 | 0.08 |
| yor-f-83 | yor-f-83.stu | 181 | 941 | 6034 | 21 | 0.29 |

Τα περισσότερα από τα σύνολα δεδομένων έχουν συνδεδεμένα στοιχεία, αλλά συνήθως υπάρχει ένα μεγάλο στοιχείο και μερικά πολύ μικρά. Το σύνολο δεδομένων sta-f-83 είναι μια αξιοσημείωτη εξαίρεση, καθώς χωρίζεται σε 3 συνδεδεμένα στοιχεία με 30, 47 και 62 εξετάσεις αντίστοιχα. Μια οπτικοποίηση τους φαίνεται στην εικόνα 13 που δείχνει άκρες με μεγαλύτερα βάρη (δηλαδή ζεύγη εξετάσεων με πιο κοινούς μαθητές) πιο σκούρες.

*Εικόνα 13. Συνδεδεμένα στοιχεία του συνόλου δεδομένων sta-f-83.*



Για τα προβλήματα αυτά έχουν υπολογιστεί τα κατώτερα όρια, αλλά είναι κοντά σε λύσεις που έχουν επιτευχθεί μόνο για περιπτώσεις όπου ο αριθμός των περιόδων είναι σχετικά μικρός και οι μαθητές είναι εγγεγραμμένοι σε πολλά μαθήματα. Για άλλες περιπτώσεις είναι υπολογιστικά δύσκολο να υπολογιστούν και πολύ αβέβαια για να έχουν πρακτική σημασία. Τα όρια που χρησιμοποιούν τον μέγιστο δυνατό αριθμό μαθητών εμφανίζονται στη στήλη «LB II» του πίνακα 5. Οι έντονες τιμές υποδεικνύουν ότι επιτεύχθηκε η βέλτιστη κατά τον υπολογισμό του ορίου από τον λύτη. Για την περίπτωση του pur-s-93, δεδομένου ότι δεν έχουμε τη βέλτιστη τιμή για τον μέγιστο αριθμό μαθητών που μπορούν να δώσουν εξετάσεις σε μία μόνο περίοδο, το όριο δεν υπολογίζεται. Η στήλη "LB II (dec.)" εμφανίζει την τιμή της στήλης «LB II» διαιρούμενη με τον αριθμό των μαθητών του αντίστοιχου συνόλου δεδομένων, που είναι ο τυπικός τρόπος αξιολόγησης του κόστους μιας λύσης στα σύνολα δεδομένων Carter.

*Πίνακας 5: Κάτω όριο λύσης των προβλημάτων*

| **Dataset** | **LB II** | **LB II (dec.)** |
|---|---|---|
| car-f-92-i | 145 | 0.0079 |
| car-s-91-i | 100 | 0.0059 |
| ear-f-83-i | **20542** | 18.2596 |
| hec-s-92-i | **10773** | 3.8162 |
| kfu-s-93-i | **30682** | 5.7360 |
| lse-f-91-i | **9147** | 3.3555 |
| pur-s-93-i | N/A | 0.0014 |
| rye-s-93-i | 43484 | 3.7868 |
| sta-f-83-i | **92900** | 152.0458 |
| tre-s-92-i | **3750** | 0.8601 |
| uta-s-92-i | 46 | 0.0022 |
| ute-s-92-i | **59398** | 21.5993 |
| yor-f-83-i | **18014** | 19.1435 |

## 6.2 Υπολογισμός λύσης

Αρχικά το πρόγραμμα υπολογίζει τα βασικά χαρακτηριστικά του προβλήματος όπως τον αριθμό εξετάσεων, τον αριθμό σπουδαστών, τον αριθμό εγγραφών σπουδαστών και την πυκνότητα συγκρούσεων. Για τον υπολογισμό της πυκνότητας έχει κατασκευαστεί ένας πίνακας συγκρούσεων. Ο πίνακας συγκρούσεων C είναι ένας δισδιάστατος πίνακας στον οποίο κάθε στοιχείο Cij = 1 αν η εξέταση i βρίσκεται σε σύγκρουση με την εξέταση j ενώ ισχύει ότι Cij = 0 σε άλλη περίπτωση. Η πυκνότητα συγκρούσεων υπολογίζεται διαιρώντας τον αριθμό των στοιχείων του πίνακα συγκρούσεων που έχουν την τιμή 1 με το συνολικό πλήθος των στοιχείων του πίνακα. Τα αποτελέσματα θα πρέπει να συμπίπτουν με τον πίνακα 4. Έτσι το κάθε στιγμιότυπο προβλήματος είναι φορτωμένο στο πρόγραμμα και είναι έτοιμο για επεξεργασία.

Οπότε τα χρώματα για το πρόβλημα των χρωματισμών κόμβων θα δημιουργήσει μια ενδεικτική λύση για το πρόβλημα του χρονοδιαγράμματος εξετάσεων. Για τον χρωματισμό των κόμβων υπάρχει πληθώρα αλγορίθμων που έχουν προταθεί. Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε ο αλγόριθμος DSatur.

Ο DSatur είναι ένας αλγόριθμος χρωματισμού γραφημάτων που προτάθηκε από τον Daniel Brélaz το 1979. Ομοίως με τον άπληστο αλγόριθμο χρωματισμού, το DSatur χρωματίζει τις κορυφές ενός γραφήματος το ένα μετά το άλλο, ξοδεύοντας ένα προηγουμένως αχρησιμοποίητο χρώμα όταν χρειάζεται. Μόλις χρωματιστεί μια νέα κορυφή, ο αλγόριθμος καθορίζει ποια από τις υπόλοιπες άχρωμες κορυφές έχει τον υψηλότερο αριθμό χρωμάτων στη γειτονιά του και χρωματίζει αυτήν την κορυφή στη συνέχεια. Ο Brélaz ορίζει αυτόν τον αριθμό ως τον βαθμό κορεσμού μιας δεδομένης κορυφής. Η συστολή του βαθμού κορεσμού σχηματίζει το όνομα του αλγορίθμου. Ο DSatur είναι ένας ευρετικός αλγόριθμος χρωματισμού

γραφημάτων, παράγει όμως ακριβή αποτελέσματα για διμερή γράφημα[1], κύκλο[2] και γραφήματα τροχών[3]. Ο DSatur έχει επίσης αναφερθεί ως κορεσμός LF στη βιβλιογραφία.

**Αλγόριθμος DSatur σε ψευδοκώδικα**

Ορίζεται ο βαθμός κορεσμού μιας κορυφής ως τον αριθμό των διαφορετικών χρωμάτων στη γειτονιά του. Δεδομένου ενός απλού, μη κατευθυνόμενου γραφήματος G που διαθέτει το σετ κορυφών V και το σετ άκρων E:

1. Δημιουργείται μια ταξινόμηση των βαθμών των κόμβων V.
2. Επιλέγεται μια κορυφή μέγιστου βαθμού και χρωματίζεται με το πρώτο χρώμα.
3. Εξετάζεται μια κορυφή με τον υψηλότερο βαθμό κορεσμού. Σε περίπτωση ισότητας κόμβων με τους υψηλότερους βαθμούς επιλέγεται ένας τυχαία.
4. Γίνεται πέρασμα στις τάξεις χρωμάτων που δημιουργήθηκαν μέχρι στιγμής και χρωματίζεται η επιλεγμένη κορυφή με το πρώτο κατάλληλο χρώμα.
5. Εκτός αν όλες οι κορυφές V είναι χρωματισμένες, επιστροφή στο βήμα 3.

Η χειρότερη περίπτωση πολυπλοκότητας του DSatur είναι **O(n^2)**, ωστόσο στην πράξη χρειάζονται κάποιοι επιπλέον υπολογισμοί που προκύπτουν από την ανάγκη συγκράτησης του βαθμού κορεσμού των άχρωμων κορυφών. Το DSatur έχει αποδειχθεί ότι είναι ακριβές για διμερή γραφήματα, καθώς και για κύκλους και γραφήματα τροχών. Σε μια εμπειρική σύγκριση του Lewis 2015, η DSatur παρήγαγε σημαντικά καλύτερα χρώματα κορυφής από τον άπληστο αλγόριθμο σε τυχαία γραφήματα με πιθανότητα ακμής p = 0,5 σε ποικίλο αριθμό κορυφών.

Με την εφαρμογή του DSatur αλγόριθμου στο γράφημα έχει γίνει μια πρώτη λύση για το πρόβλημα. Το κόστος όμως αυτής της λύσης είναι σχετικά μεγάλο. Έτσι το επόμενο βήμα είναι να γίνει μια βελτιστοποίηση της λύσης αυτής. Η τεχνική βελτιστοποίησης που επιλέχτηκε είναι ο αλγόριθμος Hill Climbing. Είναι ένας επαναληπτικός αλγόριθμος που ξεκινά με μια αυθαίρετη λύση σε ένα πρόβλημα, και στη συνέχεια προσπαθεί να βρει μια καλύτερη λύση κάνοντας μια σταδιακή αλλαγή στη λύση. Εάν η αλλαγή παράγει μια καλύτερη λύση, γίνεται μια άλλη σταδιακή αλλαγή στη νέα λύση, και ούτω καθεξής μέχρι να μην βρεθούν περαιτέρω βελτιώσεις. Ο αλγόριθμος Hill Climbing είναι ένας σχετικά απλός αλγόριθμος που σημαίνει δε χρειάζεται μεγάλο χρονικό διάστημα για να παράγει αποτελέσματα που να είναι αρκετά καλύτερα από την πρώτη λύση. Είναι ένας αλγόριθμος που μπορεί να επιστρέψει μια έγκυρη λύση σε ένα πρόβλημα ακόμα κι αν διακοπεί πριν από τη λήξη του. Ο αλγόριθμος αναμένεται να βρει καλύτερες και καλύτερες λύσεις όσο περισσότερο συνεχίζει να λειτουργεί. Στην υλοποίηση αυτή μπορούμε να θέσουμε πόσα βήματα θα κάνει ο αλγόριθμος θέτοντας έναν μέγιστο αριθμό ανταλλαγών των μαθημάτων χωρίς να έχει βρεθεί μια καλύτερη λύση.

Έχοντας τρέξει τον αλγόριθμο DSatur μέσω των δεδομένων του προβλήματος μας έχουμε πλέον στη διάθεση μας έναν αρχικό χρωματισμό για το γράφημα, ή με άλλα λόγια μια αρχική λύση. Οπότε γίνεται μετάβαση στο στάδιο της βελτιστοποίησης αυτής της αρχικής λύσης με τη χρήση ενός αλγόριθμου

[1] Ένα διμερές γράφημα είναι ένα γράφημα του οποίου οι κορυφές μπορούν να χωριστούν σε δύο χωριστά και ανεξάρτητα σύνολα U και V έτσι ώστε κάθε άκρη να συνδέει μια κορυφή στο U με ένα στο V.

[2] Ένα γράφημα κύκλου ή ένα κυκλικό γράφημα είναι ένα γράφημα που αποτελείται από έναν μόνο κύκλο, ή με άλλα λόγια, κάποιος αριθμός κορυφών (τουλάχιστον 3, εάν το γράφημα είναι απλό) συνδεδεμένο σε κλειστή αλυσίδα.

[3] Ένα γράφημα τροχών είναι ένα γράφημα που σχηματίζεται συνδέοντας μια μοναδική καθολική κορυφή σε όλες τις κορυφές ενός κύκλου

βελτιστοποίησης. Στην υλοποίηση χρησιμοποιήθηκε ο αλγόριθμος Hill Climbing κατά τον οποίο μπορούμε να επιτρέψουμε έναν μέγιστο αριθμό από βήματα (ή αλλιώς άλματα) εύρεσης μιας καλύτερης λύσης.

**Αλγόριθμος Hill Climbing σε ψευδοκώδικα (για το πρόβλημα μας)**

1. Υπολόγισε το κόστος της λύσης.
2. Επίλεξε δύο τυχαία μαθήματα.
3. Αντάλλαξε θέσεις στα δύο αυτά μαθήματα.
4. Υπολόγισε το νέο κόστος της λύσης.
5. Αν το νέο κόστος είναι μικρότερο από το προηγούμενο τότε η ανταλλαγή εγκρίνεται.
6. Αλλιώς αν το νέο κόστος είναι μεγαλύτερο από το προηγούμενο τότε η ανταλλαγή απορρίπτεται.
7. Αν έχουν γίνει ανταλλαγές όσες ο μέγιστος επιτρεπτός αριθμός χωρίς να γίνει αποδεκτή καλύτερη λύση, τότε η τελική λύση είναι η βέλτιστη λύση αυτής της στιγμής, διαφορετικά ο αλγόριθμος συνεχίζει από το βήμα 1.

Όπως είναι λογικό όσο μεγαλύτερος είναι ο επιτρεπτός αριθμός των ανταλλαγών που θα τρέξει ο αλγόριθμος Hill Climbing χωρίς να έχει βρεθεί κάποια βελτίωση, τόσο καλύτερα αποτελέσματα θα μας παράγει αλλά θα αυξηθεί σημαντικά και ο χρόνος που χρειάζεται για να παραχθούν. Στον πίνακα 6 φαίνονται τα αποτελέσματα για αντίστοιχα επιτρεπτά βήματα χωρίς βελτίωση της λύσης.

*Πίνακας 6: Αποτελέσματα κόστους/ανταλλαγές*

| **Problem** | **0 Hill Climbs** | **100 Hill Climbs** | **500 Hill Climbs** |
|---|---|---|---|
| car-f-92-i | 28.554536 | 6.960964 | 4.480591 |
| car-s-91-i | 30.897134 | 6.471846 | 5.397046 |
| ear-f-83-i | 150.232889 | 54.267953 | 39.619556 |
| hec-s-92-i | 56.209706 | 17.981257 | 11.411973 |
| kfu-s-93-i | 86.737334 | 25.592632 | 13.960292 |
| lse-f-91-i | 63.391049 | 19.565796 | 12.892883 |
| pur-s-93-i | 34.492058 | 11.779758 | 7.361950 |
| rye-s-93-i | 58.825917 | 16.176466 | 9.230341 |
| sta-f-83-i | 537.404255 | 202.327975 | 161.238953 |
| tre-s-92-i | 37.175688 | 14.196858 | 8.828532 |
| uta-s-92-i | 18.218753 | 6.233956 | 4.253832 |
| ute-s-92-i | 94.506366 | 32.472987 | 25.745726 |
| yor-f-83-i | 115.377258 | 50.164905 | 37.623804 |

Στον παρακάτω πίνακα (Πίνακας 7) γίνεται μια σύγκριση αποτελεσμάτων από την υλοποίηση που έγινε στα πλαίσια αυτής της διπλωματικής εργασίας με τα αποτελέσματα από διάφορες δημοσιεύσεις.

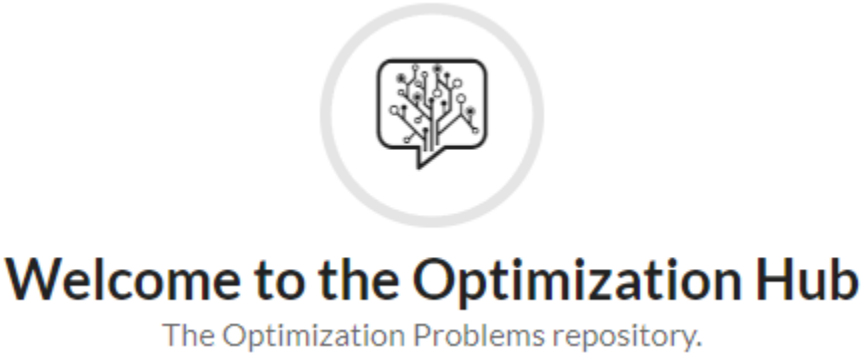
*Πίνακας 7: Σύγκριση αποτελεσμάτων*

| **Problem** | **Carter et al. 1996** | **Casey and Thompson 2003** | **Yang and Petrovic 2005** | **Burke et al. 2009** | **Caramia et al. 2008** | **OptimizationHub (Various Solvers)** | **My results** |
|---|---|---|---|---|---|---|---|
| car-f-92-i | 6.2 | 4.4 | 3.93 | 4.0 | 6.0 | 3.64 | 4.48 |
| car-s-91-i | 7.1 | 5.4 | 4.5 | 4.6 | 6.6 | 4.24 | 5.39 |
| ear-f-83-i | 36.4 | 34.8 | 33.7 | 32.8 | 29.3 | 32.42 | 39.61 |
| hec-s-92-i | 10.8 | 10.8 | 10.83 | 10.0 | 9.2 | 10.03 | 11.41 |
| kfu-s-93-i | 14.0 | 14.1 | 13.82 | 13.0 | 13.8 | 12.8 | 13.96 |
| lse-f-91-i | 10.5 | 14.7 | 10.35 | 10.0 | 9.6 | 9.77 | 12.89 |
| pur-s-93-i | 3.9 | - | - | - | - | 4 | 7.36 |
| rye-s-93-i | 7.9 | - | 8.53 | - | 6.8 | 7.84 | 9.23 |
| sta-f-83-i | 161.5 | 134.9 | 158.35 | 159.9 | 158.2 | 157.03 | 161.23 |
| tre-s-92-i | 9.6 | 8.7 | 7.92 | 7.9 | 9.4 | 7.59 | 8.82 |
| uta-s-92-i | 3.5 | - | 3.14 | 3.2 | 3.5 | 2.95 | 4.25 |
| ute-s-92-i | 25.8 | 25.4 | 25.39 | 24.8 | 24.4 | 24.76 | 25.74 |
| yor-f-83-i | 41.7 | 37.5 | 36.35 | 37.28 | 36.2 | 34.4 | 37.62 |

# 7. Optimization Hub – Διαγωνισμός ITC2007

Τα τελευταία χρόνια έχει δημιουργηθεί το “Optimization Hub”. Αυτό είναι ένα αποθετήριο περιπτώσεων προβλημάτων και λύσεων για προβλήματα βελτιστοποίησης. Ο κύριος σκοπός του αποθετηρίου είναι να παρακολουθεί πιστοποιημένες (δηλαδή, έγκυρες και χρονικά σφραγισμένες) λύσεις για προβλήματα βελτιστοποίησης μαζί με τους συντάκτες τους, προκειμένου να προωθήσει τη συγκρισιμότητα και την αναπαραγωγιμότητα των αποτελεσμάτων σε αυτόν τον ερευνητικό τομέα. Συμπληρώνει άλλα αποθετήρια προβλημάτων, όπως το OR-Library και το CSPLib, έτσι ώστε επίσης οι λύσεις προβλημάτων να διαχειρίζονται και να διατηρούνται στο σύστημα.

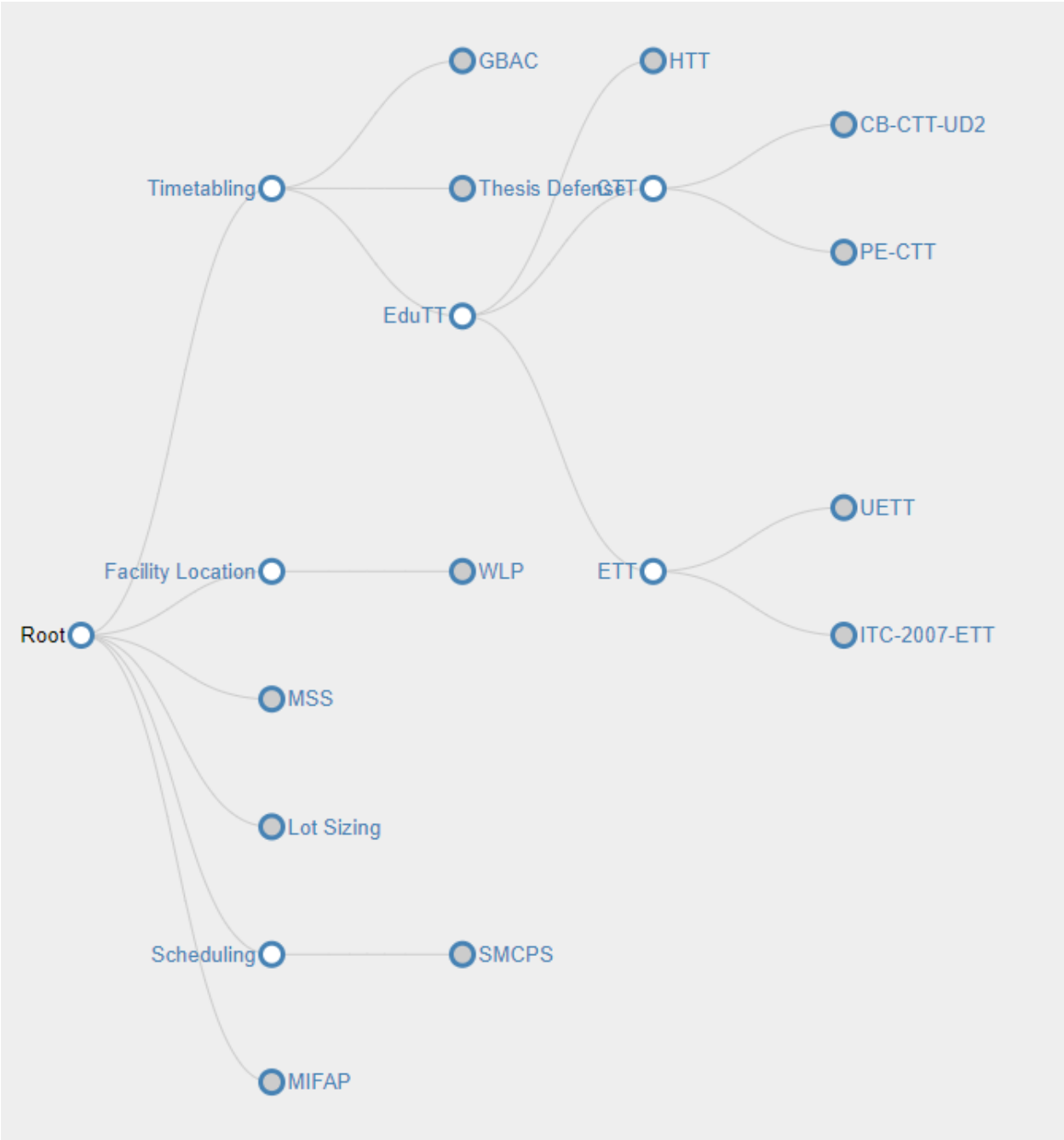
*Εικόνα 14. Αποθετήριο προβλημάτων και λύσεων*




Τα προβλήματα οργανώνονται σε μια ιεραρχική δομή δέντρου όπως φαίνεται στην Εικόνα 15, μέσα από την οποία ο κάθε ερευνητής μπορεί να βρει ένα πρόβλημα και το κόστος της βέλτιστης λύσης που υπάρχει μέχρι στιγμής. Έχει ως γενικό στόχο να προσελκύσει ερευνητές να αναπτύξουν και να δοκιμάσουν τεχνικές αιχμής σε ανταγωνιστικούς διαγωνισμούς. Στοχεύει επίσης να δημιουργήσει περαιτέρω ενδιαφέρον για την ερευνητική περιοχή παρέχοντας διάφορες διατυπώσεις των προβλημάτων χρονοδιαγράμματος που αντιμετωπίζονται στα εκπαιδευτικά ιδρύματα με βάση την προοπτική του «πραγματικού κόσμου». Ιδιαίτερη έμφαση δίνεται σε σενάρια «πραγματικού κόσμου» με στόχο την ενθάρρυνση της παραγωγής τεχνικών που έχουν τη δυνατότητα να λύσουν πρακτικές περιπτώσεις του προβλήματος. Ως εκ τούτου, ο ανταγωνισμός μπορεί να διαδραματίσει σημαντικό ρόλο στη γεφύρωση του σημερινού χάσματος που υπάρχει μεταξύ της έρευνας και της πρακτικής στον τομέα αυτό.

Μέσα σε αυτά τα προβλήματα είναι και το πρόβλημα UETT που περιεγράφηκε προηγουμένως. Αυτή είναι μια πολύ ουσιαστική έκδοση, η οποία επεκτείνει το βασικό πρόβλημα χρωματισμού γραφήματος, στο οποίο οι εξετάσεις είναι κόμβοι, οι περίοδοι είναι χρώματα και ο στόχος σχετίζεται με την απόσταση μεταξύ των περιόδων που έχουν ανατεθεί στις εξετάσεις σε σύγκρουση. Βέβαια με σκοπό να είναι πιο αντιπροσωπευτικό το πρόβλημα σε συνθήκες πραγματικού κόσμου, το πρόβλημα έχει δεχθεί μετατροπές που μπορούμε να χωρίσουμε σε τρεις παραλλαγές. Αυτές είναι το χρονοδιάγραμμα των εξετάσεων (Exam TT), το χρονοδιάγραμμα μαθημάτων μετά την εγγραφή (PostEnroll CTT) και το χρονοδιάγραμμα μαθημάτων βάσει προγράμματος σπουδών (Curriculum CTT). Αν και κάτω από τη γενική «ομπρέλα» του εκπαιδευτικού χρονοδιαγράμματος, αυτές οι τρεις προβληματικές παραλλαγές που εντοπίστηκαν έχουν σημαντικές διαφορές.

*Εικόνα 15. Δέντρο προβλημάτων βελτιστοποίησης*




Το Πρόβλημα του χρονοδιαγράμματος των εξετάσεων που εξετάζεται εδώ εισάγει μια πρακτική διατύπωση του προβλήματος που, πιστεύουν οι διοργανωτές του διαγωνισμού, προσθέτει σημαντικά στην τρέχουσα έρευνα και παρέχει μια σταθερή βάση για μελλοντικές προσπάθειες στο θέμα. Επιπλέον, το ενδιαφέρον που θα προκύψει ως μέρος αυτού του διαγωνισμού θα οδηγήσει στην ανάπτυξη, διερεύνηση και εφαρμογή μιας σειράς νέων και συναρπαστικών τεχνικών που δεν είχαν δοκιμαστεί προηγουμένως σε αυτόν τον σημαντικό τομέα αναζήτησης του πραγματικού κόσμου. Το μοντέλο προβλήματος μπορεί να περιγραφεί ως μετά την εγγραφή. Δηλαδή, οι φοιτητές που είναι εγγεγραμμένοι σε συγκεκριμένα μαθήματα που έχουν

συναφείς εξετάσεις θεωρείται ότι είναι εγγεγραμμένοι ή «δίνονται» σε αυτές τις εξετάσεις. Αν και άλλες προσεγγίσεις του προβλήματος λαμβάνονται εντός των ιδρυμάτων, αυτή είναι μακράν η πιο κοινή από πρακτική άποψη, καθώς και το πιο ευρέως αναφερόμενο μοντέλο του προβλήματος στην ακαδημαϊκή βιβλιογραφία.

Αυτό το συγκεκριμένο κομμάτι του διαγωνισμού προσθέτει σημαντικά στο ερευνητικό πεδίο με την εισαγωγή ενός πιο «πραγματικού» μοντέλου του προβλήματος όσον αφορά τα δεδομένα, τους περιορισμούς και την αξιολόγηση. Όλα τα σύνολα δεδομένων που χρησιμοποιούνται ως μέρος αυτού του διαγωνισμού προέρχονται από Ιδρύματα και έχουν ανωνυμοποιηθεί για σκοπούς διαγωνιστικής χρήσης.

## 7.1. Το μοντέλο του προβλήματος

Το μοντέλο προβλήματος χρονοδιαγράμματος εξέτασης που παρουσιάζεται εδώ επεκτείνει το τρέχον μοντέλο του προβλήματος που συνήθως αντιμετωπίζεται. Το θεμελιώδες πρόβλημα περιλαμβάνει τον προγραμματισμό των εξετάσεων σε έναν αριθμό περιόδων εντός μιας καθορισμένης συνεδρίας εξέτασης, ενώ ικανοποιούνται ορισμένοι αυστηροί περιορισμοί. Όπως και άλλοι τομείς του χρονοδιαγράμματος, μια εφικτή λύση είναι αυτή στην οποία πληρούνται όλοι οι αυστηροί περιορισμοί. Η ποιότητα της λύσης μετριέται με βάση την ικανοποίηση χαλαρών περιορισμών.

Παρέχονται νέες και πρόσθετες πληροφορίες σχετικά με τους περιορισμούς (αυστηρούς και χαλαρούς), τους πόρους και τη συνεδρία εξέτασης. Για παράδειγμα, όσον αφορά τους αυστηρούς περιορισμούς, παρέχονται αριθμοί και μεγέθη δωματίων. Επιπλέον, παρουσιάζονται επίσης πληροφορίες για τη δομή, τη διάρκεια και τον αριθμό των μεμονωμένων περιόδων. Όσον αφορά τους χαλαρούς περιορισμούς, παρέχονται πολύ πιο πρακτικές πληροφορίες όσον αφορά τον τρόπο με τον οποίο ένας οργανισμός μετρά τη συνολική ποιότητα μιας λύσης.

### 7.1.1 Περιγραφή Προβλήματος

Το πρόβλημα συνίσταται από τους εξής κανόνες:

• Μια συνεδρία εξέτασης αποτελείται από έναν αριθμό περιόδων για ένα καθορισμένο χρονικό διάστημα. Παρέχεται ο αριθμός και η διάρκεια των Περιόδων.

• Ένα σύνολο εξετάσεων που θα προγραμματιστούν σε περιόδους.

• Ένα σύνολο μαθητών που εγγράφηκαν σε ατομικές εξετάσεις. Κάθε μαθητής εγγράφεται σε έναν αριθμό εξετάσεων. Οι μαθητές που εγγράφονται σε μια εξέταση θεωρείται ότι «δέχονται» αυτήν την εξέταση. Για κάθε εξέταση παρέχεται το σύνολο των εγγεγραμμένων μαθητών.

• Παρέχεται ένα σύνολο δωματίων με ατομικές χωρητικότητες.

• Αυστηροί περιορισμοί που πρέπει να ικανοποιηθούν.

• Χαλαροί περιορισμοί που συμβάλλουν στην επιβολή κυρώσεων εάν παραβιαστούν.

• Λεπτομέρειες που υπολογίζουν το βάρος (σημαντικότητα) συγκεκριμένων χαλαρών περιορισμών.

Ένα εφικτό χρονοδιάγραμμα (λύση) είναι αυτό στο οποίο όλες οι εξετάσεις έχουν ανατεθεί σε περίοδο και αίθουσα και πληρούνται όλοι οι ακόλουθοι αυστηροί περιορισμοί:

• Κανένας μαθητής δεν δίνει περισσότερες από μία εξετάσεις ταυτόχρονα.

• Δεν γίνεται υπέρβαση της χωρητικότητας των μεμονωμένων αιθουσών σε καμία στιγμή κατά τη διάρκεια της συνεδρίας εξέτασης.

• Δεν παραβιάζονται οι διάρκειες περιόδου.

• Ικανοποίηση αυστηρών περιορισμών που σχετίζονται με την περίοδο π.χ. Εξέταση Α μετά την Εξέταση Β.

• Ικανοποίηση αυστηρών περιορισμών που σχετίζονται με το δωμάτιο π.χ. Η Εξέταση Α πρέπει να χρησιμοποιήσει την Αίθουσα 101.

Οι χαλαροί περιορισμοί μπορούν να περιγράφουν ως εξής.

• Δύο διαδοχικές εξετάσεις

Ο αριθμός των περιστατικών όπου οι μαθητές πρέπει να δώσουν δύο διαδοχικές εξετάσεις την ίδια μέρα.

• Δύο εξετάσεις την ημέρα

Ο αριθμός των περιστατικών όπου οι μαθητές πρέπει να δώσουν δύο εξετάσεις την ίδια μέρα. Αυτός ο περιορισμός γίνεται σημαντικός μόνο όταν υπάρχουν περισσότερες από δύο εξεταστικές περίοδοι τη μία ημέρα.

• Καθορισμένη κατανομή των εξετάσεων.

Ο αριθμός των περιστατικών όταν οι μαθητές πρέπει να δώσουν περισσότερες από μία εξετάσεις σε μια χρονική περίοδο που καθορίζεται από το ίδρυμα. Αυτό χρησιμοποιείται συχνά σε μια προσπάθεια να είναι όσο το δυνατόν πιο δίκαιος με όλους τους μαθητές που δίνουν εξετάσεις.

• Μικτή διάρκεια εξετάσεων σε επιμέρους περιόδους.

Ο αριθμός των εμφανίσεων των εξετάσεων που έχουν προγραμματιστεί σε αίθουσες μαζί με άλλες εξετάσεις διαφορετικής χρονικής διάρκειας.

• Μεγαλύτερες εξετάσεις εμφανίζονται αργότερα στο χρονοδιάγραμμα

Ο αριθμός των «μεγάλων» εξετάσεων που εμφανίζονται στο «τελευταίο τμήμα» του χρονοδιαγράμματος. Τόσο το "μεγάλο" και το "μεταγενέστερο τμήμα" ορίζονται από τον χρήστη.

• Χαλαροί περιορισμοί που σχετίζονται με την περίοδο

Ο αριθμός των φορών που χρησιμοποιείται μια περίοδος που έχει μια σχετική ποινή. Αυτό πολλαπλασιάζεται με την πραγματική ποινή, καθώς διαφορετικές περίοδοι μπορεί να έχουν διαφορετικούς συναφείς συντελεστές στάθμισης βάρους.

• Χαλαροί περιορισμοί που σχετίζονται με το δωμάτιο.

Ο αριθμός των φορών που χρησιμοποιείται ένα δωμάτιο που έχει μια σχετική ποινή. Αυτό πολλαπλασιάζεται με την πραγματική ποινή, καθώς διαφορετικά δωμάτια μπορεί να έχουν διαφορετικές σχετικές σταθμίσεις βάρους.

Τα ιδρύματα μπορεί να σταθμίζουν αυτούς τους χαλαρούς περιορισμούς διαφορετικά μεταξύ τους σε μια προσπάθεια να παράγουν μια λύση που είναι κατάλληλη για τις ιδιαίτερες ανάγκες τους. Αυτή είναι μια σχετική στάθμιση των χαλαρών περιορισμών που παρέχει αποτελεσματικά ένα ποιοτικό μέτρο της λύσης που θα κατασκευαστεί.

Πρέπει να σημειωθεί ότι κατά τη διαμόρφωση μιας λύσης, είναι σύνηθες για ένα ίδρυμα να αλλάζει διάφορες ρυθμίσεις χαλαρών περιορισμών σε μια προσπάθεια να παράγει λύσεις που κρίνουν ικανοποιητικές για όλους τους τελικούς χρήστες. Πράγματι, αυτός είναι ο λόγος για τον οποίο παρέχονται οι σταθμίσεις βαρών χαλαρών περιορισμών στα δεδομένα σε αντίθεση με τον ορισμό του προβλήματος.

### 7.1.2 Αξιολόγηση λύσης

Στα πλαίσια του διαγωνισμού, η ποιότητα ενός χρονοδιαγράμματος (λύσης) αντικατοπτρίζεται από δύο τιμές: τον αριθμό των παραβιάσεων αυστηρών περιορισμών (Απόσταση έως τη σκοπιμότητα) και το άθροισμα σταθμισμένων βαρών των παραβιάσεων χαλαρών περιορισμών. Για να συγκριθούν δύο λύσεις, πρώτα εξετάζεται η Απόσταση έως τη σκοπιμότητα και η λύση με τη χαμηλότερη τιμή για αυτό θα είναι ο νικητής. Εάν οι δύο λύσεις είναι ισοδύναμες, θα εξετάζεται ο αριθμό των παραβιάσεων των χαλαρών περιορισμών. Νικητής θα είναι η λύση που έχει τη χαμηλότερη αξία εδώ.

Η απόσταση από τη σκοπιμότητα είναι το σύνολο των παρακάτω αριθμών:

- Συγκρούσεις: Ο αριθμός των εμφανίσεων αντικρουόμενων εξετάσεων την ίδια περίοδο.
- Κατοχή δωματίου: Ο αριθμός των περισσότερων θέσεων που απαιτούνται σε κάθε μεμονωμένη περίοδο από αυτή που είναι διαθέσιμη.
- Χρησιμοποίηση περιόδου: Ο αριθμός των περιστατικών όταν απαιτείται περισσότερος χρόνος σε κάθε μεμονωμένη περίοδο από αυτόν που είναι διαθέσιμος.
- Σχετικά με την περίοδο: Ο αριθμός των περιστατικών όταν δεν τηρούνται οι απαιτήσεις παραγγελίας.
- Σχετικά με το δωμάτιο: Ο αριθμός των περιστατικών που δεν τηρήθηκαν οι απαιτήσεις του δωματίου.

Το προκύπτουν σύστημα είναι μια ειδική περίπτωση «ιεραρχιών περιορισμών». Στην περίπτωσή μας, έχουμε ουσιαστικά τρία επίπεδα περιορισμών

1. απαιτείται: περιορισμοί που δεν μπορούν να παραβιαστούν (π.χ. ο περιορισμός ότι οι εξετάσεις δεν χωρίζονται μεταξύ των αιθουσών)
2. αυστηροί: περιορισμοί των οποίων η παραβίαση οδηγεί σε μη μηδενική «απόσταση από τη σκοπιμότητα» (η οποία ουσιαστικά είναι απλώς η αντικειμενική συνάρτηση που σχετίζεται με αυτό το επίπεδο). Η σκοπιμότητα αντιστοιχεί στην ικανοποίηση όλων των σκληρών και απαιτούμενων περιορισμών.
3. χαλαροί: τα συνήθη σύνολα περιορισμών που προτιμούμε να ικανοποιούμε αλλά αναμένουμε ότι δεν θα είναι δυνατό να ικανοποιηθούν όλοι.

Εδώ φαίνεται πως έχει έννοια η βελτίωση της λύσης σε οποιοδήποτε επίπεδο έχει προτεραιότητα έναντι της βελτίωσής της σε χαμηλότερα επίπεδα.

## 7.2 Περιγραφή χαλαρών περιορισμών

Οι χαλαροί περιορισμοί που έχουν προταθεί, αθροίζονται σε έναν δείκτη που δείχνει την καταλληλόλητα κάθε λύσης, τον «Δείκτη Ιδρυτικού Μοντέλου».

### 7.2.1 Δύο Διαδοχικές Εξετάσεις

Αυτός ο υπολογισμός λαμβάνει υπόψη τον αριθμό των περιστατικών όπου οι μαθητές δίνουν δύο εξετάσεις η μία μετά την άλλη, δηλ. διαδοχικά. Αφού διαπιστωθεί αυτό, ο αριθμός των μαθητών αθροίζεται και

πολλαπλασιάζεται με τον αριθμό που παρέχεται στη στάθμιση «δύο στη σειρά» στον «Δείκτη Ιδρυτικού Μοντέλου». Σημειώστε ότι δύο συνεχόμενες εξετάσεις δεν υπολογίζονται σε μια νύχτα π.χ. εάν ένας μαθητής έχει εξετάσεις την τελευταία περίοδο της μιας ημέρας και μια άλλη την πρώτη περίοδο την επόμενη μέρα, αυτό δεν μετράει ως δύο στη σειρά.

### 7.2.2 Δύο Εξετάσεις την Ημέρα

Στην περίπτωση που υπάρχουν τρεις ή περισσότερες περίοδοι σε μια ημέρα, υπολογίζεται ο αριθμός των περιπτώσεων μαθητών που έχουν δύο εξετάσεις σε μια ημέρα που δεν γειτνιάζουν άμεσα, δηλαδή δεν είναι διαδοχικές. Ο συνολικός αριθμός πολλαπλασιάζεται στη συνέχεια με τη στάθμιση «δύο σε μια ημέρα» που παρέχεται στον «Δείκτη Ιδρυτικού Μοντέλου». Ως εκ τούτου, δύο εξετάσεις την ημέρα θεωρούνται αυτές που δεν γειτνιάζουν, δηλαδή έχουν ελεύθερη περίοδο μεταξύ τους. Αυτό γίνεται για να διασφαλιστεί ότι η τοποθέτηση μιας συγκεκριμένης εξέτασης σε μια λύση δεν συμβάλλει δύο φορές στη συνολική ποινή. Για παράδειγμα, εάν η Εξέταση Α και η Εξέταση Β βρίσκονταν σε παρακείμενες περιόδους την ίδια ημέρα, η ποινή θα υπολογίζεται ως μέρος της ποινής «Δύο εξετάσεις στη σειρά». Σημειώνεται ότι όπου η εξεταστική συνεδρία περιέχει ημέρες με 2 περιόδους, αυτό το στοιχείο της ποινής, αν και υπάρχει για συνέχεια, καθίσταται περιττό. Όταν συμβαίνει αυτό, αυτό το τμήμα της ποινής θα είναι πάντα ίσο με μηδέν.

### 7.2.3 Διασπορά περιόδου

Αυτός ο περιορισμός επιτρέπει σε έναν οργανισμό να «διαδώσει» τις εξετάσεις ενός ατόμου σε έναν καθορισμένο αριθμό περιόδων. Αυτό μπορεί να θεωρηθεί ως επέκταση των δύο περιορισμών που περιεγράφηκαν προηγουμένως. Στο πλαίσιο του «Δείκτη Ιδρυτικού μοντέλου», παρέχεται ένας αριθμός που σχετίζεται με πόσες περιόδους θα πρέπει να «βελτιστοποιηθεί» η λύση. Όσο μεγαλύτερος είναι αυτός ο αριθμός, τόσο καλύτερη είναι πιθανώς η διάδοση των εξετάσεων για μεμονωμένους μαθητές. Σε πολλά ιδρύματα, η κατασκευή λύσεων με ταυτόχρονη αλλαγή αυτής της ρύθμισης έχει οδηγήσει σε χρονοδιαγράμματα με τα οποία το ίδρυμα είναι πολύ πιο ικανοποιημένο.

### 7.2.4 Μικτές Διάρκειες

Αυτό επιβάλλει ποινή σε Αίθουσα και Περίοδο (όχι Εξέταση) όπου υπάρχουν μικτές διάρκειες. Η πρόθεση εδώ είναι να προσπαθήσουμε να διασφαλίσουμε ότι οι εξετάσεις θα πραγματοποιούνται μαζί, οι οποίες είναι ίσης διάρκειας. Κατά τον υπολογισμό αυτού του τμήματος της ποινής, η συνιστώσα μεικτής διάρκειας του «Δείκτη Ιδρυτικού Μοντέλου» υπολογίζεται από τον αριθμό των παραβιάσεων που εντοπίστηκαν.

### 7.2.5 Μεγαλύτερες Εξετάσεις στην έναρξη της εξεταστικής

Είναι επιθυμητό οι εξετάσεις με τον μεγαλύτερο αριθμό μαθητών να προγραμματίζονται στην αρχή της εξεταστικής. Αυτό επιτρέπει στο Ίδρυμα να προσπαθήσει να διασφαλίσει ότι μεγαλύτερες εξετάσεις πραγματοποιούνται νωρίτερα κατά τη διάρκεια της εξεταστικής περιόδου. Αυτό είναι δημοφιλές στην πράξη, καθώς οι εξετάσεις με περισσότερους εγγεγραμμένους μαθητές χρειάζονται περισσότερο χρόνο για να βαθμολογηθούν.

### 7.2.6 Ποινή δωματίου

Συχνά οι οργανισμοί θέλουν να περιορίσουν στο ελάχιστο τη χρήση ορισμένων δωματίων. Όπως και με τη συνιστώσα «Μικτές Διάρκειες» της συνολικής ποινής, αυτό το μέρος της συνολικής ποινής θα πρέπει να υπολογίζεται ανά περίοδο. Για κάθε περίοδο, εάν ένα δωμάτιο που χρησιμοποιείται στη λύση έχει μια σχετική ποινή, η ποινή για αυτό το δωμάτιο για αυτήν την περίοδο υπολογίζεται πολλαπλασιάζοντας τη σχετική ποινή επί τον αριθμό των φορών που χρησιμοποιείται το δωμάτιο.

### 7.2.7 Ποινή περιόδου

Συχνά συμβαίνει ότι οι οργανισμοί θέλουν να περιορίσουν στο ελάχιστο τη χρήση ορισμένων περιόδων. Όπως και με τις συνιστώσες «Μικτές Διάρκειες» και «Ποινή δωματίου» της συνολικής ποινής, αυτό το

μέρος της συνολικής ποινής θα πρέπει να υπολογίζεται ανά περίοδο. Για κάθε περίοδο, η ποινή υπολογίζεται πολλαπλασιάζοντας τη σχετική ποινή επί τον αριθμό των φορών των εξετάσεων που έχουν προγραμματιστεί εντός αυτής της περιόδου.

### 7.3 Συμπεράσματα διαγωνισμού

Η διατύπωση του προβλήματος του χρονοδιαγράμματος των εξετάσεων που είναι κοινό σε πολλά ιδρύματα. Αυτό το κομμάτι εισάγει μια πρακτική διατύπωση του προβλήματος που, πιστεύουν οι διοργανωτές, προσθέτει σημαντικά στην τρέχουσα έρευνα και παρέχει μια σταθερή βάση για μελλοντικές προσπάθειες στην περιοχή. Αν και μια συνάρτηση αξιολόγησης «σταθμισμένου αθροίσματος» δεν είναι ιδανική π.χ. μπορεί να έχει δυσμενείς παρενέργειες για ορισμένους μεμονωμένους μαθητές, είναι η μέθοδος που επιλέγεται εδώ λόγω της ευκολίας εφαρμογής για λόγους σύγκρισης. Ελπίζεται ότι το ενδιαφέρον που θα προκύψει από τις προσπάθειες εδώ θα οδηγήσει σε πραγματική πολυσκοπική αξιολόγηση πιθανών λύσεων. Συγκεκριμένα, αποφασίστηκε τα βάρη να είναι συμπεριλαμβανόμενα στην ίδια τη μορφή δεδομένων αντί να πρέπει οι λύτες να τα κωδικοποιήσουν στη λύση τους. Αυτό τουλάχιστον θα έπρεπε να επιτρέπει εύκολα παραλλαγές των βαρών έτσι ώστε να διερευνηθούν ιδιότητες πολλαπλών στόχων. Επίσης, είναι απίθανο κάθε ίδρυμα να έχει τα ίδια βάρη, και έτσι ο καθορισμός τους στον λύτη φαίνεται ακατάλληλος.

## 6. Συμπεράσματα

Μπορεί να φανεί ότι μια απλή προσέγγιση όπως αυτή που υλοποιήθηκε μπορεί να παράγει ισχυρά αποτελέσματα. Η λύση του προβλήματος χρονοπρογραμματισμού εξετάσεων αποτελείται από δύο μέρη, έναν αλγόριθμο εύρεσης μιας αρχικής λύσης και έπειτα ένας αλγόριθμος βελτιστοποίησης ώστε να παράγεται καλύτερη λύση σε κάθε βήμα του. Για την υλοποίηση επιλέχθηκε ο αλγόριθμος χρωματισμού DSatur και μια εκδοχή του αλγορίθμου Hill Climbing, αλλά οποιοιδήποτε αλγόριθμοι μπορούν να χρησιμοποιηθούν για κάθε μέρος.

Επίσης, η υλοποίηση που έγινε μπορεί να εφαρμοστεί και για άλλα προβλήματα χρονοπρογραμματισμού όπως χρονοδιάγραμμα μαθημάτων (Babaeiet al. 2015), κατάλογος εργασίας νοσοκόμων (Cheang et al. 2003), προγραμματισμός εργασιακών καταστημάτων (Chaundry και Khan 2016) και άλλες εκδόσεις του προβλήματος χρονοδιαγράμματος εξετάσεων όπως ένα που ιδρύθηκε στον Διεθνή Διαγωνισμό Χρονοδιαγράμματος (ITC) 2007 Track 1 (Di Gaspero et al. 2007)

Οι τεχνικές που αναφέρθηκαν είναι δυνατόν να εφαρμοστούν για όλα τα προβλήματα που μπορούν να μοντελοποιηθούν με τον ίδιο τρόπο, δηλαδή να «μεταφραστούν» σε πρόβλημα χρωματισμού γραφήματος. Αφού γίνει η μοντελοποίηση το μόνο που μένει είναι η επιλογή των αλγορίθμων για τη δημιουργία της αρχικής λύσης και για το στάδιο της βελτιστοποίησης από μια πληθώρα αλγορίθμων, μερικοί από τους οποίους περιεγράφηκαν προηγουμένως.

Η μοντελοποίηση της πολυπλοκότητας των προβλημάτων χρονοδιαγράμματος συνεχίζει να αντιπροσωπεύει σημαντικά ζητήματα στον ερευνητικό τομέα του χρονοδιαγράμματος. Ωστόσο, αυτά τα ζητήματα δεν έχουν συζητηθεί ευρέως τα τελευταία χρόνια και δεν υπάρχουν ακόμη καθολικά ολοκληρωμένα μοντέλα. Η πολυπλοκότητα αυτών των προβλημάτων μελετήθηκε επίσης, δείχνοντας ορισμένες παραλλαγές του προβλήματος ως NP-complete. Απαιτείται περαιτέρω εργασία για την αντιμετώπιση αυτών και των συναφών ζητημάτων για να παρασχεθεί θεμελιώδης υποστήριξη για την καλύτερη κατανόηση και ανάπτυξη της έρευνας σχετικά με το χρονοδιάγραμμα των εξετάσεων.

Υπάρχει επίσης έρευνα στη βιβλιογραφία σχετικά με τη δημιουργία γλωσσών και εργαλείων γενικού χρονοδιαγράμματος σε μια προσπάθεια να μοντελοποιηθούν οι πραγματικές περιπτώσεις του προβλήματος. Οι Tsang, Mills και Williams ανέπτυξαν μια γλώσσα υψηλού επιπέδου για να προσδιορίσουν τα προβλήματα χρονοδιαγράμματος των εξετάσεων ως προβλήματα ικανοποίησης περιορισμών. Οι Di Gaspero και Schaerf κατασκεύασαν ένα εργαλείο λογισμικού που ονομάζεται EASYLOCAL++ για εύκολη εφαρμογή αλγορίθμων τοπικής αναζήτησης σε γενικά προβλήματα χρονοδιαγράμματος. Ένα γενικό και επαναχρησιμοποιήσιμο πλαίσιο θα βελτιώσει περαιτέρω την τρέχουσα ανάπτυξη και θα δικαιολογήσει εύκολες επιστημονικές συγκρίσεις.

Τα τελευταία χρόνια έχει γίνει σημαντικός όγκος έρευνας που εξετάζει πτυχές τόσο από τη θεωρία όσο και από την πράξη. Οι meta-heuristic αλγόριθμοι (π.χ. Tabu Search, Simulated Annealing, Genetic algorithms, κ.λπ.) αντιπροσωπεύουν τις πιο αποτελεσματικές προσεγγίσεις τελευταίας τεχνολογίας σε τυπικά σημεία αναφοράς. Υπάρχει επίσης μεγάλος όγκος εργασίας όπου μελετώνται οι υβριδισμοί μεταξύ meta-heuristic. Αυτό περιλαμβάνει την αποτελεσματική ενσωμάτωση των πρώιμων τεχνικών χρονοδιαγράμματος, όπως heuristic αλγόριθμοι χρωματισμού γραφημάτων και οι τεχνικές που βασίζονται σε περιορισμούς. Μαζί με αυτά τα κύρια θέματα έρευνας, υπάρχει επίσης μια σειρά από νέες τάσεις, συμπεριλαμβανομένου του πιο αποτελεσματικού σχεδιασμού δομών γειτονιάς (δηλαδή αναζήτηση μεταβλητής γειτονιάς για χρονοδιάγραμμα, κ.λπ.). Η ευελιξία της αναζήτησης βελτιώνεται έτσι για την αντιμετώπιση πιο περίπλοκων προβλημάτων με ένα ευρύτερο φάσμα περιορισμών στο χρονοδιάγραμμα των εξετάσεων. Ορισμένες έρευνες με κίνητρο τον στόχο της αύξησης της γενικότητας των προσεγγίσεων χρονοδιαγράμματος έχουν επίσης ελπιδοφόρα αποτελέσματα, καθώς οι hyper-heuristic αλγόριθμοι είναι ένας από τους τομείς που προσελκύουν μεγάλη ερευνητική προσοχή.

# 7. Βιβλιογραφία


1. Babaei H, Karimpour J, Hadidi A (2015) A survey of approaches for university course timetabling problem. Comput Ind Eng.
2. Burke, E. K., Bykov, Y., Newall, J., & Petrovic, S. (2004). A time-predefined local search approach to exam timetabling problems. *IIE Transactions*.
3. Burke, E. K., & Kendall, G. e. (2005). *Search methodologies: introductory tutorials in optimization and decision support techniques*, Springer: New York.
4. Burke, E. K., & Bykov, Y. (2008). An adaptive flex-deluge approach to university exam timetabling. Submitted to *INFORMS Journal of Computing.*
5. Burke, E. K., Eckersley, A. J., McCollumn, B., Petrovic, S., & Qu, R. (2009). Hybrid variable neighborhood approaches to university exam timetabling. *European Journal of Operational Research*.
6. Caramia, M., Dell'Olmo, P., & Italiano, G. (2008). Novel local search-based approaches to university examination timetabling. *INFORMS Journal of Computing*.
7. Carter, M. W., Laporte, G., & Lee, S. (1996). Examination timetabling: algorithmic strategies and applications. *Journal of Operational Research Society*.
8. Casey, S., & Thompson, J. (2003). GRASPing the examination scheduling problem. *PATAT IV, Selected Revised Papers,* Springer LNCS.
9. Chaudhry IA, Khan AA (2016) A research survey: review of flexible job shop scheduling techniques.
10. Cheang B, Li H, Lim A, Rodrigues B (2003) Nurse rostering problems—a bibliographic survey.
11. Di Gaspero L, McCollum B, Schaerf A (2007) The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Tech. rep., Citeseer
12. Dueck G. (1993). New optimization heuristics. The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*.
13. Glover, F., & Kochenberger, G.A. (2003). *Handbook of metaheuristics*. Kluwer Academic Publishers: Dordrecht.
14. Hu, T. C., Kahng, A. B, & Tsao, C.-W. A. (1995). Old bachelor acceptance: a new class of nonmonotone threshold accepting methods. *ORSA Journal on Computing*.
15. Laguna, M., & Glover, F. (1996). What is tabu search? *Colorado Business Review*.
16. Mohammed Azmi Al-Betar (2020). A β-hill climbing optimizer for examination timetabling problem. *Journal of Ambient Intelligence and Humanized Computing*.
17. Paula Amaral, Tiago Cardal Pais. (2016). Compromise ratio with weighting functions in a Tabu Search multi-criteria approach to examination timetabling. *Department of Mathematics and CMA, Universidade Nova de Lisboa, Portugal b School of Computer Science, University of Nottingham, Nottingham, UK.*
18. Qu, R., Burke, E. K., McCollumn, B., Merlot, L. T. G., & Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of Scheduling*.
19. Taillard, E. D., Gambardella, L. M., Gendreau, M., & Potvin J.-Y. (2001). Adaptive memory programming: a unified view of metaheuristics. *European Journal of Operational Research*.
20. Yang, Y., & Petrovic, S. (2005). A novel similarity measure for heuristic selection in examination timetabling. PATAT V. Revised Selected Papers. Springer LNCS.
21. Yang, X. S. (2008). Nature-Inspired Metaheuristic Algorithms. Luniver Press
22. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61.

23. Yang, X., & Hossein Gandomi, A. (2012). Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5), 464–483.
24. Lindfield, G., & Penny, J. (2017). The Firefly Algorithm. Introduction to Nature-Inspired Optimization, 85–100.
25. Payne RB, Sorenson MD, Klitz K. The cuckoos. Oxford, UK: Oxford University Press; 2005.
26. Pavlyukevich I. Lévy flights, non-local search and simulated annealing. J Comput Phys 2007;
27. Reynolds AM, Frye MA. Free-flight odor tracking in Drosophila is consistent with an optimal intermittent scale-free search. PLoS One 2007.
28. Yang XS, Deb S. Engineering optimization by cuckoo search. Int J Math Model Numer Optimisation 2010
29. Gandomi AH, Yang XS, Alavi AH. Cuckoo search algorithm: a meteheuristic approach to solve structural optimization problems. Eng Comput 2013
30. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems.
31. A. Borning, R. Duisberg, B. Freeman-Benson, A. Kramer, and M. Woolf. Constraint hierarchies. In Norman Meyrowitz, editor, Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), volume 22, pages 48–60, New York, NY, 1987. ACM Press.
32. Alan Borning, Bjorn Freeman-Benson, and Molly Wilson. Constraint hierarchies. Lisp and Symbolic Computation, 5(3):223–270, Sep 1992.
33. P. David. A constraint-based approach for examination timetabling using local repair techniques. In E. K. Burke and M. W. Carter, editors, Practice and Theory of Automated Timetabling: Selected Papers from the 2nd International Conference., volume 1408 of Springer Lecture Notes in Computer Science, pages 169–186, 1998.
34. T. A. Duong and K. H. Lam. Combining constraint programming and simulated annealing on university exam timetabling. In Proceedings of the 2nd International Conference in Computer Sciences, Research, Innovation & Vision for the Future (RIVF2004), pages 205–210, Hanoi, Vietnam, February 2004.
35. E. K. Burke, A. J. Eckersley, B. McCollum, S. Petrovic, and R. Qu. Hybrid variable neighbourhood approaches to university exam timetabling. Technical Report NOTTCS-TR- 2006-2, School of CSiT, University of Nottingham, 2006.
36. S. Ahmadi, R. Barone, P. Cheng, P. Cowling, and B. McCollum. Perturbation based variable neighbourhood search in heuristic space for examination timetabling problem. In Proceedings of Multidisciplinary International Scheduling: Theory and Applications (MISTA 2003), pages 155–171, Nottingham, August 13-16,, 2003. ISBN: 0-9545821-2-8.
37. Alefragis, P., Gogos, C., Valouxis, C., & Housos, E. (2021). A multiple metaheuristic variable neighborhood search framework for the Uncapacitated Examination Timetabling Problem. In Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT (Vol. 1, pp. 159-171).Gogos, C., Dimitsas, A., Nastos, V., & Valouxis, C. (2021, September). Some insights about the uncapacitated examination timetabling problem. In 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM) (pp. 1-7). IEEE.