

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΜΕΣΣΟΛΟΓΓΙΟΥ
ΤΜΗΜΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ & ΔΙΚΤΥΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**" Μελέτη αλγορίθμων χρονοπρογραμματισμού σε
τριτοβάθμια ιδρύματα "**

ΠΑΝΑΓΙΩΤΗΣ ΦΡΑΓΚΟΥΛΗΣ

ΕΠΙΒΛΕΠΩΝ: Παναγιώτης Αλεφραγκής – Καθηγητής Εφαρμογών

ΝΑΥΠΑΚΤΟΣ 2013

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Ναύπακτος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Ονοματεπώνυμο, Υπογραφή
2. Ονοματεπώνυμο, Υπογραφή
3. Ονοματεπώνυμο, Υπογραφή

Περίληψη:

Ο προγραμματισμός μαθημάτων και εξετάσεων στα ιδρύματα της τριτοβάθμιας εκπαίδευσης είναι γνωστό ότι είναι ένα πρόγραμμα με εξαιρετικά αυστηρούς περιορισμούς. Το γεγονός ότι πολλά ιδρύματα αποτελούνται πλέον από αρκετά τμήματα οδήγησε σε σημαντική αύξηση της πολυπλοκότητας, επιβάλλοντας ακόμη περισσότερες δυσκολίες στους διαχειριστές που πρέπει να βρουν μια λύση, συχνά χωρίς τη βοήθεια ηλεκτρονικού υπολογιστή. Από τις πολλές μεθόδους που έχουν εφαρμοστεί για την επίλυση του προβλήματος αυτόματα, οι εξελικτικές τεχνικές έχουν δείξει πολλές υποσχέσεις που οφείλεται στις ικανότητες βελτιστοποίησης γενικού σκοπού. Επίσης έχει διαπιστωθεί ότι οι υβριδικές εξελικτικές μέθοδοι μπορεί να αποφέρουν ακόμα καλύτερα αποτελέσματα. Σε αυτό το έγγραφο θα παρουσιαστεί μια έρευνα σχετικά με τις τρεις βασικές κατηγορίες χρονικού προγραμματισμού, όπως οι Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling), Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling), Κατάρτιση Προγράμματος Εξεταστικής (Examination Timetabling) και τις τεχνικές που χρησιμοποιούν οι αντίστοιχοι αλγόριθμοι. Τέλος θα δούμε μέσω ενός γραφικού περιβάλλοντος πως παρουσιάζονται τα αποτελέσματα που προκύπτουν από τις παραπάνω τεχνικές.

Summary:

The scheduling of courses and examinations to institutions of higher education is known to be a program with very strict limitations. The fact that many institutions are now several segments led to a significant increase in complexity, imposing even more problems for administrators who need to find a solution, often without the computer. Of the many methods have been applied to solve this problem, evolutionary techniques have shown great promise due to the general purpose optimization capabilities. He has also found that the hybrid evolutionary methods may yield even better results. In this paper will present an investigation into three basic categories of scheduling, such as Curriculum Based Course Timetabling, Post Enrollment based Course Timetabling, Examination Timetabling and techniques using the corresponding algorithms. Finally we see through a graphical user interface that presents the results obtained from these techniques.

Πίνακας περιεχομένων

Κεφάλαιο 1.....	10
Εισαγωγή.....	10
Το Πρόβλημα Χρονοπρογραμματισμού (Timetabling Problem).....	10
Τεχνικές Επίλυσης	11
Εφικτή λύση, Βελτιστοποίηση και Πολυπλοκότητα.....	11
Αντικείμενο της Έρευνας.....	12
Κεφάλαιο 2.....	13
Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling).....	13
Γενικά.....	13
Μεταβλητές και Περιορισμοί.....	14
Τρόποι Προσέγγισης – Αλγόριθμοι	16
Προεπεξεργασία.....	16
1. Προσέγγιση Αντιδραστικής Δομής.....	18
Εποικοδομητική Προσέγγιση.....	18

Διαδικασίες Βελτίωσης	22
2. Προσέγγιση Αποδοχής Κατωφλίου (Threshold Accepting).	24
Μέθοδος προσέγγισης Μυωπικής Δομής (Myopic Construction).	24
3. Προσέγγιση Διαδοχικής Τοπικής Αναζήτησης (Sequential Local Search).....	26
Εποικοδομητικός Αλγόριθμος (Construction Algorithm).....	26
Αλγόριθμος Βελτίωσης (Improvement Algorithm)	27
4. Προσέγγιση Υβριδικού Ευρετικού Αλγορίθμου (Hybrid Heuristic Algorithm).....	30
Γειτονική Δομή (Neighborhood Structure)	32
Αλγόριθμος Tabu Search χρησιμοποιώντας μια συνδυασμένη γειτονιά.	35
Διατάραξη	36
Συνδυασμός του Tabu Search με Διαταραχές.....	37
Κεφάλαιο 3.....	39
Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling).....	39
Γενικά.....	39
Περιγραφή του προβλήματος.....	41
1. Υβριδική προσέγγιση για την Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling).	44
Ο ενισχυμένος γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης - Φάση I του HGATS	46
Αρχικοποίηση του πληθυσμού.....	48
Η δομή των δεδομένων <i>MEM</i>	49
Δημιουργώντας ένα παιδί από την οργανωμένη στρατηγική αναζήτησης.....	51
Μεταλλαγή.....	53
Τοπικές Μέθοδοι Αναζήτησης.....	53
The tabu search heuristic - Φάση II του HGATS.....	58
2. Μη-κυριαρχούμενος Γενετικός Αλγόριθμος Ταξινόμησης II με μεταβαλλόμενο μέγεθος πληθυσμού - (Non-dominated Sorting Genetic Algorithm–II with a variable population size - NSGA-II VPS)	60
Περιγραφή του προβλήματος.....	60
Α. Αρχικό πρόβλημα.....	61
Β. Νέος Ελαστικός Περιορισμός.....	63
Non-dominated Sorting Genetic Algorithm–II with a variable population size – (NSGA-II VPS)	65
Α. NSGA-II VPS.....	65
Β. Διαδικασία Διασταύρωσης και Μετάλλαξης (Crossover and Mutation process)	68
C. Γειτονικές Δομές (Neighbourhood Structures)	69

Υπολογισμοί Διάρκειας Ζωής (Lifetime Calculations).....	70
3. Ένας Μεταερευνητικός Αλγόριθμος Βασισμένος στο Χρόνο (A time-dependent metaheuristic algorithm .).....	70
1. Περιγραφή του προβλήματος και συναρτήσεις κόστους	70
2. Περιγραφή του Αλγορίθμου.....	74
2.1 Θέματα Κωδικοποίησης και Προεπεξεργασίας.	76
2.2 Περιγραφή του Αλγορίθμου: Στάδιο 1.....	77
2.3 Περιγραφή του Αλγορίθμου: Στάδιο 2.....	82
2.4 Περιγραφή του Αλγορίθμου: Στάδιο 3.....	88
4. Λύνοντας το πρόβλημα κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών με βελτιστοποίηση αποικίας μυρμηγκιών (Ant Colony Optimization)	90
1. Ο Αλγόριθμος	90
5. Ένα Elitist-Ant σύστημα για επίλυση προβλημάτων κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem).....	100
1. Ο Αλγόριθμος	100
2. Πειραματικά αποτελέσματα	105
Κεφάλαιο 4.....	108
Κατάρτιση Προγράμματος Εξεταστικής	108
Γενικά	108
Τα προβλήματα κατάρτισης προγράμματος εξεταστικής.	111
Προσεγγίσεις / Τεχνικές Κατάρτισης Προγράμματος Εξεταστικής	114
1. Διαδοχικές Τεχνικές Βασισμένες σε Γράφους	114
2. Τεχνικές Βασισμένες σε Παραμέτρους.....	118
3. Τεχνικές βασισμένες στην Τοπική Αναζήτηση.....	121
4. Αλγόριθμοι Βασισμένοι στον Πληθυσμό.....	130
5. Τεχνικές Πολλαπλών Κριτηρίων.....	138
6. Υπέρ - Ευρετικά	139
Κεφάλαιο 5.....	143
Πειράματα και Συγκρίσεις	143
1. Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling)	143
Περιπτώσεις Προβλήματος και Πειραματικό Πρωτόκολλο.....	143
Συγκριτικά Αποτελέσματα και Συζήτηση	144
Συμπεράσματα.....	146

2. Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling).....	146
Ανάλυση ευαισθησίας των βασικών παραμέτρων του HGATS	147
Σύγκριση με σχετικούς αλγορίθμους.....	152
Σύγκριση με αλγορίθμους από τη βιβλιογραφία	155
Συμπέρασμα και μελλοντική εργασία	158
3. Κατάρτιση Προγράμματος εξεταστικής (Examination Timetabling)	162
Δεδομένα αναφοράς κατάρτισης προβλήματος εξεταστικής	163
3.1. Δεδομένα Αναφοράς του Πανεπιστήμιου του Τορόντο	163
3.2. Δεδομένα Αναφοράς του Πανεπιστήμιου του Νότινχαμ.....	167
3.3. Δεδομένα Αναφοράς του Πανεπιστήμιου της Μελβούρνης	170
3.4. Αποτελέσματα των Προβλημάτων Αναφοράς.....	171
Συμπεράσματα και Μελλοντικές Κατευθύνσεις.....	175
Βιβλιογραφία.....	180

Κεφάλαιο 1

Εισαγωγή.

Το Πρόβλημα Χρονοπρογραμματισμού (Timetabling Problem)

Ένα πρόβλημα χρονοπρογραμματισμού (timetabling problem) αναφέρεται στον προγραμματισμό ενεργειών σε συγκεκριμένες χρονικές περιόδους και βάσει ορισμένων προϋποθέσεων, οι οποίες αφορούν τους πόρους που είναι διαθέσιμοι. Δοθέντος ενός συνόλου πόρων με συγκεκριμένες δυναμικότητες, ενός συνόλου από δραστηριότητες με συγκεκριμένες διάρκειες και απαιτήσεις πόρων και ενός συνόλου χρονικών περιορισμών μεταξύ των δραστηριοτήτων, ένα τέτοιο πρόβλημα αποτελείται από τη λήψη αποφάσεων σχετικά με το πότε μια δραστηριότητα πρέπει να εκτελεστεί έτσι ώστε τόσο οι χρονικοί περιορισμοί όσο και περιορισμοί των πόρων να ικανοποιούνται. Κλασσικό πρόβλημα χρονοπρογραμματισμού αποτελεί η εύρεση ενός ωρολογίου προγράμματος μαθημάτων, που παραδίδουν καθηγητές σε ένα σύνολο μαθητών η φοιτητών. Σε αυτήν την περίπτωση, οι διαθέσιμοι πόροι είναι οι αίθουσες οι οποίες έχουν κάποια συγκεκριμένη χωρητικότητα και οι δραστηριότητες είναι οι εξετάσεις των μαθημάτων που πρέπει να πραγματοποιηθούν κάποια συγκεκριμένη χρονική στιγμή σε ένα σύνολο αιθουσών, έτσι ώστε το άθροισμα των χωρητικότητων των αιθουσών να υπερβαίνει τον αριθμό των φοιτητών που έχουν να εξεταστούν στο συγκεκριμένο μάθημα. Είναι ένα πρόβλημα που συναντάται αρκετά συχνά σε εκπαιδευτικά ιδρύματα, όπως σε σχολεία και πανεπιστήμια, καθώς τουλάχιστον μια φορά το χρόνο κατασκευάζεται ένα πρόγραμμα μαθημάτων σε ένα σχολείο και αρκετές σε ένα πανεπιστήμιο. Ένας μεγάλος αριθμός παραλλαγών του προβλήματος χρονοπρογραμματισμού

έχει προταθεί, ανάλογα με τον τύπο του ιδρύματος που αφορά ένα πρόβλημα (πανεπιστήμιο ή σχολείο) και το είδος των περιορισμών που περιλαμβάνει.

Τεχνικές Επίλυσης

Διάφορες τεχνικές έχουν προταθεί για την επίλυση προβλημάτων χρονοπρογραμματισμού. Οι τεχνικές αυτές είναι νευρωνικά δίκτυα, Ευρετικές τεχνικές (heuristics), χρωματισμός γράφων, ακέραιος προγραμματισμός, οι γενετικοί αλγόριθμοι, και προγραμματισμός με λογική περιορισμών. Τα μοντέλα που διατυπώθηκαν από ορισμένες τεχνικές δεν μπορούν εύκολα να προσαρμοστούν ή να αναδιατυπωθούν για να υποστηρίξουν αλλαγές. Μια πιθανή λύση είναι να χρησιμοποιήσουμε προγραμματισμό με λογική περιορισμών (Constraint Logic Programming - CLP). Ο προγραμματισμός με λογική περιορισμών αποτελείται από το ενσωμάτωση των δύο μεθόδων, του λογικού προγραμματισμού και την επίλυση των περιορισμών. Ο λογικός προγραμματισμός έχει τη δυνατότητα υποστήριξης δηλωτικού προγραμματισμού όπου οι χρήστες διαμορφώνουν το μοντέλο τους με το να αναφέρουν απλώς το τι είναι αληθινό και τι είναι ψευδές στη γλώσσα των χρηστών. Η επίλυση περιορισμών του προγραμματισμού με λογική περιορισμών μειώνει το χώρο αναζήτησης ενός συνδυαστικού προβλήματος με τον αποκλεισμό όλων των αδύνατων τιμών μέσω της διάδοσης των περιορισμών.

Εφικτή λύση, Βελτιστοποίηση και Πολυπλοκότητα

Σε ορισμένες περιπτώσεις, το πρόβλημα χρονοπρογραμματισμού αποτελείται από την εύρεση οποιουδήποτε χρονοδιαγράμματος που ικανοποιεί όλους τους περιορισμούς. Σε αυτές τις περιπτώσεις, το πρόβλημα διατυπώνεται ως πρόβλημα αναζήτησης.

Σε άλλες περιπτώσεις, το πρόβλημα διατυπώνεται ως πρόβλημα βελτιστοποίησης. Αυτό που απαιτείται είναι ένα χρονοδιάγραμμα που ικανοποιεί όλους τους αυστηρούς περιορισμούς (hard constraints) και ελαχιστοποιεί (ή μεγιστοποιεί) μια συγκεκριμένη αντικειμενική συνάρτηση η οποία ενσωματώνει τους ελαστικούς περιορισμούς (soft constraints). Όπως φαίνεται αργότερα, σε κάποιες προσεγγίσεις, η διαμόρφωση βελτιστοποίησης είναι απλά ένα μέσο για να εφαρμοστούν οι τεχνικές βελτιστοποίησης σε ένα πρόβλημα αναζήτησης. Αυτό γίνεται γενικά για να διευκολυνθεί η αναζήτηση για την καλύτερη λύση.

Και στις δύο περιπτώσεις (αναζήτηση και βελτιστοποίηση), ορίζουμε το βασικό πρόβλημα, που είναι το πρόβλημα για να αποφασιστεί αν υπάρχει μια λύση, στην περίπτωση ενός πρόβλημα αναζήτησης, και το πρόβλημα για να αποφασιστεί αν υπάρχει μια λύση με δεδομένη τιμή της αντικειμενικής συνάρτησης, στην περίπτωση ενός προβλήματος βελτιστοποίησης. Όταν αναφέρουμε την πολυπλοκότητα του προβλήματος, αναφερόμαστε στην πολυπλοκότητα του βασικού προβλήματος.

Αντικείμενο της Έρευνας

Στην περίπτωση μας θα αναλύσουμε τρεις κατηγορίες προβλημάτων χρονοπρογραμματισμού : Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling), Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling), Κατάρτιση Προγράμματος Εξεταστικής (Examination Timetabling) και τις μεθόδους που χρησιμοποιούνται σε αυτές.

Κεφάλαιο 2

Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling)

Γενικά

Το ωρολόγιο πρόγραμμα βάση του προγράμματος σπουδών είναι μια συγκεκριμένη παραλλαγή ενός προβλήματος εκπαιδευτικού χρονοδιαγράμματος. Αποτελείται από την κατασκευή ενός εβδομαδιαίου ωρολογίου προγράμματος με την ανάθεση διαλέξεων για διάφορα πανεπιστημιακά μαθήματα σε ένα δεδομένο αριθμό δωματίων και χρονικών περιόδων. Η σκιαγραφημένη κατάσταση μπορεί να βρεθεί σε πολλά πανεπιστήμια, όπου τα λεγόμενα προγράμματα σπουδών χρησιμοποιούνται για να περιγράψουν σύνολα μαθημάτων / διαλέξεων που μοιράζονται κοινούς φοιτητές. Η κυρίως λογική βασίζεται στην παραδοχή, ότι οι μαθητές οι οποίοι είναι εγγεγραμμένοι στο ίδιο πρόγραμμα, προχωρούν μαζί κατά τη διάρκεια των σπουδών τους. Ως εκ τούτου, οι εν λόγω σπουδαστές υποτίθεται ότι παρευρίσκονται σε ένα προηγούμενο καλώς ορισμένο σύνολο διαλέξεων. Αυτό μπορεί να φαίνεται αντίθετο με τα προβλήματα ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών, όπου οι μαθητές εγγράφονται ρητά για μαθήματα που επιθυμούν να παρακολουθήσουν. Ενώ σε αυτή τη ρύθμιση, λεπτομερείς πληροφορίες σχετικά με οποιονδήποτε μαθητή μπορούν να ανακτηθούν, στο ωρολόγιο πρόγραμμα με βάση το πρόγραμμα σπουδών, οι εγγραφές των μαθητών σε μαθήματα δεν απαιτούνται. Οι διαθέσιμοι περιορισμοί είναι αποκλειστικά βασισμένοι στον ορισμό του ωρολογίου προγράμματος.

Τα στοιχεία για το ωρολόγιο πρόγραμμα βάση του προγράμματος σπουδών είναι συγκριτικά εύκολο να επιτευχθούν. Μόλις καθοριστεί το αναλυτικό πρόγραμμα, που συνήθως δεν αλλάζει πολύ συχνά, ο χρονοπρογραμματισμός μπορεί να επεκταθεί για αρκετά χρόνια. Από την άλλη αντίθετα, οι φοιτητές που δεν ακολουθούν τους ορισμούς των προγραμμάτων σπουδών τους, μπορεί να καταλήξουν να έχουν πρόβλημα, καθώς πολλές διαλέξεις θα προγραμματιστούν εκ παραλλήλου. Η εφαρμογή του προβλήματος ωρολογίου προγράμματος βάση του προγράμματος σπουδών επομένως απαιτεί από τους μαθητές να ακολουθούν τη δομή του προγράμματός τους.

Μεταβλητές και Περιορισμοί

Το ωρολόγιο πρόγραμμα βάση του προγράμματος μαθημάτων αποτελείται από τον εβδομαδιαίο προγραμματισμό των διαλέξεων για αρκετά πανεπιστημιακά μαθήματα σε ένα δεδομένο αριθμό δωματίων και χρονικών περιόδων. Οι διενέξεις μεταξύ των μαθημάτων καθορίζονται σύμφωνα με το αναλυτικό πρόγραμμα που δημοσιεύθηκε από το Πανεπιστήμιο και όχι με βάση τα στοιχεία της εγγραφής.

Το πρόβλημα συνίσταται από τις ακόλουθες οντότητες:

Ημέρες, ώρες διεξαγωγής, καθώς και περίοδοι. Ο αριθμός των διδακτικών ημερών την εβδομάδα είναι δεδομένες (συνήθως 5 ή 6). Κάθε μέρα είναι χωρισμένη σε ένα σταθερό αριθμό των χρονοθυρίδων, ο οποίος είναι ίσος για όλες τις ημέρες. Μια περίοδος είναι ένα ζευγάρι που αποτελείται από μια ημέρα και μια χρονική στιγμή. Ο συνολικός αριθμός των προγραμματισμένων περιόδων είναι το γινόμενο του αριθμού των ημερών επί τον αριθμό των καθημερινών χρονοθυρίδων.

Μαθήματα και καθηγητές. Κάθε μάθημα αποτελείται από ένα σταθερό αριθμό διαλέξεων που θα προγραμματιστούν σε διαφορετικές περιόδους, παρακολουθείται από ένα δεδομένο αριθμό

των φοιτητών, και διδάσκεται από έναν καθηγητή. Για κάθε μάθημα υπάρχει ένας ελάχιστος αριθμός των ημερών κατά τις οποίες οι διαλέξεις του μαθήματος θα έπρεπε να εξαπλωθούν, άλλωστε, υπάρχουν ορισμένες περιόδους κατά τις οποίες το μάθημα δεν μπορεί να προγραμματιστεί.

Οι αίθουσες. Κάθε αίθουσα έχει χωρητικότητα, που εκφράζεται ως ο αριθμός των διαθέσιμων θέσεων. Όλες οι αίθουσες είναι εξίσου κατάλληλες για όλα τα μαθήματα (αν είναι αρκετά μεγάλες).

Ωρολόγια προγράμματα σπουδών. Ένα ωρολόγιο πρόγραμμα σπουδών είναι μια ομάδα μαθημάτων, έτσι ώστε κάθε ζευγάρι μαθημάτων στην ομάδα να έχει κοινούς μαθητές. Οι διενέξεις μεταξύ των μαθημάτων, και άλλοι ελαστικοί περιορισμοί (soft constraints), βασίζονται στα προγράμματα σπουδών.

Η λύση του προβλήματος είναι η ανάθεση μιας περιόδου (ημέρας και χρονοθυρίδας) και μία αίθουσα σε όλες τις διαλέξεις του κάθε μαθήματος. Οι ακόλουθοι αυστηροί περιορισμοί (hard constraints) θα πρέπει να πληρούνται:

Όλες οι διαλέξεις των μαθημάτων θα πρέπει να προγραμματιστούν, και θα πρέπει να ανατεθούν σε διαφορετικές περιόδους.

Δύο διαλέξεις δεν μπορεί να πραγματοποιηθούν στον ίδιο χώρο κατά την ίδια περίοδο.

Διαλέξεις των μαθημάτων στο ίδιο πρόγραμμα σπουδών, ή που διδάσκονται από τον ίδιο καθηγητή, πρέπει να προγραμματιστούν σε διαφορετικές περιόδους.

Αν ο καθηγητής του μαθήματος δεν είναι διαθέσιμος για να διδάξει το μάθημα σε μια δεδομένη περίοδο, τότε δεν μπορούν να προγραμματιστούν διαλέξεις των μαθημάτων κατά την περίοδο αυτή.

Το πρόβλημα περιλαμβάνει τις ακόλουθους ελαστικούς περιορισμούς (soft constraints):

Χωρητικότητα αίθουσας. Για κάθε διάλεξη, ο αριθμός των φοιτητών που παρακολουθούν το μάθημα πρέπει να είναι μικρότερος ή ίσος με τον αριθμό των θέσεων σε όλα τα δωμάτια που φιλοξενούν διαλέξεις του μαθήματος. Κάθε παραπάνω φοιτητής από την χωρητικότητα της αίθουσας μετριέται ως ένας βαθμός ποινής.

Ελάχιστες ημέρες εργασίας. Οι διαλέξεις του κάθε μαθήματος θα πρέπει να κατανεμηθούν στον ελάχιστο αριθμό ημερών που δίνεται. Κάθε μέρα κάτω από το ελάχιστο μετράει ως πέντε βαθμούς ποινής.

Συγκέντρωση ωρολογίου προγράμματος. Διαλέξεις που ανήκουν σε ένα ωρολόγιο πρόγραμμα σπουδών θα πρέπει να είναι κοντινές μεταξύ τους (για παράδειγμα, σε διαδοχικές περιόδους). Για ένα δεδομένο πρόγραμμα λαμβάνεται υπόψη σαν παράβαση κάθε φορά που μια διάλεξη δεν είναι κοντά σε οποιαδήποτε άλλη διάλεξη, κατά την ίδια ημέρα. Κάθε απομονωμένη διάλεξη σε ένα πρόγραμμα σπουδών μετρά ως δύο βαθμοί ποινής.

Σταθερότητα αίθουσας. Όλες οι διαλέξεις ενός μαθήματος θα πρέπει να παραδίδονται στην ίδια αίθουσα. Κάθε ξεχωριστό δωμάτιο που χρησιμοποιείται για τις διαλέξεις ενός μαθήματος εκτός από του αρχικού, μετριέται ως ένας βαθμός ποινής.

Λαμβάνοντας υπόψη όλους τους παραπάνω περιορισμούς παρακάτω θα αναλυθούν οι τρόποι με τους οποίους προσεγγίζονται καθώς και οι αλγόριθμοι που χρησιμοποιούνται.

Τρόποι Προσέγγισης – Αλγόριθμοι

Προεπεξεργασία

Πριν από τον υπολογισμό μιας πρώτης λύσης, πραγματοποιείται κάποια προεπεξεργασία. Συνοπτικά, υιοθετούνται μερικά ειδικά χαρακτηριστικά του προβλήματος, προσθέτοντας κάποια επιπλέον δομή στο πρόβλημα.

Για κάθε δεδομένη διάλεξη L_i , συμβάντα E_{i1}, \dots, E_{ie} δημιουργούνται τα οποία αργότερα ανατίθενται στις χρονοθυρίδες. Ο αριθμός των συμβάντων e δίνονται στις περιπτώσεις του προβλήματος. Η δημιουργία συμβάντων για κάθε διάλεξη οδηγεί σε μια πιο γενική περιγραφή του προβλήματος, και η προσέγγιση της λύσης πρέπει να επικεντρωθεί μόνο για την ανάθεση όλων των συμβάντων, σε μία και μόνο χρονοθυρίδα, σε αντίθεση με την παρακολούθηση της ανάθεσης μιας διάλεξης σε μια e χρονοθυρίδα.

Δεύτερον, έχουν κατηγοριοποιηθεί για κάθε διάλεξη L_i (και κατά συνέπεια για κάθε συμβάν που ανήκει στη διάλεξη L_i) τα διαθέσιμα δωμάτια σε τρεις διαζευκτικές κατηγορίες R_{i1} , R_{i2} , R_{i3} .

Η R_{i1} αναφέρεται στις αίθουσες στις οποίες η διάλεξη ταιριάζει καλύτερα, που είναι οι αίθουσες R_k με την ελάχιστη θετική ή μηδενική αξία των $c_k - s_i$, με το c_k να είναι η χωρητικότητα της αίθουσας, και το s_i ο αριθμός των μαθητών της διάλεξης L_i . Η κλάση R_{i2} αποθηκεύει τις αίθουσες στις οποίες ταιριάζει η διάλεξη L_i , όπου ισχύει $s_i < c_k$, αλλά όχι βέλτιστα, και η R_{i3} περιέχει τις αίθουσες στις οποίες η διάλεξη L_i δεν ταιριάζει. Όσον αφορά τη συγκεκριμένη κατάσταση του προβλήματος, συμβάντα διαλέξεων μπορούν να ανατεθούν σε χρονοθυρίδες αιθουσών στην R_{i3} , αυτό όμως οδηγεί σε μία ποινή.

Η βασική παραδοχή της κατάταξης των αιθουσών είναι ότι τα συμβάντα ανατίθενται κατά προτίμηση σε χρονοθυρίδες που ανήκουν σε μια αίθουσα της κλάσης R_{i1} , ακολουθούμενη από τις R_{i2} και R_{i3} . Πρέπει να αναφερθεί ωστόσο, ότι αυτό δεν μπορεί να θεωρηθεί ως ένας δεσμευτικός, γενικός κανόνας, αλλά μάλλον θα πρέπει να θεωρηθεί ως σύσταση. Μια τυχαιοποιημένη διαδικασία ως εκ τούτου, πρόκειται να εφαρμοστεί κατά την ανάθεση συμβάντων σε χρονοθυρίδες, που επιτρέπουν μια συγκεκριμένη παρέκκλιση από την υπολογισμένη κατάταξη των αιθουσών.

1. Προσέγγιση Αντιδραστικής Δομής

Μετά από ένα στάδιο προεπεξεργασίας, στο οποίο κάποιες ιδιότητες σχετικά με το πρόβλημα προ-υπολογίζονται και δομούνται, δύο βασικές μέθοδοι χρησιμοποιούνται για την κατασκευή και βελτίωση χρονοδιαγραμμάτων για το ποσοτικό μοντέλο:

1. Μια εποικοδομητική προσέγγιση χρησιμοποιείται για να ληφθεί μια πρώτη εφικτή ανάθεση όλων των διαλέξεων. Σε αυτή τη φάση, οι ελαστικοί περιορισμοί (soft constraints) δεν λαμβάνονται ακόμη υπόψη, αλλά η μέθοδος επικεντρώνεται στους αυστηρούς περιορισμούς (hard constraints) του προβλήματος.
2. Μια επαναληπτική προσέγγιση εκτελείται συνεχόμενα για τη βελτίωση του χρονοδιαγράμματος που λαμβάνονται από την εποικοδομητική προσέγγιση. Κατά τη διάρκεια αυτής της φάσης, πραγματοποιούνται ανακατανομές των διαλέξεων με τελικό στόχο τον εντοπισμό ενός βέλτιστου χρονοδιαγράμματος.

Εποικοδομητική Προσέγγιση

Με βάση τα αποτελέσματα για την αρχική εποικοδομητική προσέγγιση, προτείνεται μια αντιδραστική διαδικασία που αυτό-προσαρμόζεται στο σύνολο των μη ανατιθέμενων συμβάντων από προηγούμενες αναλύσεις. Η λογική πίσω από αυτή την προσέγγιση είναι ότι η εποικοδομητική διαδικασία «ανακαλύπτει» συμβάντα που είναι δύσκολο να ανατεθούν, δίνοντάς τους προτεραιότητα σε διαδοχικές ανατρέξεις. Επιπλέον, δανείζονται ιδέες από την προσέγγιση της βελτιστοποίησης “τσιριχτού” τροχού (squeaky wheel optimization). Σε αυτή την περίπτωση, οι εναλλακτικές λύσεις κατασκευάζονται και αναλύονται επαναλαμβανόμενα. Δυσμενείς πτυχές της παρούσας λύσης ανακαλύπτονται σε κάθε ανάλυση, οι οποίες στη συνέχεια επιλύονται στη διαδοχική επανάληψη. Γεγονότα που δεν έχουν ανατεθεί σε

προηγούμενες εποικοδομητικές αναλύσεις εδώ θεωρούνται ότι είναι οι δυσμενείς πτυχές της παρούσας λύσης.

Στη συνέχεια, υποθέτουμε ότι η \mathcal{E}^p είναι το σύνολο των πιο σημαντικών συμβάντων, η \mathcal{E}^a το σύνολο των όχι σημαντικών συμβάντων, και η \mathcal{E}^c , το σύνολο των συμβάντων που δεν έχουν ανατεθεί κατά τη διάρκεια της κατασκευαστικής φάσης. Είναι απαραίτητο ότι $\mathcal{E}^p \sqcup \mathcal{E}^a \sqcup \mathcal{E}^c = \mathcal{E}$, $\mathcal{E}^p \cap \mathcal{E}^a = \emptyset$, και $\mathcal{E}^p \cap \mathcal{E}^c = \emptyset$.

Ο παρακρατώ αλγόριθμος περιγράφει την αντιδραστική διαδικασία κατασκευής.

Algorithm Reactive construction

Require: *Maxloops*

1: Set $\mathcal{E}^p = \emptyset$, $\mathcal{E}^a = \emptyset$, *loops* = 0

2: **repeat**

3: $\mathcal{E}^p \leftarrow \mathcal{E}^c$

4: $\mathcal{E}^a \leftarrow \emptyset$

5: $\mathcal{E}^c \leftarrow \mathcal{E} \setminus \mathcal{E}^p$

6: **while** $\mathcal{E}^p \neq \emptyset$ **do**

7: Select the most critical event E from \mathcal{E}^p , that is the event with the smallest number of available timeslots

8: **if** E can be assigned to at least one timeslot **then**

```

9:         Select some available timeslot  $T$  for  $E$ 
10:        Assign  $E$  to the timeslot  $T$ 
11:    else

12:         $\mathcal{E}^a \leftarrow \mathcal{E}^a \cup E$ 

13:    end if
14:     $\mathcal{E}^p \leftarrow \mathcal{E}^p \setminus E$ 
15:    end while

16:    while  $\mathcal{E}^p \neq \emptyset$  do

17:        Select the most critical event  $E$  from  $\mathcal{E}^p$ , that is the event with the smallest
        number of available timeslots
18:        if  $E$  can be assigned to at least one timeslot then
19:            Select some available timeslot  $T$  for  $E$ 
20:            Assign  $E$  to the timeslot  $T$ 
21:        else

22:             $\mathcal{E}^a \leftarrow \mathcal{E}^a \cup E$ 

23:        end if
24:         $\mathcal{E}^p \leftarrow \mathcal{E}^p \setminus E$ 
25:        end while
26:         $loops \leftarrow loops + 1$ 



---


27: until  $\mathcal{E}^a = \emptyset$  or  $loops = Maxloops$ 

```

Loops	Feasible solutions
1	0
2	56
3	272
4	387
5	511
6	608
7	688
8	754
9	802
10	831

Πίνακας 1. Εφικτές λύσεις μετά από ένα συγκεκριμένο αριθμό δοκιμών (1000 δοκιμές)

Όπως έχει δοθεί στον ψευδό-κώδικα, η κατασκευή των λύσεων πραγματοποιείται σε έναν βρόγχο είτε μέχρι να εντοπιστεί μια εφικτή λύση, είτε μέχρι να επιτευχτεί ο μέγιστος αριθμός των επαναλαμβανόμενων Maxloops. Όταν κατασκευάζεται μια λύση, ένα σύνολο συμβάντων \mathcal{E}^i διατηρούνται για εκείνα που δεν έχει βρεθεί χρονοθυρίδα. Όταν ανακατασκευάζεται μια λύση, αυτά τα συμβάντα έχουν προτεραιότητα έναντι των άλλων. Υπό αυτή την έννοια, η εποικοδομητική προσέγγιση είναι μεροληπτική σε σχέση με τις προηγούμενες αναλύσεις, εντοπίζει συμβάντα που αποδεικνύονται δύσκολο να ανατεθούν.

Μετά από ένα μέγιστο αριθμό επί το πλείστον Maxloops επαναλήψεων, η εποικοδομητική διαδικασία επιστρέφει μια λύση που είναι είτε εφικτή ($\mathcal{E}^i = \emptyset$) είτε όχι ($\mathcal{E}^i \neq \emptyset$).

Γίνεται σαφές ότι η αντιδραστική διαδικασία βασίζεται κυρίως στην προηγούμενη απλή άπληστη ευρετική διαδικασία. Η επιλογή του πιο κρίσιμου συμβάντος, καθώς και η επιλογή της χρονοθυρίδας παραμένουν αμετάβλητα. Μόνο η ιεράρχηση των συμβάντων μέσω της διαίρεσης τους σε δύο διαζευκτικά υποσύνολα είναι ένα πρόσθετο χαρακτηριστικό της αναθεωρημένης μεθόδου.

Διαδικασίες Βελτίωσης

Μια επαναληπτική διαδικασία συνεχίζεται με την εναλλακτική λύση να βρίσκεται στην εποικοδομητική προσέγγιση, που ψάχνει για μια βέλτιστη λύση όσον αφορά τις των ελαστικών περιορισμών (soft constraints).

Σε κάθε βήμα του αλγορίθμου, ένας αριθμός από τυχαία επιλεγμένα συμβάντα απορρίπτονται από το χρονοδιάγραμμα και την επανατοποθετούνται στο σύνολο της \mathcal{E}^n . Ακολουθεί μια φάση επανάθεσης. Σε αντίθεση με την εποικοδομητική προσέγγιση, όπου τα συμβάντα επιλέγονται με βάση σχετικά με το αν είναι κρίσιμης σημασίας σε σχέση με τις διαθέσιμες χρονοθυρίδες, τα συμβάντα τώρα επιλέγονται τυχαία από την \mathcal{E}^n , και κάθε συμβάν με ίση πιθανότητα. Η επιλογή της χρονοθυρίδας για το συμβάν βασίζεται στη λογική που περιγράφεται στην φάση της προεπεξεργασίας, δίνοντας προτεραιότητα σε χρονοθυρίδες συγκεκριμένων κλάσεων αιθουσών.

Οι ακόλουθες διαφορετικές παραλλαγές τοπικής αναζήτησης έχουν εφαρμοστεί και δοκιμαστεί:

- Ανάβαση Λόφου (Hillclimbing).

Σε αυτή την παραλλαγή τοπικής αναζήτησης, γίνονται δεκτές μόνο βελτιωμένες αναδιοργανώσεις των συμβάντων. Αναμένεται πιθανώς ότι αυτή η στρατηγική δεν οδηγεί

σε καλύτερα αποτελέσματα. Ωστόσο, για λόγους σύγκρισης, μια εφαρμογή θα είναι ενδιαφέρουσα, απλά επειδή η αποτελεσματικότητα των εναλλακτικών στρατηγικών τοπικής αναζήτησης μπορούν να μελετηθούν σε αντίθεση με αυτό το σχετικά απλό αλγόριθμο.

➤ Επαναλαμβανόμενη Τοπική Αναζήτηση (Iterated Local Search).

Η επαναλαμβανόμενη τοπική αναζήτηση βασίζεται σε έναν αλγόριθμο ανάβασης λόφων (Hillclimbing), ο οποίος χρησιμοποιείται για πρώτη φορά για να υπολογίσει μία τοπικά βέλτιστη λύση. Στη συνέχεια, αφού γίνεται σύγκλιση προς αυτή η εναλλακτική λύση, ένας μηχανισμός διαφυγής ενεργοποιείται, που αποτελείται από μία επιδεινωμένη κίνηση διατάραξης μέσω κάποιας γειτονιάς. Η αναζήτηση συνεχίζεται από αυτήν την εναλλακτική διαταραγμένη λύση, και πάλι εκτελώντας μια ανάβαση λόφου (Hillclimbing).

➤ Αποδοχή Κατωφλίου (Threshold Accepting).

Η ιδέα της αποδοχής κατωφλίου περιγράφει μια ντετερμινιστική παραλλαγή της προσομοιωμένης απόπτωσης (Simulated Annealing). Κινήσεις επιδείνωσης είναι δεκτές μέχρι ένα ορισμένο κατώφλι, επιτρέποντας έτσι μια απόδραση από την διαδικασία αναζήτησης από το τοπικό βέλτιστο. Καθ' όλη την εκτέλεση της προσέγγισης τοπικής αναζήτησης, το κατώφλι στη συνέχεια μειώθηκε, και έγινε παρόμοιο με αυτό που αναφέρεται ως ένα «πρόγραμμα ψύξης» στην προσομοιωμένη απόπτωση (Simulated Annealing).

Όπως προηγούμενες έρευνες έχουν δείξει ότι οι απλουστεύσεις της προσομοιωμένης απόπτωσης (Simulated Annealing) μπορεί να είναι πολύ αποτελεσματικές για τα προβλήματα χρονοπρογραμματισμού, υποψιαζόμαστε ότι αυτή η προσέγγιση αποδεικνύεται επιτυχής για το πρόβλημα.

2. Προσέγγιση Αποδοχής Κατωφλίου (Threshold Accepting).

Μέθοδος προσέγγισης Μυωπικής Δομής (Myopic Construction).

Η εποικοδομητική φάση προσπαθεί να αποκτήσει μια πρώτη εφικτή ανάθεση όλων των συμβάντων σε χρονοθυρίδες. Μια απλή ευρετική προσέγγιση χρησιμοποιείται, διαδοχικά αναθέτοντας όλα τα συμβάντα σε χρονοθυρίδες, ένα κάθε φορά, με το δεδομένο ψευδό-κώδικα του αλγορίθμου Μυωπικής Δομής. Σε αυτήν την περιγραφή, ορίζουμε το σύνολο όλων των συμβάντων με \mathcal{E} , και το σύνολο των μη εκχωρημένων (ανοιχτών) συμβάντων με \mathcal{E}° . Κατά τη διάρκεια της διαδικασίας διαδοχικών αναθέσεων, ένα σύνολο από συμβάντα που ήταν αδύνατο να ανατεθούν διατηρείται, και συμβολίζεται με \mathcal{E}'' . Σε περιπτώσεις ανάθεσης όλων των συμβάντων σε χρονοθυρίδες, επιστρέφεται $\mathcal{E}'' = \emptyset$.

Μια άπληστη προσέγγιση χρησιμοποιείται κατά τη διαδικασία ανάθεσης, επιλέγοντας σε κάθε βήμα το «πιο κρίσιμο» συμβάν E από το \mathcal{E}° , που είναι το συμβάν με τον μικρότερο αριθμό των χρονοθυρίδων στις οποίες μπορεί να ανατεθεί.

Η επιλογή των χρονοθυρίδων για τα συμβάντα που καθρεπτίζει την αρχική κατηγοριοποίηση των αιθουσών. Με μία πιθανότητα της τάξης του 0,5, οι χρονοθυρίδες των αιθουσών στην R_{i1} είναι προτιμότερες από των R_{i2} και R_{i3} , και με πιθανότητα της τάξης του 0,5, οι χρονοθυρίδες των αιθουσών της R_{i2} προτιμάται σε σχέση με εκείνες των R_{i1} και R_{i3} . Μέσα σε κάθε τάξη, οι χρονοθυρίδες επιλέγονται τυχαία με ίση πιθανότητα. Σε περιπτώσεις όπου μία πλέον προτιμώμενη κλάση χρονοθυρίδων είναι άδεια, η επιλογή γίνεται από τη μικρότερη προτιμώμενη τάξη και ούτω καθεξής.

Όπως αναφέρθηκε παραπάνω, οι χρονοθυρίδες των αιθουσών της κλάσης R_{i1} είναι προτιμότερες από αυτές των κλάσεων των R_{i2} και R_{i3} . Η τυχαιοποιημένη διαδικασία ανάθεσης γενικά λαμβάνει υπόψη αυτή την πτυχή, εντούτοις επιτρέποντας κάποια παρέκκλιση από τον κανόνα. Αυτό γίνεται, όπως είμαστε σε θέση να παρατηρήσουμε ότι η ανάθεση των συμβάντων σε χρονοθυρίδες ακολουθώντας μια μόνο σειρά δεν οδηγεί σε ικανοποιητικά αποτελέσματα. Στην περίπτωση αυτή, η επιλογή των χρονοθυρίδων είναι απλά πάρα πολύ περιοριστική.

Έχει επισημανθεί στο πλαίσιο αυτό ότι η πιθανότητα ανάθεσης συμβάντων σε χρονοθυρίδες της $R_{i1} \rightarrow R_{i2} \rightarrow R_{i3}$ αναμένεται να είναι μεγαλύτερη από εκείνη της σειράς $R_{i2} \rightarrow R_{i1} \rightarrow R_{i3}$. Ενώ γενικά συμφωνούμε με αυτό το σχόλιο, άλλες πιθανότητες εκτός από 0,5 για και οι δύο σειρές δεν έχουν διερευνηθεί ακόμα. Κατά συνέπεια, επόμενα πειράματα σίγουρα θα πρέπει να εξετάσουν την επίδραση αυτής της παραμέτρου ελέγχου σχετικά με τα επιτευχθέντα αποτελέσματα.

Algorithm Myopic construction

1: Set $\mathcal{E}^0 = \mathcal{E}$

2: $\mathcal{E}^u \leftarrow \emptyset$

```

3: while  $\mathcal{E}^o \neq \emptyset$  do
4:   Select the most critical event  $E$  from  $\mathcal{E}^o$ , that is the event with the smallest number of
   available timeslots
5:   if  $E$  can be assigned to at least one timeslot then
6:     Select some available timeslot  $T$  for  $E$ 
7:     Assign  $E$  to the timeslot  $T$ 
8:   else
9:      $\mathcal{E}^u \leftarrow \mathcal{E}^u \sqcup E$ 
10:  end if
11:   $\mathcal{E}^o \leftarrow \mathcal{E}^o \setminus E$ 
12: end while

```

Instance	Cases with $\mathcal{E}^u = \emptyset$
comp01.ctt	1,000
comp02.ctt	354
comp03.ctt	377
comp04.ctt	1,000
comp05.ctt	0
comp06.ctt	953
comp07.ctt	827

Πίνακας 2. Αριθμός δοκιμών όπου όλα τα συμβάντα έχουν ανατεθεί επιτυχώς (1000 δόκιμες)

3. Προσέγγιση Διαδοχικής Τοπικής Αναζήτησης (Sequential Local Search)

Εποικοδομητικός Αλγόριθμος (Construction Algorithm)

Ο προσέγγιση του αλγόριθμου που αναλύεται εδώ χωρίζεται σε δύο στάδια: την αρχική κατασκευή και την βελτίωση. Από τη βιβλιογραφία μπορεί να παρατηρηθεί ότι οι περισσότερες λύσεις της αρχικής δομής σε σχέση με τα προβλήματα χρονοπρογραμματισμού

πανεπιστήμιου δημιουργούνται με ευρετικά χρωματισμού γράφων. Αυτό έχει άμεση σχέση με το πώς το πρόβλημα εν γένει αντιπροσωπεύεται. Βελτίωση επομένως συχνά επιτυγχάνεται μέσω της χρήσης μιας μεταευριστικής προσέγγισης. Στη συνέχεια περιγράφεται η προσέγγιση που χρησιμοποιείται εδώ.

Ο εποικοδομητικός αλγόριθμος που προτείνεται από τον Chiarandini et al. και τους Landa-Silva και Obit , χρησιμοποιείται για να παράγει μεγάλους αριθμούς από τυχαίες αρχικές λύσεις. Ο εποικοδομητικός αλγόριθμος αποτελείται από τρεις φάσεις, το ευρετικό μεγαλύτερου βαθμού, αναζήτηση γειτονιάς (neighborhood search) και την tabu search όπως παρουσιάζονται σε μορφή ψευδό-κώδικα στην εικόνα. 1. Θεωρούμε αυτόν τον αλγόριθμο ως έναν υβριδικό εποικοδομητικό αλγόριθμο. Η προσέγγιση αυτή επιλέχθηκε κυρίως επειδή ήταν σε θέση να παράγει εφικτές λύσεις για όλα τα σύνολα δεδομένων χάριν στο συνδυασμό της δύναμης από τις τρεις φάσεις που εμπλέκονται.

```
set population size, Popsize
for i =0 to i < Popsize
    for j=0 to j < Number of lectures
        Phase 1:Apply largest degree heuristic
    end for
    do while (timetable infeasible)
        Phase 2:Apply neighborhood search
        Phase 3:Apply tabu search
    end do
end for
return population of feasible timetables
```

Εικόνα. 1 Ψευδό-κώδικας για τον εποικοδομητικό αλγόριθμο

Αλγόριθμος Βελτίωσης (Improvement Algorithm)

Ο αλγόριθμος βελτίωσης εμπλέκει την εκτέλεση γενετικού αλγορίθμου και διαδοχική τοπική αναζήτηση. Ο γενετικός αλγόριθμος παρουσιάστηκε από τον John Holland το 1975 , ο οποίος

είναι μια οικογένεια υπολογιστικών μοντέλων που χρησιμοποιεί μεθόδους που βρίσκονται στη φυσική βιολογική εξέλιξη. Ο γενετικός αλγόριθμος είναι μία προσαρμοστική μέθοδος που χρησιμοποιείται για την επίλυση προβλημάτων αναζήτησης και βελτιστοποίησης. Οι γενετικοί αλγόριθμοι κωδικοποιούν μία ενδεχόμενη επίλυση ενός συγκεκριμένου προβλήματος σε ένα απλό χρωμόσωμα, όπως μία δομή δεδομένων και εφαρμόζουν φορείς ανασυνδυασμού σε αυτές τις δομές, έτσι ώστε να διατηρηθούν κρίσιμες πληροφορίες. Χρησιμοποιείται για αναζήτηση χώρου μεγάλων και μη γραμμικών λύσεων όπου εξειδικευμένη γνώση παραλείπεται ή είναι δύσκολο να κωδικοποιηθεί. Επιπλέον, δεν απαιτεί καθόλου βαθμονομημένες πληροφορίες, εξελίσσεται από το ένα πληθυσμό στον άλλον και παράγει πολλαπλές βέλτιστες λύσεις παρά από μόνο μία τοπική. Αυτά τα χαρακτηριστικά καθιστούν τους γενετικούς αλγόριθμους το πιο κατάλληλο εργαλείο για προβλήματα βελτιστοποίησης.

Ο κανονικός γενετικός αλγόριθμος εστιάζει στην καθολική πλευρά της περιοχής αναζήτησης, ενώ οι τοπικές μέθοδοι αναζήτησης βασίζονται κυρίως στην τοπική περιοχή. Όσο πιο εντατική είναι η τοπική αναζήτηση, τόσο ισχυρότερη είναι η ανάγκη εξειδικευμένων πληροφοριών για τη λειτουργία που πρέπει να ελαχιστοποιηθεί ή να μεγιστοποιηθεί (εξαρτάται από το πρόβλημα που πρέπει να επιλυθεί). Μάλιστα, ο 'υβριδισμός' μεταξύ γενετικών αλγορίθμων και άλλων μεθόδων τοπικής αναζήτησης έχει ήδη προταθεί και δοκιμαστεί σε ένα σημαντικό αριθμό εργασιών και δείχνουν την αποτελεσματικότητα του συνδυασμού αυτού. Λόγω αυτής της δύναμης, ο αλγόριθμος που χρησιμοποιείται εδώ συνδυάζει γενετικό αλγόριθμο και διαδοχική τοπική αναζήτηση. Η Εικόνα 2 δείχνει μια σχηματική όψη της προσέγγισης. Στην αρχή της έρευνας, ένα σύνολο πληθυσμών σχηματίζεται από τις παραγόμενες αρχικές λύσεις. Σε κάθε γενιά, πολλαπλές μεμονωμένες οντότητες επιλέγονται από τον τρέχον πληθυσμό με βάση την καταλληλότητά τους, ανασυνδυάζονται και μεταλλάσσονται να σχηματίσουν ένα νέο πληθυσμό, ο οποίος παίρνει την θέση του προηγούμενου στην επόμενη επανάληψη του

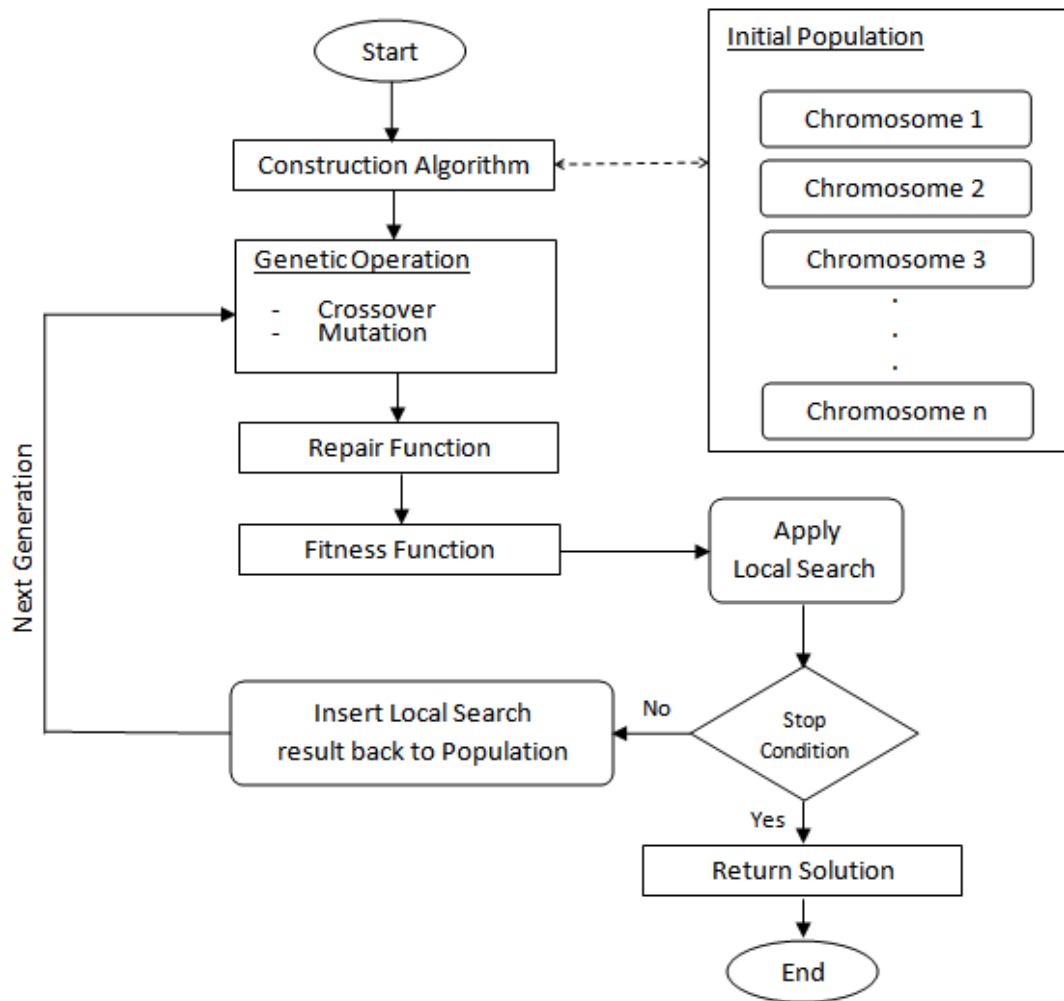
αλγόριθμου. Η επισκευαστική λειτουργία εφαρμόζεται να κάνει ανέφικτες λύσεις εφικτές, καθώς αυτό λειτουργεί σε τρία στάδια, όπως παρακάτω:

Βήμα 1 : Εύρεση ελεύθερων χρονοθυρίδων για κάθε αίθουσα

Βήμα 2 : Εύρεση ελεύθερων χρονοθυρίδων για κάθε συμβάν

Βήμα 3 : Εύρεση εφικτών χρονοθυρίδων για αίθουσες και συμβάντα (π.χ. μια διασταύρωση μεταξύ του βήματος 1 και 2).

Διαδοχική Τοπική Αναζήτηση: Πριν αναλυθεί η επόμενη γενιά, θα εφαρμοστεί ένα αλγόριθμος γειτονικής αναζήτησης για τη βελτίωση του χρονοδιαγράμματος. Αυτή η διαδικασία κάνει τη σύγκλιση του γενετικού αλγορίθμου γρηγορότερα. Τοποθετείται εκ νέου τη λύση της τοπικής αναζήτησης πίσω στον γενετικό αλγόριθμο για να εξεταστεί στην επόμενη γενιά.



Εικόνα 2. Μια σχηματική όψη της προσέγγισης

4. Προσέγγιση Υβριδικού Ευρετικού Αλγορίθμου (Hybrid Heuristic Algorithm)

Η βασική ιδέα του υβριδικού ευρετικού αλγορίθμου μας είναι ο συνδυασμός των πλεονεκτικών χαρακτηριστικών της Tabu Search (TS) και της Επαναλαμβανόμενης Τοπικής Αναζήτησης (ILS). Σχεδιάζεται σε αυτό το κεφάλαιο ένας προσαρμοστικός αλγόριθμος TS του οποίου τα εξαρτήματα και μηχανισμοί περιγράφονται στα ακόλουθα υποκεφάλαια.

Ο αλγόριθμος Tabu Search βασίζεται στην πεποίθηση ότι η έξυπνη αναζήτηση θα πρέπει να είναι συστηματικά βασισμένη στην προσαρμοστική μνήμη και τη μάθηση. Ο αλγόριθμος

Tabu Search μπορεί να χρησιμοποιηθεί τόσο με μεγάλους όσο και με μικρούς προϋπολογισμούς υπολογισμών. Σε γενικές γραμμές, οι μεγάλοι προϋπολογισμοί υπολογισμών θα οδηγήσουν σε καλύτερα αποτελέσματα. Ωστόσο, εάν ο συνολικός χρόνος υπολογισμού είναι περιορισμένος, θα ήταν προτιμώμενο να συνδυαστούν μικρής διάρκειας αλγόριθμοι Tabu Search με ορισμένους ισχυρά διαφοροποιήσιμους τελεστές.

Είναι ενδιαφέρον, ότι ο αλγόριθμος Επαναλαμβανόμενης Τοπικής Αναζήτησης παρέχει τέτοιους μηχανισμούς διαφοροποίησης για να καθοδηγήσουν την αναζήτηση ώστε να ξεφύγει από ένα τοπικό βέλτιστο και να κινηθεί προς νέες περιοχές στο χώρο των λύσεων. Όταν η καλύτερα γνωστή λύση δεν μπορεί να βελτιωθεί πλέον χρησιμοποιώντας τον αλγόριθμο Tabu Search, εφαρμόζεται ένας τελεστής διαταραχής καθοδηγούμενος από ποινές για να καταστρέψει το παραγόμενο τοπικό βέλτιστο.

Σημειώστε ότι ξεκινώντας από ένα άδειο χρονοδιάγραμμα, έχει δημιουργηθεί πρώτα μια αρχική εφικτή λύση μέσω ενός γραφικά χρωματιζόμενου άπληστου ευρετικού. Αρκεί να αναφερθεί ότι για την όλες τις δοκιμασμένες περιπτώσεις, αυτή η άπληστη ευρετική μπορούν εύκολα να αποκτήσει εφικτές λύσεις. Μόλις ένα εφικτό χρονοδιάγραμμα που να ικανοποιεί όλους τους αυστηρούς περιορισμούς που έχει επιτευχθεί, ο παρών αλγόριθμος Προσαρμοσμένης Tabu Αναζήτησης χρησιμοποιείται για την ελαχιστοποίηση της λειτουργίας κόστους ελαστικών περιορισμών χωρίς να παραβιάζει αυστηρούς περιορισμούς. Ως εκ τούτου, ο χώρος αναζήτησης του παρόντος αλγόριθμου Προσαρμοσμένης Tabu Αναζήτησης περιορίζεται στα εφικτά χρονοδιαγράμματα.

Ένα ενδιαφέρον ζήτημα αφορά την επίδραση της αρχικής λύσης στην τελική λύση στην οποία κατέληξε ο αλγόριθμος Προσαρμοσμένης Tabu Αναζήτησης. Πειραματισμοί δείχνουν ότι αλγόριθμος Προσαρμοσμένης Tabu Αναζήτησης δεν είναι ευαίσθητος στην ποιότητα της αρχικής λύσης.

Γειτονική Δομή (Neighborhood Structure)

Σε μια διαδικασία γειτονικής αναζήτησης, εφαρμόζοντας μια κίνηση mn σε μια υποψήφια λύση X οδηγεί σε μια νέα λύση που συμβολίζεται με $X \oplus mn$. Ας οριστεί το $M(X)$ να είναι το σύνολο όλων των πιθανών κινήσεων που μπορούν να εφαρμοστούν στην X και δεν δημιουργούν καμία μη εφικτή λύση, τότε η «γειτονιά» της X ορίζεται από: $N(X) = \{ X \oplus mn \mid mn \in M(X) \}$. Για το πρόβλημα του Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών,

χρησιμοποιούμε δύο διαφορετικές κινήσεις που αναφέρονται ως SimpleSwap και KempeSwap. Αντίστοιχα, οι δύο «γειτονιές» που συμβολίζονται από $N1$ και $N2$, ορίζονται ως εξής.

Γειτονιά $N1$ (Neighborhood $N1$): Μια κίνηση SimpleSwap αποτελείται από την ανταλλαγή των φιλοξενουμένων περιόδων και τις αίθουσες που έχουν ανατεθεί σε δύο διαλέξεις διαφορετικών μαθημάτων. Εφαρμόζοντας την SimpleSwap κίνηση σε δύο διαφορετικά μαθήματα x_i, j και $x_{i'}, j'$ για τη λύση X αποτελείται από την ανάθεση της τιμής του x_i, j στο $x_{i'}, j'$ και αντιστρόφως την τιμή του $x_{i'}, j'$ στο x_i, j . Σημειώνεται ότι η μετακίνηση μια διάλεξης ενός μαθήματος σε μια ελεύθερη θέση είναι μια ειδική περίπτωση της κίνησης SimpleSwap όπου μία από τις ανταλλασσόμενες διαλέξεις είναι μηδενική και επίσης περιλαμβάνεται στη «γειτονιά» $N1$.

Γειτονιά $N2$ (Neighborhood $N2$): Η κίνηση KempeSwap ορίζεται από την εναλλαγή δύο Kempe αλυσίδων. Αν εστιάσουμε μόνο στα μαθήματα και τις συγκρούσεις, κάθε στιγμιότυπο του προβλήματος μπορεί να θεωρηθεί ως ένας γράφος G όπου οι κόμβοι είναι τα μαθήματα και οι άκρες συνδέουν μαθήματα με μαθητές ή κοινούς καθηγητές. Σε ένα εφικτό

χρονοδιάγραμμα, μια αλυσίδα Kempe είναι το σύνολο των κόμβων που σχηματίζουν ένα συνδεδεμένο στοιχείο στον υπογράφο του G που προκαλείται από τους κόμβους που ανήκουν σε δύο περιόδους. Μια κίνηση KempeSwar παράγει μία νέα εφικτή ανάθεση με την εναλλαγή των ετικετών περιόδου που είναι αναθεμένες στα τα μαθήματα που ανήκουν σε δύο συγκεκριμένες αλυσίδες Kempe. Μόλις τα μαθήματα έχουν προγραμματιστεί σε περιόδους, η ανάθεση αιθουσών μπορεί να γίνει με την επίλυση ενός διμερούς προβλήματος αντιστοίχισης.

Πιο επίσημα, ορίζουμε $K1$ και $K2$ ως δύο αλυσίδες Kempe στον υπογράφο με έμφαση σε δύο περιόδους ti και tj , μια KempeSwar κίνηση παράγει μια ανάθεση με την αντικατάσταση της ti με την $(ti \setminus (K1 \cup K2)) \cup (tj \cap (K1 \cup K2))$ και με tj $(tj \setminus (K1 \cup K2)) \cup (ti \cap (K1 \cup K2))$.

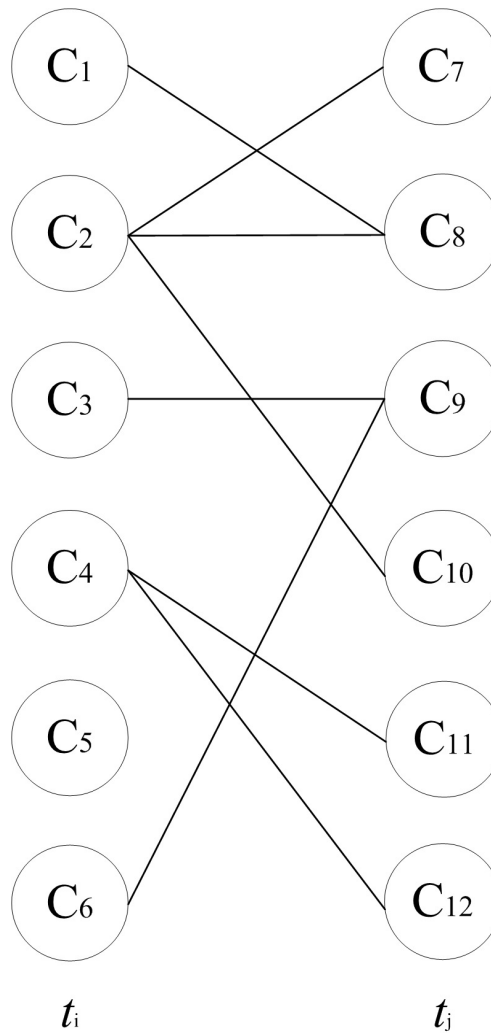
Για παράδειγμα, η εικόνα 3 απεικονίζει έναν υπογράφο συναγόμενο από δύο περιόδους ti και tj και υπάρχουν τέσσερις αλυσίδες Kempe: $Ka = \{c1, c2, c7, c8, c10\}$, $Kb = \{c3, c6, c9\}$, $Kc = \{c4, c11, c12\}$ και $Kd = \{c5\}$. Στη συνέχεια, αν ανταλλάξουμε δύο αλυσίδες Kempe Kb και Kc , μία KempeSwar παράγει μια ανάθεση με την κίνηση των $\{c3, c4, c6\}$ στην tj και των $\{c9, c11, c12\}$ στην ti .

Αξίζει να υπογραμμιστεί ότι η διπλή ανταλλαγή αλυσίδων Kempe μπορεί να

θεωρείται ως μια γενίκευση της ενιαίας ανταλλαγής αλυσίδας Kempe ως γνωστόν στην λογοτεχνία. Στο προηγούμενο ορισμό της μονής γειτονιάς αλυσίδας Kempe, κάθε κίνηση αφορά μόνο μία συνδεδεμένη συνιστώσα, δηλαδή, μία από τις δύο αλυσίδες Kempe στον ορισμό μας είναι άδεια. Τυπικά, αυτό σημαίνει την αντικατάσταση της ti με $(ti \setminus K) \cup (tj \cap K)$

και της t_j με $(t_j \setminus K) \cup (t_i \cap K)$, όπου η K είναι η μη-κενή αλυσίδα Kempe. Κατά συνέπεια, η

μονή ανταλλαγή αλυσίδας Kempe είναι ειδική περίπτωση της κίνησης KempeSwap και περιλαμβάνεται στο γειτονιά $N2$. Αν και δεν αναφέρονται, μια λεπτομερής ανάλυση και σύγκριση αυτών και άλλων γειτονιών διεξάγεται, δείχνοντας την αποτελεσματικότητα της νέας προτεινόμενης γειτονιάς διπλής αλυσίδας Kempe.



Εικόνα 3. Εικονογραφήσεις της αλυσίδας Kempe

Αλγόριθμος Tabu Search χρησιμοποιώντας μια συνδυασμένη γειτονιά.

Η βασική μηχανή αναζήτησης του προσαρμοζόμενου αλγόριθμου Tabu Search είναι φυσικά βασισμένη στον Tabu Search. Η Tabu Search διαδικασία εκμεταλλεύεται τις δύο γειτονιές $N1$ και $N2$ με τρόπο Token-Ring. Πιο συγκεκριμένα, ξεκινά η διαδικασία Tabu Search με μια γειτονιά. Όταν η αναζήτηση τελειώνει με το καλύτερο τοπικό βέλτιστο της, γίνεται επανεκκίνηση της Tabu Search από αυτό το τοπικό βέλτιστο, αλλά με την άλλη γειτονιά. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να μην είναι δυνατή άλλη βελτίωση και τότε θεωρείται επιτυχής. Στην περίπτωση μας, η Tabu Search διαδικασία ξεκινά από τη βασική γειτονιά $N1$ και στη συνέχεια $N2$: $N1 \rightarrow N2 \rightarrow N1 \rightarrow N2 \dots$

Στον αλγόριθμο Tabu Search, μία Tabu λίστα εισάγεται να απαγορεύσει τις κινήσεις που έχουν ήδη επισκεφθεί. Σε κάθε επανάληψη, μια βέλτιστη μη-Tabu κίνηση mn εφαρμόζεται στην τρέχουσα λύση X ακόμα και αν η $X' \neq X \oplus mn$ δεν βελτιώνει την ποιότητα λύσης. Στον Tabu Search, όταν μετακινείτε μια διάλεξη από τη μία θέση (ζεύγος περιόδου-αίθουσας) στην άλλη ($N1$), ή από μία περίοδο σε μια άλλη ($N2$), αυτή η διάλεξη δεν μπορεί να μετακινηθεί πίσω στην αρχική της θέση ($N1$) ή περίοδο ($N2$) για τις επόμενες tt επαναλήψεις (tt ονομάζεται Tabu tenure χρόνος κατοχής). Ο Tabu χρόνος κατοχής μίας διάλεξης x συντονίζεται προσαρμοστικά σύμφωνα με την ισχύουσα ποιότητα λύσης f και την συχνότητα της κίνησης $freq(x)$, δηλαδή

$$tt(x) = f + \varphi \cdot freq(x)$$

όπου φ είναι μια παράμετρος που βασίζεται στο $[0, 1]$. Το κριτήριο αναρρόφησης δέχεται μία Tabu κίνηση εάν βελτιώνει το τρέχον καλύτερο αποτέλεσμα ή το σύνολο των μη-Tabu κινήσεων να είναι άδειο στη τρέχουσα γειτονιά. Η Tabu Search διαδικασία βασισμένη σε κάθε γειτονιά σταματά όταν η καλύτερη λύση δεν μπορεί να βελτιωθεί μέσα σε ένα συγκεκριμένο αριθμό βημάτων (συμβολίζονται με θ) και καλείται, ως το βάθος του Tabu Search. Η βασική διαδικασία Tabu Search περιγράφεται στον παρακάτω αλγόριθμο :

Algorithm Tabu Search procedure: $TS(X_0, \theta)$

- 1: // X_0 is the feasible initial solution
 - 2: // θ is the depth of TS
 - 3: **repeat**
 - 4: $X^* = TS_{N1}(X_0)$ based on $N1$ with depth of $TS \theta$
 - 5: $X^{*'} = TS_{N2}(X^*)$ based on $N2$ with depth of $TS \theta/3$
 - 6: $X_0 = X^{*'}$
 - 7: **until** (no improvement is reached)
-

Λόγω της υψηλής υπολογιστικής προσπάθεια στην αξιολόγηση της γειτονιάς $N2$, ο Tabu Search χρησιμοποιεί ένα πολύ μικρότερο βάθος (εμπειρικά καθορίζεται σε $\theta / 3$), όταν είναι η $N2$ χρησιμοποιείται. Σημειώστε ότι η **συμβολική-δαχτυλίδι** αναζήτηση βασισμένη στον Tabu Search σταματά όταν καμία βελτίωση είναι δυνατή. Στο αυτό το σημείο, μια φάση Tabu Search έχει τελειώσει.

Διατάραξη

Όταν μια φάση Tabu Search τερματίζει, εμπλέκεται ένας τελεστής διαταραχής για να καταστρέψει την μέχρι τώρα τοπική λύση για την επανέναρξη μιας νέας φάσης Tabu Search από αυτήν την διαταραγμένη λύση. Ο τελεστής διαταραχής αποτελείται από έναν τυχαία επιλεγμένου δεδομένου αριθμού κινήσεων SimpleSwap ή KempeSwap, όπου τουλάχιστον ένα μία από τις μετακινούμενες διαλέξεις ανήκει στην πρώτη k που έχουν υψηλό βαθμό ποινής. Συγκεκριμένα, όταν η τρέχουσα φάση Tabu Search τερματίζει, όλες οι διαλέξεις

κατατάσσονται σε μια μη-αυξανόμενη σειρά σύμφωνα με τα ελαστικά κόστη τους. Στη συνέχεια, ένας ορισμένος αριθμός διαλέξεων επιλέγεται από τα πρώτα k -κατάταξης (που έχουν υψηλό βαθμό ποινής). Παρατηρείτε ότι ο περιορισμός της επιλογής των διαλέξεων με υψηλό βαθμό ποινής είναι απαραίτητος επειδή είναι αυτές οι διαλέξεις που συντελούν σε μεγάλο βαθμό στις παραβιάσεις των περιορισμών (και το κόστος λειτουργίας).

Προφανώς, η δύναμη διαταραχής (συμβολίζεται με γ) είναι ένα από τα πιο σημαντικά συστατικά της ILS και καθορίζει το ποιοτικό χάσμα μεταξύ των δύο λύσεων πριν και μετά την διαταραχή. Στην περίπτωσή μας, η γ είναι προσαρμοστικά ρυθμισμένη και παίρνει τιμές στο διάστημα $[\gamma_{min}, \gamma_{max}]$. Για κριτήριο αποδοχής στη διαδικασία διατάραξης, χρησιμοποιούμε μια ισχυρή τεχνική εκμετάλλευσης, δηλαδή, γίνονται δεκτές μόνο καλύτερες λύσεις.

Συνδυασμός του Tabu Search με Διαταραχές

Το βάθος του Tabu Search θ και η ισχύς διαταραχής γ είναι δύο βασικές παράμετροι που ελέγχουν τη συμπεριφορά του αλγόριθμου Adaptive TabuSearch. Από τη μία πλευρά, μια μεγαλύτερη τιμή του θ εξασφαλίζει μια πιο εντατική αναζήτηση. Από την άλλη πλευρά, η μεγαλύτερη τιμή της γ αντιστοιχεί με περισσότερες πιθανότητες διαφυγής από το τρέχων τοπικό ελάχιστο. Για να πραγματοποιηθεί μια συνεχής ανταλλαγή μεταξύ της εντατικοποίησης και διαφοροποίησης, επινοείται ένας μηχανισμός για να ρυθμίσει δυναμικά και προσαρμοστικά αυτές τις δύο σημαντικές παραμέτρους σύμφωνα με τα ιστορικά αρχεία της αναζήτησης. Σημειώστε ότι εδώ οι αρχικές τιμές των δύο αυτών και άλλων παραμέτρων ορίζονται εμπειρικά και είναι όλες ανεξάρτητες από τα στιγμιότυπα. Είναι πιθανό ότι η καλύτερες λύσεις θα βρεθούν χρησιμοποιώντας ένα σύνολο παραμέτρων που εξαρτώνται από το στιγμιότυπο. Ωστόσο, στόχος μας είναι να σχεδιαστεί ένας ισχυρός λύτης που είναι σε θέση να επιλύσει αποτελεσματικά ένα μεγάλο πίνακα περιπτώσεων.

Στην αρχή της έρευνας, παίρνουμε μια βασική Tabu Search όπου το βάθος του Tabu Search θ είναι ένας μικρός θετικός αριθμός, ως πούμε $\theta = \theta_0$ ($\theta_0 = 10$). Όταν ο Tabu Search δεν μπορούν να βελτιώσουν την καλύτερή του λύση, η διαταραχή εφαρμόζεται σε αυτήν την καλύτερη λύση με μια αδύναμη δύναμη ($\gamma = \gamma_{min}$). Όταν η αναζήτηση προχωράει, καταγράφεται ο αριθμός των επαναλήψεων της Tabu Search φάσης (συμβολίζεται με ζ) για τις οποίες δεν έχει βρεθεί ακόμα βελτιωμένη λύση. Το βάθος της Tabu Search θ και η δύναμη διατάραξης γ δυναμικά προσαρμόζονται ως εξής: Όταν το τοπικό ελάχιστο που λαμβάνεται με Tabu Search είναι υποσχόμενο, δηλαδή, όταν είναι κοντά στην τρέχουσα βέλτιστη λύση ($f \leq f_{best} + 2$), το βάθος του Tabu Search αυξάνεται σταδιακά για να εξασφαλιστεί μία όλο και πιο εντατική έρευνα μέχρι να παύσει να υπάρχει βελτίωση, δηλαδή, $\theta = (1 + \eta) \theta$ σε κάθε επανάληψη ($\eta = 0,6$). Ομοίως, η δύναμη διαταραχής αυξάνεται σταδιακά, έτσι ώστε να διαφοροποιηθεί πολύ περισσότερο η αναζήτηση, εάν ο αριθμός των μη βελτιώσιμων επαναλήψεων φάσης Tabu Search αυξάνεται.

Εδώ, χρησιμοποιείται ο όρος λήξης που απαιτείται από τους κανόνες του διαγωνισμού ITC-2007. Τέλος, ο υβριδικός αλγόριθμος Adaptive Tabu Search περιγράφεται στον αλγόριθμο παρακάτω.

Algorithm. Adaptive Tabu Search scheme

- 1: X_0 is a feasible solution, X^* is the best solution found so far
- 2: set $\zeta = 0$, $\theta = \theta_0$, $\gamma = \gamma_{min}$
- 3: apply TS to X_0 with *depth of TS* θ : $X^* = TS(X_0, \theta)$
- 4: **repeat**
- 5: perturb X^* with perturbation strength γ , get X'
- 6: apply TS to X' with depth of TS θ , get $X^{*'}$
- 7: **if** the local minimum solution $X^{*'}$ is promising, i.e., $f(X^{*'}) \leq f(X^*) + 2$ then
- 8: **repeat**
- 9: call the TS procedure with a gradually increased θ : $\theta = (1 + \eta)\theta$
- 10: **until** no better solution is obtained
- 11: **end if**
- 12: **if** better solution $X^{*'}$ has been found, i.e., $f(X^{*'}) < f(X^*)$ **then**

```
13:         accept  $X^{*}$ ' as the current best solution:  $X^* = X^{*}$ '
14:         reset to the basic TS ( $\theta = \theta_0$ ) with weak perturbation ( $\gamma = \gamma_{min}$ )
15:     else
16:         reset to the basic TS:  $\theta = \theta_0$ 
17:          $\xi = \xi + 1$ 
18:         update the perturbation strength:  $\gamma = \min\{\gamma_{min} + \lambda \cdot \xi, \gamma_{max}\}$ 
19:     end if
20: until (timeout condition is met)
```

Κεφάλαιο 3

Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling)

Γενικά

Το είδος του προβλήματος χρονοπρογραμματισμού που περιγράφεται σε αυτήν την ενότητα είναι το πρόβλημα Κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling). Το συγκεκριμένο μοντέλο έχει σκοπό να μιμηθεί μία κατάσταση από τον πραγματικό κόσμο, όπου οι μαθητές έχουν τη δυνατότητα επιλογής των διαλέξεων που επιθυμούν να παρακολουθήσουν, καθώς και το χρονοδιάγραμμα κατασκευάζεται στη συνέχεια σύμφωνα με αυτές τις επιλογές (δηλαδή, το χρονοδιάγραμμα είναι να κατασκευαστεί αφού φοιτητές έχουν επιλέξει διαλέξεις που επιθυμούν να παραστούν). Η πρόθεσή μας σε αυτήν την ενότητα είναι να περιγραφεί αυτό το πρόβλημα με κάθε λεπτομέρεια, να υπογραμμιστούν τα κριτήρια αξιολόγησης και άλλοι συναφείς κανόνες που χρησιμοποιούνται με αυτό το μοντέλο, και να συζητηθούν γενικά τα πλεονεκτήματα και οι περιορισμοί του.

Στον αρχικό διαγωνισμό χρονοπρογραμματισμού, ένα μοντέλο προβλήματος είχε χρησιμοποιηθεί στο οποίο μια σειρά από «συμβάντα» έπρεπε να προγραμματιστούν σε αίθουσες και «χρονοθυρίδες», σύμφωνα με μια σειρά από περιορισμούς. Οι περιορισμοί αυτοί μπορούν να χωριστούν σε δύο κατηγορίες: οι αυστηροί περιορισμοί (hard constraints) και οι ελαστικοί περιορισμοί (soft constraints). Οι πρώτοι είναι υποχρεωτικοί στην ικανοποίησή τους και αντανακλούν περιορισμούς που πρέπει να πληρούνται προκειμένου το χρονοδιάγραμμα να είναι αξιοποιήσιμο. Οι τελευταίοι είναι εκείνοι που πρέπει να ικανοποιηθούν μόνο αν αυτό είναι δυνατόν και προορίζονται να κάνουν ένα χρονοδιάγραμμα "ωραίο" για τους ανθρώπους που πρέπει να το χρησιμοποιούν.

Ένα σημαντικό χαρακτηριστικό του αρχικού μοντέλου του διαγωνισμού ήταν ο τρόπος με τον οποίο η ποιότητα των λύσεων των νεοεισερχόμενων μετρήθηκε. Αποφασίστηκε από τους διοργανωτές του διαγωνισμού εκ των προτέρων ότι τα χρονοδιαγράμματα θα πρέπει να κριθούν μόνο από τον υπολογισμό του αριθμού των παραβιάσεων των ελαστικών περιορισμών κατά την προτεινόμενη λύση. Στην πραγματικότητα, οι αλγόριθμοι που είχαν το δικαίωμα να επιλεγούν στο διαγωνισμό, ήταν αυτοί που θα μπορούσαν να παράγουν εντός της προθεσμίας εφικτά χρονοδιαγράμματα. Ένας λόγος για αυτό ήταν να αποφευχθεί το πρόβλημα της απόφασης πώς να συγκριθούν δύο λύσεις με διαφορετικό αριθμό από παραβιασμένους αυστηρούς περιορισμούς και διαφορετικό αριθμό από παραβιασμένους ελαστικούς περιορισμούς. Κατά συνέπεια, οι περιπτώσεις του προβλήματος που χρησιμοποιήθηκαν σε αυτό το διαγωνισμό είναι ειδικά κατασκευασμένες έτσι ώστε οι αυστηροί περιορισμοί σε κάθε περίπτωση ήταν γενικά αρκετά εύκολο να ικανοποιηθούν.

Ένα αποτέλεσμα αυτού του κριτικού κριτηρίου ήταν ότι η πλειοψηφία των ιδεών που παρήχθησαν στον πρώτο διαγωνισμό είχε να κάνει με την ικανοποίηση των ελαστικών περιορισμών. Δηλαδή, πολλοί από τους αλγόριθμους που έχουν εγγραφεί θα λειτουργούν ικανοποιώντας γρήγορα τους αυστηρούς περιορισμούς μιας συγκεκριμένης περίπτωσης του

προβλήματος και στη συνέχεια θα αφιερώσουν το μεγαλύτερο μέρος του χρόνου τους και τις προσπάθειες που καταβάλλονται προκειμένου να ικανοποιηθούν οι ελαστικοί περιορισμοί του προβλήματος (καθώς δεν παραβιάζεται κανένας από τους αυστηρούς περιορισμούς στη διαδικασία). Μερικοί από αυτούς τους αλγόριθμους ήταν πολύ αποτελεσματικοί και πρόσθεσαν πολύτιμες γνώσεις στον τομέα. Ωστόσο, σε πολλές πραγματικές καταστάσεις χρονοπρογραμματισμού, η ικανοποίηση των αυστηρών περιορισμών ενός συγκεκριμένου προβλήματος μπορεί να μην είναι πάντα τόσο εύκολο.

Όπως προαναφέρθηκε, το μοντέλο προβλήματος που χρησιμοποιείται στο δεύτερο κομμάτι του δεύτερου Διεθνούς Διαγωνισμού Χρονοπρογραμματισμού είναι μια επέκταση του μοντέλου προβλήματος που χρησιμοποιήθηκε στον πρώτο διαγωνισμό. Ωστόσο, στην περίπτωση αυτή, οι επιπλέον περιορισμοί έχουν επίσης προστεθεί στο μοντέλο ώστε να προχωρήσει περαιτέρω προς την κατεύθυνση του πραγματικού χρονοπρογραμματισμού. Αυτό έχει επιτευχθεί με την προσθήκη δύο επιπλέον αυστηρών περιορισμών, οι οποίοι θα παρουσιαστούν εδώ.

Περιγραφή του προβλήματος

Το μοντέλο χρονοδιαγράμματος Κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών χρησιμοποιείται στο δεύτερο κομμάτι του Δεύτερου Διεθνούς Διαγωνισμού Χρονοπρογραμματισμού μπορεί να οριστεί ως εξής. Κατ' αρχάς, κάθε πρόβλημα αποτελείται από τις ακόλουθες πληροφορίες (σημειώστε ότι η ακριβής διάταξη των πληροφοριών σε αρχείο κάθε περίπτωση προβλήματος δίνεται στην ιστοσελίδα του διαγωνισμού www.cs.qub.ac.uk/itc2007/):

- Ένα σύνολο από n συμβάντα που πρόκειται να προγραμματιστούν σε 45 χρονοθυρίδες (5 ημέρες με 9 ώρες το καθένα).

- Ένα σύνολο r αιθουσών, που η καθεμία έχει μια συγκεκριμένη χωρητικότητα, στη οποία τα συμβάντα λαμβάνουν χώρα.
- Ένα σύνολο f χαρακτηριστικών αιθουσών που είναι ικανοποιημένα από τις αίθουσες και τα οποία απαιτούνται από τα συμβάντα.
- Ένα σύνολο s φοιτητών που παρακολουθούν πολλούς διαφορετικούς συνδυασμούς των συμβάντων.
- Ένα σύνολο διαθέσιμων χρονοθυρίδων για κάθε ένα από τα συμβάντα n (δεν θα είναι όλα τα συμβάντα διαθέσιμα σε όλες τις χρονοθυρίδες).
- Ένα σύνολο από απαιτήσεις προτεραιότητας που δηλώνουν ότι ορισμένα συμβάντα πρέπει να συμβούν πριν από κάποια άλλα.

Ο στόχος είναι να δοκιμάσουμε και τοποθετήσουμε κάθε ένα από τα n συμβάντα στο χρονοδιάγραμμα (δηλαδή, η ταξινόμηση καθένα από τα n συμβάντα σε μία από τις r αίθουσες και μία από τις 45 χρονοθυρίδες), ενώ πληρούν τους ακόλουθους πέντε αυστηρούς περιορισμούς:

- 1) Κανένας φοιτητής δεν θα πρέπει να απαιτείται να παρακολουθήσει περισσότερες από ένα συμβάν παράλληλα.
- 2) Σε κάθε περίπτωση, η αίθουσα πρέπει να είναι αρκετά μεγάλη για όλους τους φοιτητές που συμμετέχουν και θα πρέπει να πληροί όλα τα χαρακτηριστικά που απαιτούνται από το συμβάν.
- 3) Μόνο ένα συμβάν τοποθετείται σε κάθε δωμάτιο σε οποιαδήποτε χρονική στιγμή.
- 4) Τα συμβάντα θα πρέπει να ανατεθούν σε χρονοθυρίδες που είναι προ-ορίζονται ως "διαθέσιμες" γι' αυτά τα συμβάντα.
- 5) Σε περίπτωση που ορίζονται, τα συμβάντα θα πρέπει να προγραμματιστούν να πραγματοποιηθούν με τη σωστή σειρά μέσα στην εβδομάδα.

Σημειώστε ότι οι άνω σκληροί περιορισμοί 1, 2, και 3 είναι ακριβώς ίδιοι με τους σκληρούς περιορισμούς που χρησιμοποιήθηκαν στον πρώτο διαγωνισμό. Οι περιορισμοί 4 και 5, εν τω μεταξύ, είναι νέες προσθήκες στο μοντέλο.

Δεδομένου ότι είναι πλέον μη-ρεαλιστικό να αναμένουμε από όλους τους αλγορίθμους να ικανοποιούν όλους των αυστηρούς περιορισμούς εντός της δεδομένης προθεσμίας, θα έπρεπε να αντιμετωπιστεί το πρόβλημα του πώς να αντιμετωπιστούν με ανέφικτα χρονοδιαγράμματα. Η λύση ήταν να θεωρηθεί ότι οι λύσεις που υποβάλλονται ακόμα έπρεπε να μην έχουν παραβιάσεις αυστηρών περιορισμών, αλλά αυτό θα μπορούσε να επιτευχθεί, αφήνοντας μερικά συμβάντα έξω από το χρονοδιάγραμμα ή "χωρίς θέση".

Στο σημείο αυτό είναι χρήσιμο για μας να καθορίσουμε κάποια ορολογία:

- Ένα **έγκυρο** χρονοδιάγραμμα είναι αυτό στο οποίο δεν υπάρχουν περιστατικά από παραβιάσεις αυστηρών περιορισμών, αλλά μερικά από τα συμβάντα έχουν μείνει σε μία μεριά χωρίς θέση.
- Ένα **εφικτό** χρονοδιάγραμμα είναι εκείνο στο οποίο δεν υπάρχουν περιστατικά από παραβιάσεις αυστηρών περιορισμών, και όλα τα συμβάντα υπάρχουν στο χρονοδιάγραμμα.

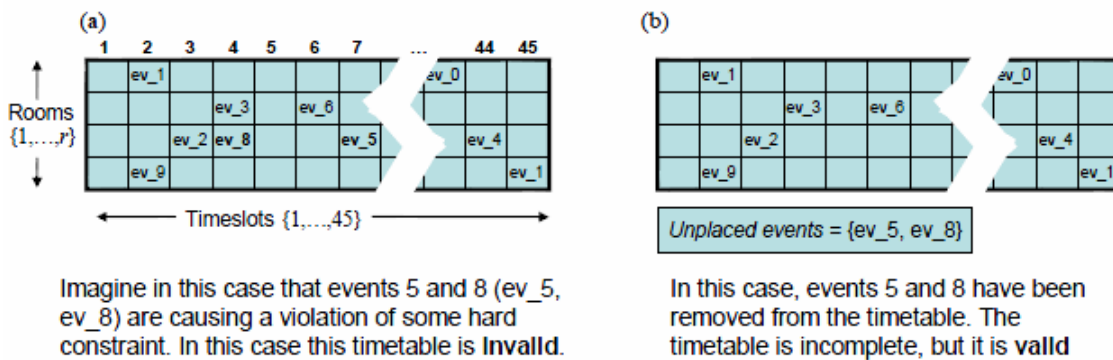
Για λόγους σαφήνειας, οι έννοιες αυτές απεικονίζονται στην εικόνα 4.

Επιπλέον, με τους πέντε αυστηρούς περιορισμούς που δίνονται παραπάνω, σε αυτό το μοντέλο προβλήματος είναι επίσης ενδιαφέρον η ικανοποίηση μιας σειράς από ελαστικούς περιορισμούς. Αυτοί έχουν ως εξής:

- 1) Οι φοιτητές δεν θα πρέπει να προγραμματιστεί να παραστούν σε ένα συμβάν κατά την τελευταία χρονοθυρίδα της ημέρας (δηλαδή, στις χρονοθυρίδες 9, 18, 27, 36, ή 45).
- 2) Οι φοιτητές δεν θα πρέπει να παρακολουθήσουν τρία (ή περισσότερα) συμβάντα σε διαδοχικές χρονοθυρίδες που συμβαίνουν την ίδια μέρα.

- 3) Οι φοιτητές δεν πρέπει να υποχρεωθούν να παρακολουθήσουν μόνο ένα συμβάν σε μια συγκεκριμένη ημέρα.

Αυτοί οι τρεις ελαστικοί περιορισμοί είναι ίδιοι με εκείνους που χρησιμοποιήθηκαν στον πρώτο διαγωνισμό.



Εικόνα 4. Παράδειγμα από άκυρα και έγκυρα χρονοδιαγράμματα σύμφωνα με τα κριτήρια του διαγωνισμού.

1. Υβριδική προσέγγιση για την Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling).

Οι γενετικοί αλγόριθμοι είναι μια κατηγορία από ισχυρά γενικού σκοπού εργαλεία βελτιστοποίησης που μοντελοποιούν τις αρχές της φυσικής εξέλιξης. Οι γενετικοί αλγόριθμοι είναι βασισμένες στον πληθυσμό ευρετικές μέθοδοι, οι οποίες ξεκινούν από έναν αρχικό πληθυσμό από τυχαίες λύσεις για ένα δεδομένο πρόβλημα. Κάθε λύση στον πληθυσμό ονομάζεται άτομο. Κάθε άτομο αξιολογείται σύμφωνα με μία αντικειμενική συνάρτηση για το συγκεκριμένο πρόβλημα, που συνήθως ονομάζεται συνάρτηση καταλληλότητας. Μετά την αξιολόγηση, υπάρχει είναι μια φάση επιλογής στην οποία, ενδεχομένως, πιθανότατα καλά άτομα θα επιλεγούν από έναν τελεστή επιλογής για να υποβληθούν στην διαδικασία ανασυνδυασμού. Στη φάση του ανασυνδυασμού, τελεστές ταυτοποίησης και μετάλλαξης

χρησιμοποιούνται για τη δημιουργία νέων ατόμων προκειμένου να εξερευνηθούν το χώρο της λύσης. Τα νεοσύστατα άτομα αντικαθιστούν τα παλιά άτομα, συνήθως τα χειρότερα από αυτά, του πληθυσμού με βάση την καταλληλότητα. Αυτή η διαδικασία επαναλαμβάνεται μέχρι να επιτευχθεί ένα κριτήριο διακοπής, το οποίο μπορεί να είναι ο μέγιστος αριθμός των γενεών ή χρονική προθεσμία.

Στην δική μας περίπτωση, προτείνουμε μια υβριδική προσέγγιση που υβριδοποιεί τον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης με νέες τεχνικές LocalSearch και μία TabuSearch ευρετική για την για την κατάρτιση ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών. Ο ψευδοκώδικα του προτεινόμενου υβριδικού αλγόριθμου και της TabuSearch προσέγγισης, συμβολίζεται HGATS, και παρουσιάζεται παρακάτω. Ο HGATS λειτουργεί σε δύο φάσεις. Στην πρώτη φάση, ο γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης που χρησιμοποιεί γενετικούς τελεστές, μια οργανωμένη στρατηγική αναζήτησης, και δύο ισχυρές τεχνικές LocalSearch, χρησιμοποιείται για να εξελιχθεί ένας πληθυσμός υποψήφιων λύσεων προς όλο και καλύτερες λύσεις, για την εξεύρεση της ιδανικής βέλτιστης λύσης. Συνήθως, οι γενετικοί αλγόριθμοι είναι σε θέση να εντοπίσουν πολλά υποσχόμενες περιοχές για την καθολική βέλτιστη λύση στον χώρο αναζήτησης, αλλά έχουν μερικές φορές δυσκολίες στην εύρεση της ακριβούς βέλτιστης λύσης σε εξαιρετικά προβλήματα περιορισμών. Πολλά παραδείγματα μπορούν να βρεθούν από τη βιβλιογραφία, όπου μία λύση που λαμβάνεται από έναν γενετικό αλγόριθμο βελτιώνεται από μία άλλη τεχνική βελτιστοποίησης. Επίσης θα χρησιμοποιηθεί αυτή η τεχνική στον HGATS, προσπαθώντας να βρει τη βέλτιστη λύση για την κατάρτιση ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών. Λαμβάνοντας υπόψη τη σκληρότητα του ωρολογίου προγράμματος, μόνο αν εφικτές λύσεις που βρέθηκαν κατά τη διάρκεια της πρώτης φάσης του HGATS, η δεύτερη φάση εκτελείται, η οποία χρησιμοποιεί μια ευρετική TabuSearch για την βελτίωση της

εφικτής λύσης προς την βέλτιστη λύση. Παρακάτω περιγράφονται οι δύο φάσεις του HGATS λεπτομερώς.

Algorithm Hybrid Approach—HGATS

```
1: input: A problem instance I
2: set the generation counter  $g := 0$ 
3: for  $i := 1$  to population size do
4:    $s_i \leftarrow \text{InitializeIndividual}(i)$ 
5:    $s_i \leftarrow$  solution  $s_i$  after applying LS operator 1 (LS1)
6:    $s_i \leftarrow$  solution  $s_i$  after applying LS operator 2 (LS2)
7: end for
8: while the termination condition is not reached do
9:   if  $(g \bmod \tau) == 0$  then
10:     apply ConstructMEM() to construct MEM
11:   end if
12:    $s \leftarrow$  child solution by applying
     GuidedSearchByMEM() or Crossover() with a probability  $\gamma$ 
13:    $s \leftarrow$  child solution after mutation with a probability  $P_m$ 
14:    $s \leftarrow$  child solution after applying LS operator 1 (LS1)
15:    $s \leftarrow$  child solution after applying LS operator 2 (LS2)
16:   replace the worst individual of the population by  $s$ 
17:    $g := g + 1$ 
18: end while
19: if  $s$  is an optimal solution then
20:   go to line 24
21: else
22:    $s \leftarrow$  Apply TabuHeuristic() on the best solution obtained
     in the first phase
23: end if
24: output: The best solution  $s_{best}$  achieved for the problem
     instance I
```

Ο ενισχυμένος γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης - Φάση I του HGATS

Η πρώτη φάση του HGATS χρησιμοποιεί τον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης, ο οποίος είναι προσαρμοσμένος και ενισχυμένος σύμφωνα με το πρόβλημα κατάρτισης ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών, να λύσει το πρόβλημα. Το πλαίσιο του γενετικού αλγόριθμου καθοδηγούμενης αναζήτησης βασίζεται σε μια σταθερή κατάσταση γενετικού αλγόριθμου, όταν μία υπό-λύση παράγεται ανά

επανάληψη (ή ανά γενιά) . Ο γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης ξεκινά από έναν αρχικό πληθυσμό των ατόμων ο οποίος δημιουργείται με τυχαίο τρόπο από την *InitializeIndividual()*, όπου τα συμβάντα έχουν ανατεθεί σε αίθουσες και χρονοθυρίδες για κάθε λύση που βασίζεται στην ιδιοκτησία του κάθε συμβάντος. Συνήθως, για τους γενετικούς αλγορίθμους, η ποιότητα των αρχικών λύσεων επηρεάζει τις τελικές λύσεις και οι ερευνητές έχουν δείξει ότι η καλές αρχικές λύσεις παράγουν συνήθως καλά ή απαιτούμενα αποτελέσματα μέσα στον λιγότερο υπολογιστικό χρόνο. Ως εκ τούτου, θέλουμε να δημιουργηθεί ένας καλός αρχικός πληθυσμός, που θα βοηθήσει να εξελιχθεί ο γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης προς τη βέλτιστη λύση γρήγορα. Για το σκοπό αυτό, δύο μέθοδοι τοπικής αναζήτησης εφαρμόζονται σε κάθε άτομο του αρχικού πληθυσμού. Οι μέθοδοι τοπικής αναζήτησης χρησιμοποιούν έξι γειτονικές δομές για να μετακινήσουν αρχικά συμβάντα σε χρονοθυρίδες και στη συνέχεια χρησιμοποιούν τον αντίστοιχο αλγόριθμο για να διαθέσουν αίθουσες και χρονοθυρίδες στα συμβάντα.

Μετά την αρχικοποίηση του πληθυσμού, μια δομή δεδομένων (συμβολίζεται *MEM* εδώ) κατασκευάζεται, η οποία αποθηκεύει μία λίστα από ζευγάρια αιθουσών και χρονοθυρίδων (r, t) για όλα τα συμβάντα στο σύνολο E'_e που έχουν μηδενική ποινή (δηλαδή, να μην παραβιάζουν αυστηρούς και ελαστικούς περιορισμούς σε αυτά τα συμβάντα) των επιλεγμένων καλών ατόμων από τον πληθυσμό. Μετά από αυτό, η *MEM* μπορεί να χρησιμοποιηθεί για καθοδηγήσει την παραγωγή απογόνων για τις επόμενες γενιές. Η δομή δεδομένων *MEM* ανοικοδομείται τακτικά, π.χ., κάθε τ γενιές. Σε κάθε γενιά του γενετικού αλγορίθμου καθοδηγούμενης αναζήτησης, ένα παιδί δημιουργείται για πρώτη φορά είτε με τη χρήση της *MEM* ή με την εφαρμογή του τελεστή crossover, ανάλογα με την πιθανότητα γ . Στη συνέχεια, το παιδί θα υποστεί την λειτουργία μετάλλαξης ακολουθούμενη από τις μεθόδους τοπικής αναζήτησης για δυνατότητες βελτίωσης. Τέλος, το χειρότερο μέλος στον πληθυσμό αντικαθίσταται με το πιο πρόσφατο παιδί που δημιουργήθηκε. Αυτή η επανάληψη

συνεχίζεται έως ότου να επιτευχθεί ένας όρος τερματισμού, π.χ., μία προκαθορισμένη προθεσμία t_{max} επιτυγχάνεται ή η καλύτερη λύση βρέθηκε και δεν έχει καθόλου παράβαση ελαστικού ή αυστηρού περιορισμού.

Αρχικοποίηση του πληθυσμού

Κάθε άτομο I_i του αρχικού πληθυσμού δημιουργείται από τον αλγόριθμο *InitializeIndividual*. Διαιρούμε το σύνολο των συμβάντων E σε δύο κατηγορίες: συμβάντα στην E'_e και συμβάντα που δεν είναι στην E_e . Αν ένα συμβάν δεν έχει κανένα ιδιαίτερο περιορισμό χρονοθυρίδας, καταλαμβάνεται μία τυχαία χρονοθυρίδα t από το σύνολο T'_s των μη περιορισμένων χρονοθυρίδων και μία κατάλληλη αίθουσα. Διαφορετικά, το συμβάν έχει εκχωρηθεί σε μία τυχαία χρονοθυρίδα που αντιστοιχεί σε ένα συμβάν και καταλαμβάνει τυχαία μία αίθουσα από το σύνολο των κατάλληλων αιθουσών.

Με αυτό τον τρόπο, κάθε άτομο που δημιουργείται θα ικανοποιήσει τους αυστηρούς περιορισμούς H2 και H4. Ωστόσο, δεν είναι εγγυημένο ότι είναι εφικτό. Ένα ανέφικτο άτομο θα πρέπει να ελέγχεται από τις ακόλουθες διεργασίες Τοπικής Αναζήτησης, που θα προσπαθήσουν να το καταστήσουν εφικτό.

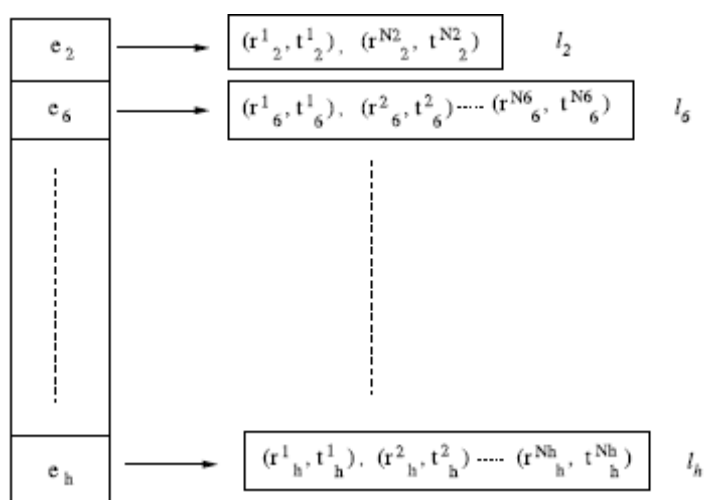
Algorithm *InitializeIndividual*(i)

```
1: input: The index  $i$  of individual  $I_i$ 
2: for each event  $e_j$  of  $I_i$  do
3:   if event  $e_j \in E'_e$  then
4:     assign a random time slot from  $T'_s$  to  $e_j$ 
5:     assign a random room from a list of suitable rooms
6:   else
7:     assign a random time slot from  $ET_j$  to  $e_j$ 
8:     assign a random room from a list of suitable rooms
9:   end if
10: end for
11: output: The generated individual  $I_i$ 
```

Η δομή των δεδομένων MEM

Υπήρξαν μια σειρά από έρευνες στη βιβλιογραφία για την χρησιμοποίηση επιπλέον δομών δεδομένων ή μνήμης για την αποθήκευση χρήσιμων πληροφοριών προκειμένου να ενισχυθεί η απόδοση των γενετικών αλγορίθμων και άλλων μετά-ευρετικών μεθόδων για τη βελτιστοποίηση και την αναζήτηση. Στον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης, μπορούμε επίσης να χρησιμοποιήσουμε μία MEM δομή δεδομένων για την καθοδήγηση της γενιάς των απογόνων με την επανεισαγωγή του καλύτερου μέρους των ατόμων από προηγούμενες γενιές. Αυτή η δομή δεδομένων MEM χρησιμοποιείται για να παράσχει περαιτέρω κατεύθυνση της εξερεύνησης και εκμετάλλευσης στον χώρο αναζήτησης.

Η εικόνα 5 δείχνει τις λεπτομέρειες της δομής δεδομένων MEM, η οποία είναι μια δομή δύο επιπέδων. Το πρώτο επίπεδο είναι μια λίστα των συμβάντων και το δεύτερο επίπεδο είναι μία λίστα l_i από ζεύγη αιθουσών και χρονοθυρίδων που αντιστοιχούν σε κάθε συμβάν e_i από την λίστα του πρώτου επιπέδου. Στην Εικόνα 5, η N_i αντιπροσωπεύει το συνολικό αριθμό των ζευγών στη λίστα δεύτερου επιπέδου l_i .



Εικόνα 5. Αναπαράσταση της Δομής Δεδομένων MEM.

Η δομή δεδομένων MEM ανοικοδομείται τακτικά κάθε τ γενιές. Ο αλγόριθμος *ConstructMEM()* δείχνει το περίγραμμα της κατασκευής της MEM. Όταν η MEM πρόκειται να κατασκευαστεί εκ νέου, πρώτα επιλέγονται $\alpha \times N$ καλύτερα άτομα από τον P πληθυσμό για να σχηματίσουν ένα σύνολο Q , όπου το N δηλώνει το μέγεθος του πληθυσμού. Μετά από αυτό, για κάθε άτομο $I_j \in Q$, ελέγχουμε κάθε συμβάν $e_i \in E'_e$ από την τιμή της ποινής του, δηλαδή, τις παραβιάσεις αυστηρών και ελαστικών περιορισμών που σχετίζονται με αυτό το συμβάν. Αν ένα συμβάν έχει μηδενική τιμή ποινής, τότε αποθηκεύονται οι πληροφορίες που αντιστοιχούν σε αυτό το συμβάν στην MEM. Για παράδειγμα, εάν το συμβάν e_2 ενός ατόμου $I_j \in Q$ έχει ανατεθεί στην αίθουσα 2 στην χρονοθυρίδα 13 και έχει μηδενική τιμή ποινής, τότε προσθέτουμε το ζεύγος (2, 13) στην λίστα l_2 . Ομοίως, τα συμβάντα του επόμενου ατόμου $I_{j+1} \in Q$ επίσης, ελέγχονται για τις τιμές των ποινών τους. Αν το συμβάν e_2 στο I_{j+1} έχει μηδενική ποινή, τότε προσθέτουμε το ζευγάρι της αίθουσας και της χρονοθυρίδας που έχουν ανατεθεί στο e_2 στο I_{j+1} στην υπάρχουσα λίστα l_2 . Αν για ένα συμβάν e_i , δεν υπάρχει λίστα l_i ακόμα, τότε η λίστα l_i προστίθεται στη δομή δεδομένων MEM.

Παρόμοια διαδικασία πραγματοποιείται για τα επιλεγμένα άτομα Q και τελικά η MEM αποθηκεύει ζεύγη αιθουσών και χρονοθυρίδων που αντιστοιχούν σε αυτά τα συμβάντα με μηδενική ποινή των καλύτερων ατόμων του τρέχοντος πληθυσμού. Αυτή η πρόσφατα ανοικοδομημένη MEM δομή δεδομένων χρησιμοποιείται στη συνέχεια για να κατευθύνει την γενιά των απογόνων για τις επόμενες γενιές τ . Ενημερώνουμε την MEM κάθε τ γενιές αντί για κάθε γενιά, προκειμένου να υπάρξει μια ισορροπία μεταξύ της ποιότητας λύσης και του υπολογιστικού το κόστους του γενετικού αλγόριθμου καθοδηγούμενης αναζήτησης.

Algorithm *ConstructMEM()*

- 1: **input:** The whole population P with the population size N
- 2: sort the population P according to the fitness of individuals
- 3: $Q \leftarrow$ select the best $\alpha \times N$ individuals in P
- 4: **for** each individual I_j in Q **do**
- 5: **for** each event ($e_i \in E'_e$) in I_j **do**

```

6:   calculate the penalty value of event  $e_i$  from  $I_j$ 
7:   if  $e_i$  is feasible (i.e.,  $e_i$  has zero penalty) then
8:       add the room and time slot pair  $(r_i, t_i)$  assigned
        to  $e_i$  into the list  $l_i$ 
9:   end if
10:  end for
11: end for
12: output: The data structure MEM

```

Δημιουργώντας ένα παιδί από την οργανωμένη στρατηγική αναζήτησης

Στον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης, ένα παιδί δημιουργείται μέσω της καθοδηγούμενης αναζήτησης από την MEM ή ένας τελεστής διασταύρωσης με μια πιθανότητα γ . Δηλαδή, όταν ένα νέο παιδί είναι να δημιουργηθεί, ένας τυχαίος αριθμός $\rho \in [0,0, 1,0]$ δημιουργείται πρώτα. Αν ο ρ είναι μικρότερος από ότι ο γ , ο αλγόριθμος *GuidedSearchByMEM* () θα χρησιμοποιηθεί για να παράγει το νέο παιδί. Αλλιώς, ένας τελεστής διασταύρωσης όπως φαίνεται στον αλγόριθμο *Crossover* () θα χρησιμοποιηθεί για να δημιουργηθεί το νέο παιδί.

Εάν ένα παιδί πρόκειται να δημιουργηθεί χρησιμοποιώντας τη δομή δεδομένων MEM, επιλέγουμε πρώτα ένα σύνολο E_s του $\beta * |E'_e|$ τυχαίων γεγονότων από το E'_e να δημιουργηθούν από την MEM. Εδώ, το β είναι ένα ποσοστό επί της αξίας και το $|E'_e|$ είναι το μέγεθος του συνόλου E'_e . Επιλέγεται τυχαία ένα ζευγάρι (r_i^j, t_i^j) , $j = 1, \dots, N_i$, από την λίστα l_i που αντιστοιχεί σε ένα συμβάν e_i και εκχωρείται το επιλεγμένο ζεύγος στο e_i για το παιδί. Εάν υπάρχει ένα συμβάν e_i στο E_s , αλλά δεν υπάρχει λίστα l_i στην MEM, τότε τυχαία εκχωρείται μία αίθουσα και μία χρονοθυρίδα από τις πιθανές αίθουσες και χρονοθυρίδες στο e_i για το παιδί. Αυτή η διαδικασία γίνεται για όλα τα συμβάντα στην E_s . Για τα υπόλοιπα συμβάντα που δεν υπάρχουν στην E_s , υπάρχουν καταχωρημένες χρονοθυρίδες και αίθουσες σύμφωνα με τις ιδιαίτερες απαιτήσεις τους για το παιδί.

Εάν ένα παιδί είναι να δημιουργηθεί χρησιμοποιώντας τον τελεστή διασταύρωσης, επιλέγονται πρώτα δύο άτομα από τον τρέχων πληθυσμό όπως οι γονείς από την επιλογή τουρνουά με μέγεθος τουρνουά 2. Στη συνέχεια, ένα παιδί δημιουργείται ως εξής: για κάθε συμβάν, επιλέγεται πρώτα ο γονέας που έχει την μικρότερη τιμή ποινής που αντιστοιχεί σε αυτό το συμβάν, και στη συνέχεια καταλαμβάνει το αντίστοιχο ζευγάρι αίθουσας και χρονοθυρίδας στο συμβάν του παιδιού.

Algorithm *GuidedSearchByMEM()*

1: **input:** The MEM data structure

2: $E_s :=$ randomly select $\beta * |E'_e|$ events from E'_e

3: **for** each event e_i in E_s **do**

4: randomly select a room and time slot pair from the list li

5: assign the selected pair to event e_i for the child

6: **end for**

7: **for** each remaining event e_i not in E_s **do**

8: **if** $e_i \square ET$ **then**

9: assign a particular time slot and suitable room to e_i

10: **else**

11: assign a random time slot and room to e_i

12: **end if**

13: **end for**

14: **output:** A new child generated using MEM

Algorithm *Crossover()*

1: **input:** The current population

2: Select parents $P1$ and $P2$ by the tournament selection

3: **for** each event e_i of the child Ch **do**

4: **if** penalty value of e_i of $P1 <$ penalty value of e_i of $P2$ **then**

5: e_i of $Ch \leftarrow$ the time slot and room allocated to e_i of $P1$

```
6:   else
7:        $e_i$  of  $Ch \leftarrow$  the time slot and room allocated to  $e_i$ 
       of  $P2$ 
8:   end if
9: end for
10:output: A new child generated using crossover
```

Μεταλλαγή

Μετά από ένα παιδί που παράγεται με τη χρήση είτε της MEM ή είτε με crossover, ένας τελεστής μετάλλαξης χρησιμοποιείται με πιθανότητα P_m . Ο τελεστής μετάλλαξης αρχικά επιλέγει τυχαία μία από τις τέσσερις γειτονικές δομές N1, N2, N3, και N4, και στη συνέχεια να κάνει μια κίνηση στο πλαίσιο της επιλεγμένης γειτονικής δομής.

Τοπικές Μέθοδοι Αναζήτησης

Μετά τη λειτουργία μετάλλαξης, δύο φορείς LS (τοπικής αναζήτησης), που συμβολίζονται LS1 και LS2, αντίστοιχα, εφαρμόζονται για την επίλυση του παιδιού για πιθανή βελτίωση. Οι αλγόριθμοι *Local Search Operator 1 (LS1)* και *Local Search Operator 2 (LS2)* συνοψίζουν τα LS1 και LS2 συστήματα, που χρησιμοποιούνται στον προτεινόμενο αλγόριθμο, αντίστοιχα. Το LS1 λειτουργεί σε όλα τα συμβάντα, ενώ το LS2 λειτουργεί σε ένα σύνολο συμβάντων. Εδώ, υποθέτουμε ότι κάθε συμβάν εμπλέκεται με παραβιάσεις ελαστικών και αυστηρών περιορισμών. Οι μέθοδοι LS βασίζονται σε έξι γειτονικές δομές, που συμβολίζονται ως N1, N2, N3, N4, N5, και N6, αντίστοιχα. Αυτές περιγράφονται ως εξής:

- N1: η γειτονιά που ορίζεται από έναν τελεστή που κινεί ένα συμβάν από μια χρονοθυρίδα σε μία άλλη.
- N2: η γειτονιά που ορίζεται από έναν τελεστή που ανταλλάσσει τις χρονοθυρίδες δύο συμβάντων.

- N3: η γειτονιά που ορίζεται από έναν τελεστή που μεταθέτει τρία συμβάντα σε τρεις διακριτές χρονικές περιόδους με έναν από τους δύο πιθανούς τρόπους εκτός από την υπάρχουσα μετάθεση από τα τρία συμβάντα.
- N4: η γειτονιά που ορίζεται από έναν τελεστή που λαμβάνει δύο τυχαία συμβάντα από το σύνολο E'_e και αντικαθιστά τις χρονοθυρίδες τους με τυχαίες από το T'_s .
- N5: η γειτονιά που ορίζεται από έναν τελεστή που λαμβάνει κάθε συμβάν e_i από την λίστα της EE και προσπαθεί να βρει μια θέση στο χρονοδιάγραμμα πριν από όλα τα συμβάντα στην EE_i .
- N6: η γειτονιά που ορίζεται από έναν τελεστή που λαμβάνει ένα υποσύνολο χρονοθυρίδων μεταξύ όλων των κατεχομένων χρονοθυρίδων. Αναμεταξύ του υποσυνόλου αυτού, η χειρότερη χρονοθυρίδα (που περιέχει συμβάντα που συλλογικά έχουν και τη μεγαλύτερη τιμή ποινή) έχει επιλεγεί και τα συμβάντα της μετακινούνται σε άλλη τυχαία επιλεγμένη χρονοθυρίδα στο υποσύνολο.

Algorithm Local Search Operator 1 (LS1)

```

1: input: Individual I from the population
2: for  $i := 1$  to  $n$  do
3:   if event  $e_i$  is infeasible then
4:     if there is untried move left then
5:       calculate the moves: first in N1, then in N2 if N1
       fails, then in N3 if N2 also fails, and finally in
       N4 if N3 also fails
6:       apply the matching algorithm to the time slots
       affected by the move and delta evaluate the result.
7:       if moves reduce hard-constraint violations then
8:         make the moves and go to line 4
9:       end if
10:    end if
11:  end if
12: end for
13: if no any hard-constraint violations remain then
14:   for  $i := 1$  to  $n$  do

```

```

15:         if event  $i$  has soft-constraint violations then
16:             if there is untried move left then
17:                 calculate the moves: first in N1, then in N2
                    if N1 fails, then in N3 if N2 also fails, and
                    finally in N4 if N3 also fails
18:                 apply the matching algorithm to the time slots
                    affected by the move and delta evaluate the
                    result
19:                 if moves reduce soft-constraint violations
                    then
20:                     make the moves and go to line 16
21:                 end if
22:             end if
23:         end if
24:     end for
25: end if
26: output: A possibly improved individual  $I$ 

```

Algorithm Local Search Operator 2 (LS2)

```

1: input: Individual  $I$  after LS1 is applied
2: for each event  $e_i$  of  $I$  in  $EE$  do
3:     for each event  $e_j$  in  $EE_i$  do
4:         try to place event  $e_j$  in the timetable after the time
            slot of  $e_i$  by calculating a move of  $e_i$  in the neighborhood
            N1 and N2
5:         apply the matching algorithm to the time slots affected
            by the move
6:         compute the penalty of  $e_j$  and delta evaluate the result
7:         apply the move if it reduces hard- or soft-constraint
            violations
8:     end for
9: end for
10:  $S :=$  randomly pick a percentage of occupied time slots from  $T$ 
11: for each time slot  $t_i \in S$  do
12:     for each event  $e_j$  in the time slot  $t_i$  do
13:         calculate the penalty value of event  $e_j$ 
14:     end for
15:     sum the total penalty value of events in the time slot  $t_i$ 
16: end for
17: select the time slot  $w_t$  with the biggest penalty value from  $S$ 
18: for each event  $e_i$  in  $w_t$  do
19:     calculate a move of  $e_i$  in the neighbourhood N1
20:     apply the matching algorithm to the time slots affected
        by the move
21:     compute the penalty of  $e_i$  and delta evaluate the result
22: end for
23: if all the moves of events in  $w_t$  together reduce hard- or soft-constraint

```

violations **then**
24: apply the moves
25: **else**
26: delete the moves
27: **end if**
28: **output:** A possibly improved individual **I**

Όπως αναφέρθηκε πριν, το LS1 δουλεύει σε όλα τα συμβάντα και βασίζεται σε δύο βήματα. Στο πρώτο βήμα (γραμμές 2-12 στον αλγόριθμο Local Search Operator 1 (LS1)), το LS1 ελέγχει τις παραβιάσεις σκληρών περιορισμών κάθε συμβάντος αγνοώντας τις παραβιάσεις ελαστικών περιορισμών. Αν υπάρχουν παραβιάσεις αυστηρών περιορισμών για ένα συμβάν, το LS1 προσπαθεί να τους επιλύσει εφαρμόζοντας κινήσεις στις γειτονικές δομές N1, N2, N3, N4 με τη σειρά μέχρι να επιτευχθεί μια προϋπόθεση τερματισμού, π.χ., η βελτίωση επιτυγχάνεται ή ο μέγιστος αριθμός των βημάτων s_{max} επιτυγχάνεται, που έχει οριστεί σε διαφορετικές τιμές για τις περιπτώσεις του προβλήματος της πειραματικής μελέτης.

Μετά από κάθε κίνηση, εφαρμόζεται ο αντίστοιχος αλγόριθμος στις χρονοθυρίδες που επηρεάζονται από την κίνηση και γίνεται προσπάθεια να επιλυθεί η διαταραχή στην κατανομή των αιθουσών και να αξιολογηθεί **DELTA** το αποτέλεσμα της κίνησης (δηλαδή, ο υπολογισμός των παραβιάσεων αυστηρών και ελαστικών περιορισμών πριν και μετά την κίνηση). Αν δεν υπάρχει καμία αδοκίμαστη κίνηση στη γειτονιά για ένα συμβάν, το LS1 συνεχίζει στο επόμενο συμβάν. Μετά την εφαρμογή όλων των γειτονικών κινήσεων σε κάθε συμβάν, αν υπάρχει ακόμη παράβαση αυστηρού περιορισμού, τότε η LS θα σταματήσει. Διαφορετικά, το LS1 θα εκτελέσει το δεύτερο βήμα (γραμμές 13-25 στον αλγόριθμο Local Search Operator 1 (LS1)). Στο δεύτερο βήμα, μετά την επίτευξη μίας εφικτής λύσης, η LS1 μέθοδος χρησιμοποιείται για να ασχοληθεί με ελαστικούς περιορισμούς. Το LS1 εκτελεί μια παρόμοια διαδικασία όπως και στο πρώτο βήμα σε κάθε συμβάν για τη μείωση των παραβιάσεων των ελαστικών περιορισμών του. Για κάθε συμβάν, το LS1 προσπαθεί να κάνει κινήσεις στη γειτονιά N1, N2, N3 και / ή N4 με τη σειρά χωρίς να παραβιάζονται οι αυστηροί

περιορισμοί. Για κάθε κίνηση, ο αντίστοιχος αλγόριθμος εφαρμόζεται για την κατανομή αιθουσών στα επηρεαζόμενα συμβάντα και το αποτέλεσμα είναι αξιολογημένο **DELTA**.

Ο αλγόριθμος Local Search Operator 2 (LS2) περιγράφει το δεύτερο τελεστή LS, LS2, που χρησιμοποιείται στον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης. Ο LS2 λειτουργεί σε ένα σύνολο συμβάντων με N5 (που αντιστοιχούν στις γραμμές 2-9 στον αλγόριθμο Local Search Operator 2 (LS2)) και N6 (που αντιστοιχούν στις γραμμές 10-27 στον αλγόριθμο Local Search Operator 2 (LS2)). Η βασική ιδέα είναι ότι το LS2 προσπαθεί πρώτα να τοποθετήσει ένα συμβάν e_i (που συμμετέχει στον περιορισμό προτεραιότητας H5) σε μια χρονοθυρίδα πριν από την αντίστοιχη λίστα συμβάντων EE_i . Μετά τη μετακίνηση ενός συγκεκριμένου συμβάντος σε μια χρονοθυρίδα στις γειτονικές δομές N1 και N2 κάθε φορά, υπολογίζεται η νέα τιμή ποινή του συμβάντος. Αν η κίνηση μειώνει την τιμή της ποινής, τότε σώζεται, διαφορετικά όχι.

Μετά την εφαρμογή της N5, το LS2 εφαρμόζει την N6. Πρώτα επιλέγει τυχαία ένα ποσοστό χρονοθυρίδων (π.χ., 20%, όπως χρησιμοποιείται εδώ) από το σύνολο των χρονοθυρίδων στο T . Στη συνέχεια, η τιμή ποινής κάθε επιλεγμένης χρονοθυρίδας υπολογίζεται και η χρονοθυρίδα w_i που έχει τη μεγαλύτερη τιμή ποινής επιλέγεται για την τοπική αναζήτηση. Με αυτό τον τρόπο, το LS2 στοχεύει να συμβάλει στη βελτίωση του υπάρχοντος αποτελέσματος. Αφού λαμβάνει τη χειρότερη χρονοθυρίδα, το LS2 προσπαθεί να κάνει μια κίνηση στην γειτονική δομή N1 για κάθε συμβάν της w_i και ελέγχει την τιμή ποινής του κάθε συμβάντος πριν και μετά την εφαρμογή της κίνησης. Αν όλες οι κινήσεις μαζί στην w_i μειώνουν τις παραβιάσεις αυστηρών ή ελαστικών περιορισμών, τότε θα εφαρμόζονται όλες οι κινήσεις. Διαφορετικά, δεν πραγματοποιούνται αυτές οι κινήσεις. Με αυτό τον τρόπο, το LS2 δεν μπορεί μόνο να τοποθετήσει τα συμβάντα σύμφωνα με την προτεραιότητα τους, αλλά και να ελέγξει τη χειρότερη χρονοθυρίδα και να μειώσει την τιμή της ποινής για ορισμένα συμβάντα με το να τα μετακινεί σε άλλες χρονοθυρίδες. Σε γενικές γραμμές, το LS2 αναμένεται να

ενισχύσει τα άτομα του πληθυσμού και να αυξήσει την ποιότητα της εφικτής λύσης, μειώνοντας τον αριθμό των παραβιάσεων των περιορισμών. Όταν τελειώνει το LS2, επιστρέφει ένα πιθανώς καλύτερο και εφικτό άτομο.

Στο τέλος της κάθε γενιάς του γενετικού αλγόριθμου καθοδηγούμενης αναζήτησης, η παραγόμενη child λύση αντικαθιστά το χειρότερο μέλος του πληθυσμού για να κάνει έναν καλύτερο πληθυσμό στην επόμενη γενιά. Με το τέλος της φάσης I, ο γενετικός αλγόριθμος καθοδηγούμενης αναζήτησης μπορεί να παράγει διάφορες βέλτιστες ή σχεδόν βέλτιστες λύσεις.

The tabu search heuristic - Φάση II του HGATS

Προσπαθούμε να βρούμε μία βέλτιστη λύση με τον παραπάνω προτεινόμενο γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης. Ωστόσο, λόγω της αυστηρότητας του PECTP, μετά η πρώτη φάση του HGATS, μερικές φορές μία βέλτιστη ή εφικτή λύση μπορεί να μην επιτευχθεί. Για την περαιτέρω βελτίωση της ποιότητας της λύσης που επιτυγχάνεται με τον γενετικό αλγόριθμο καθοδηγούμενης αναζήτησης, ένα απλό ευρετικό TS *TabuHeuristic* (), το οποία παρουσιάζεται στον Αλγόριθμος *TabuHeuristic()*—Phase II of HGATS, εφαρμόζεται ως η δεύτερη φάση του HGATS με την ελπίδα να επιτευχθεί μια βελτιωμένη και εφικτή λύση που έχει ήδη επιτευχθεί από την καλύτερη λύση της φάσης I. Η ευρετική αναζήτηση είναι ένα είδος ευριστικών μεθόδων, οι οποίες έχουν το πλεονέκτημα ότι διαθέτουν εσωτερική μνήμη. Αυτή η εσωτερική μνήμη αποτρέπει την ευρετική αναζήτηση από την αναθεώρηση περιοχών του χώρου αναζήτησης που επισκοπήθηκαν νωρίτερα. Ως εκ τούτου, είναι πιο εύκολο να ξεφύγουν από την τοπική βέλτιστη λύση και να προσεγγίσουν την καθολική ή σχεδόν καθολική βέλτιστη λύση σε σύντομο χρονικό διάστημα. Η TS είναι γνωστό πως είναι συνήθως ένα ισχυρό εργαλείο για όλους τύπους προβλημάτων χρονοπρογραμματισμού.

Ελέγχουμε πρώτα την καλύτερη λύση που λαμβάνεται από την πρώτη φάση. Εάν είναι η βέλτιστη, η Φάση II δεν θα εκτελεστεί. Διαφορετικά, εάν πρόκειται για εφικτή λύση, τότε βελτιώνουμε την λύση εφαρμόζοντας το TS heuristic. Εάν η λύση δεν είναι εφικτή, πρώτα αφαιρούνται όλα τα συμβάντα που περιλαμβάνουν παραβιάσεις αυστηρών περιορισμών και τους επανεξετάζει, εάν και μόνο εάν πληρούν όλους τους αυστηρούς περιορισμούς κατά τη διάρκεια της γειτονικής αναζήτησης.

Εφαρμόζουμε τις N1, N2, N4 και ως γειτονικές δομές για τη μετακίνηση μίας λύσης. Μια κίνηση μίας λύσης ορίζεται από την μετακίνηση ένα τυχαίου συμβάντος της λύσης χρησιμοποιώντας την N1, εναλλάσσοντας δύο τυχαία συμβάντα της λύσης χρησιμοποιώντας την N2, ή εναλλάσσοντας δύο συγκεκριμένα συμβάντα του διαλύματος σε χρονοθυρίδες χρησιμοποιώντας την N4, με τη σειρά. Ο λόγος για τη μη εφαρμογή των N5 και N6 στην κίνηση έγκειται στο ότι η χρήση των N5 και N6 παίρνει χρόνο και επιπλέον εργασία για την άρση μίας παραβίασης αυστηρού περιορισμού.

Μια κίνηση είναι μια tabu move αν τουλάχιστον ένα από τα εμπλεκόμενα συμβάντα έχει μετακινηθεί λιγότερο από l βήματα πριν, όπου l είναι το μήκος της tabu list TL . Το μήκος της tabu list έχει οριστεί από τον αριθμό των συμβάντων που χωρίζονται από μια σταθερά K ($K = 100$). Προκειμένου να μειωθεί η πιθανότητα δημιουργίας κύκλων των κινήσεων και να ενισχυθεί η εξερεύνηση, ένα μεταβλητό σύνολο γειτονιών εφαρμόζεται, όπου κάθε κίνηση χρησιμοποιεί την γειτονιά N1, N2, ή N4 με πιθανότητα 0,1.

Για να διερευνήσουν το χώρο αναζήτησης πιο αποτελεσματικά, δεχόμαστε μία tabu move εάν βελτιώνει την καλύτερη λύση μέχρι στιγμής. Εν περιλήψει, το TS heuristics θεωρεί ένα μεταβλητό σύνολο γειτονιών και πραγματοποιεί και την καλύτερη κίνηση που βελτιώνει την καλύτερη λύση μέχρι στιγμής. Αλλιώς, εκτελεί την καλύτερη μη- tabu move που επιλέγεται μεταξύ εκείνων που ανήκουν στο τρέχον σύνολο μεταβλητών των γειτόνων. Το TS heuristics

συνεχίζεται μέχρι να έχει επιτευχθεί ένα χρονικό όριο ή η καλύτερη μέχρι τώρα λύση δεν έχει παραβιάσεις ελαστικών και αυστηρών περιορισμών (δηλαδή, μία βέλτιστη λύση επιτυγχάνεται).

Algorithm TabuHeuristic()—Phase II of HGATS

```
1: input: The best solution  $s_{best}$  from Phase I (GSGA)
2:  $s \leftarrow s_{best}$ 
3: if  $s$  is not feasible then
4:     remove all events that involve hard-constraint violations
5: end if
6:  $TL \leftarrow \emptyset$ 
7: while the termination condition is not reached do
8:     for  $i := 0$  to 10% of the neighbours do
9:          $s_i \leftarrow s$  after the  $i$ th move
10:        compute the objective value  $f(s_i)$ 
11:    end for
12:    if  $\exists s_j | f(s_j) < f(s)$  and  $f(s_j) \leq f(s_i) \forall i$  then
13:         $s \leftarrow s_j$ 
14:         $TL \leftarrow TL \cup E_i$  where  $E_i$  is the set of events moved
            to get  $s_j$ 
15:    else
16:         $s \leftarrow$  the best non-tabu moves among all  $s_i$ 
17:         $TL \leftarrow TL \cup E_b$  where  $E_b$  is the set of events moved
            by the best non-tabu move
18:    end if
19:     $s_{best} \leftarrow$  the best solution so far
20: end while
21: output: The optimised solution  $s_{best}$ 
```

2. Μη-κυριαρχούμενος Γενετικός Αλγόριθμος Ταξινόμησης II με μεταβαλλόμενο μέγεθος πληθυσμού - (Non-dominated Sorting Genetic Algorithm—II with a variable population size - NSGA-II VPS)

Περιγραφή του προβλήματος

Το πρόβλημα χρονοπρογραμματισμού μαθημάτων πανεπιστήμιου θεωρείται ότι είναι ένας ειδικός τύπος προβλήματος βελτιστοποίησης που παίζει ένα σημαντικό ρόλο στα ιδρύματα που είναι βασισμένα στην εκπαίδευση. Αυτό το πρόβλημα μπορεί να οριστεί ως η ανάθεση ενός συνόλου διαλέξεων σε χρονοθυρίδες και αίθουσες που υπόκεινται σε μια ποικιλία από αυστηρούς και ελαστικούς περιορισμούς. Οι αυστηροί περιορισμοί αποτελούν μια απόλυτη απαίτηση. Ένα χρονοδιάγραμμα το οποίο ικανοποιεί τους σκληρούς περιορισμούς είναι γνωστό ως μία εφικτή λύση. Στην περίπτωσή μας, οι αυστηροί περιορισμοί παραμένουν ίδιοι, ενώ προστέθηκε ένας νέος ελαστικός περιορισμός στους αρχικούς περιορισμούς. Οι επόμενες παράγραφοι περιγράφουν το αρχικό πρόβλημα και τον νέο ελαστικό περιορισμό με λεπτομέρεια.

A. Αρχικό πρόβλημα

Οι ακόλουθοι αυστηροί περιορισμοί παρουσιάζονται ως εξής:

H1: Κανένας μαθητής μπορεί να καταχωρηθεί σε περισσότερες από μία διάλεξη την ίδια στιγμή.

H2: Η αίθουσα θα πρέπει να πληροί τα χαρακτηριστικά που απαιτούνται από την διάλεξη.

H3: Ο αριθμός των φοιτητών που παρακολουθούν το μάθημα θα πρέπει να είναι μικρότερος ή ίσος με τη χωρητικότητα της αίθουσας.

H4: Όχι περισσότερες από μία διάλεξη επιτρέπονται σε μία χρονοθυρίδα σε κάθε αίθουσα.

Ελαστικοί περιορισμοί που εξίσου παίρνουν ποινή έχουν ως εξής:

S1: Ένας φοιτητής έχει μόνο μία διάλεξη σε μια μέρα.

S2: Ένας φοιτητής έχει περισσότερες από 2 συνεχόμενες διαλέξεις.

S3: Ένας φοιτητής έχει μία διάλεξη προγραμματισμένη στην τελευταία χρονοθυρίδα της ημέρας.

Το πρόβλημα έχει:

- Σύνολο (E) διαλέξεων, $e = \{e_1, \dots, e_E\}$
- Σύνολο χρονοθυρίδων, 45 χρονοθυρίδες (T).
- Σύνολο (R) αιθουσών.
- Σύνολο (F) χαρακτηριστικά αιθουσών
- Σύνολο (St) φοιτητές.

Ο στόχος είναι να ικανοποιηθούν οι αυστηροί περιορισμοί και να ελαχιστοποιηθεί η παραβίαση των ελαστικών περιορισμών. Στις καταστάσεις στον πραγματικό κόσμο, είναι συνήθως αδύνατο να ικανοποιηθούν όλοι ελαστικοί περιορισμοί, αλλά ελαχιστοποιώντας τις παραβιάσεις των ελαστικών περιορισμών αντιπροσωπεύει αύξηση στην ποιότητα της λύσης.

Ο Πίνακας 1 δείχνει τα κύρια χαρακτηριστικά για τις περιπτώσεις:

Table 1: Dataset features

	Small	Medium	Large
Number of events (E)	100	400	400
Number of rooms (R)	5	10	10
Number of features (F)	5	10	10
Approx features per room	3	3	5
Percent feature use	70	80	90
Number of students (St)	80	200	400
Max events per student	20	20	20
Max students per event	20	50	100

B. Νέος Ελαστικός Περιορισμός

Ο πρόσθετος ελαστικός περιορισμός (κωδικοποιείται ως **S4**) που παρουσιάζεται εδώ ορίζεται ως ο αριθμός των χρονοθυρίδων αναμονής μεταξύ δύο επάλληλων κύκλων για όλους τους μαθητές μέσα στην ίδια μέρα. Για παράδειγμα, αν ένας μαθητής έχει δύο μαθήματα, σε μια μέρα όπως την χρονοθυρίδα 1 (κωδικοποιείται ως T1) και την χρονοθυρίδα 8 (κωδικοποιείται ως T8). Αυτό σημαίνει ότι οι χρονοθυρίδες μεταξύ των T1 και T8 (δηλ. T2, T3, T4, T5, T6 και T7) θεωρούνται ως χρονοθυρίδες αναμονής. Έτσι, η παραβίαση του ελαστικού περιορισμού **S4** είναι 6. Η μείωση των χρονοθυρίδων αναμονής, θα οδηγήσει σε ένα πιο συμπαγές χρονοδιάγραμμα, το οποίο με τη σειρά του οδηγεί σε υψηλότερης ποιότητας χρονοδιαγράμματα καθώς οι μαθητές είναι σε θέση να παρακολουθήσουν διαλέξεις σε μπλοκ, σε αντίθεση με το να περιμένουν όλη μέρα άσκοπα. Ωστόσο, η εισαγωγή του περιορισμού αυτού συγκρούεται με τους αρχικούς ελαστικούς περιορισμούς, όπου κάθε φοιτητής δεν θα πρέπει να έχει περισσότερες από δύο επάλληλες διαλέξεις (βλ. ελαστικός περιορισμός S2). Για να λυθεί αυτό το πρόβλημα, ο πρόσθετος περιορισμός πρέπει να εφαρμόζεται προσεκτικά για να εξασφαλιστεί ότι επιτυγχάνεται μια ισορροπία μεταξύ των αρχικών ελαστικών περιορισμών και του νέου.

Οι χρονοθυρίδες ταξινομούνται σε "καλές" και "κακές" χρονοθυρίδες σε σχέση με ένα φοιτητή που παρακολουθεί διαλέξεις σε μια μέρα. Για παράδειγμα, στον Πίνακα 2 ο φοιτητής A (Student A) έχει 5 χρονοθυρίδες με καμία τάξη, δηλ. T3, T5, T6, T7 και T8. Η χρονοθυρίδα T3 θεωρείται "καλή" χρονοθυρίδα επειδή βρίσκεται μετά από δύο διαδοχικές διαλέξεις, ενώ οι T5, T6, T7 και T8 είναι θεωρούνται ως "κακές" χρονοθυρίδες επειδή ο φοιτητής A έχει να περιμένει 4 χρονοθυρίδες μεταξύ των τάξεων στις T4 και T9 (όπως φαίνεται στον Πίνακα 2).

Ο Πίνακας 3 παρουσιάζει τον αριθμό των "καλών" και "κακών" χρονοθυρίδων για όλους τους τρεις φοιτητές. Για παράδειγμα ο φοιτητής B (Student B) δεν έχει καθόλου "καλές"

χρονοθυρίδες και έχει τέσσερις "κακές". Το σύνολο των «κακών» χρονοθυρίδων θεωρούνται ως παραβίαση του S4. Ως εκ τούτου, η αξιολόγηση του νέου ελαστικού περιορισμού (S4) είναι 11 για τους φοιτητές A, B και Γ (A,B,C). Η συνολική παραβίαση των ελαστικών περιορισμών (S1, S2, S3 και S4) είναι 12 (δηλ. 1+11).

Ο Πίνακας 4 δείχνει ένα παράδειγμα μίας βέλτιστης λύσης. Για παράδειγμα, η διάλεξη για τον φοιτητή A (Student A) που είχε προηγουμένως ανατεθεί στην χρονοθυρίδα T9 (βλέπε πίνακα 2) είναι τώρα ανατιθέμενο στην χρονοθυρίδα T5 (βλέπε Πίνακα 4), που μειώνει τις «κακές» χρονοθυρίδες για τον φοιτητή A στο μηδέν. Αυτή η διάταξη παρέχει μια καλύτερη λύση, στα οποία η συνολική παραβίαση των ελαστικών περιορισμών (S1, S2, S3 και S4) είναι τώρα 2.

Table 2: Illustrated example of student matrix

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Student A	1	1	0	1	0	0	0	0	1
Student B	1	0	0	0	1	1	0	0	1
Student C	0	0	0	0	1	0	0	1	0

Table 3: Computing the free good/bad timeslots based on Table 2

	Student A	Student B	Student C	Total
Total "good" timeslots	1	0	0	1
Total "bad" timeslots	4	5	2	11

Table 4: Optimal representation of timetable

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
Student A	1	1	0	1	1	0	0	0	0
Student B	1	1	0	1	1	0	0	0	0
Student C	0	0	0	0	1	1	0	0	0

Table 5: Computing the free good/bad timeslots based on table 4

	Student A	Student B	Student C	Total
Total "good" timeslots	1	1	0	2
Total "bad" timeslots	0	0	0	0

Ορίζουμε την αντικειμενική συνάρτηση ως ένα άθροισμα του αρχικών ελαστικών περιορισμών και του νέου ελαστικού περιορισμού ως εξής:

$$\text{Min} \sum_{i=1}^{St} (S1 + S2 + S3 + S4)$$

Non-dominated Sorting Genetic Algorithm–II with a variable population size – (NSGA-II VPS)

Ο Μη-κυριαρχούμενος Γενετικός Αλγόριθμος Ταξινόμησης (NSGA) αναπτύχθηκε για πρώτη φορά το 1998 από τον Srinivas και Deb. Αργότερα, ο Deb βελτίωσε τον πρωτότυπο NSGA και τον ονόμασε NSGA-II. Η κύρια διαφορά είναι ότι NSGA-II χρησιμοποιεί ένα αρχείο για να κρατήσει τις καλύτερες λύσεις και έναν κανόνα για την προσαρμογή αναθέσεων που λαμβάνει υπόψη τόσο τον βαθμό όσο και την απόσταση της κάθε λύσης. Αυτή η ενότητα θα αναλύσει την προτεινόμενη προσέγγιση (NSGA-II VPS), λειτουργίες crossover / μετάλλαξης και γειτονικές δομές που χρησιμοποιούνται για να λύσουν το πρόβλημα.

A. NSGA-II VPS

Σε αυτό την περίπτωση, έχουμε τροποποιήσει τον NSGA-II για να τρέξει με ένα μεταβλητό μέγεθος πληθυσμού και όχι ένα σταθερό μέγεθος πληθυσμού. Κάθε άτομο έχει μια διάρκεια ζωής, η οποία αξιολογείται με βάση την ποιότητα του ατόμου και τα χαρακτηριστικά του πληθυσμού (ο μέσος όρος από τις ποιότητες λύσης στον πληθυσμό) κατά τη στιγμή της δημιουργία του. Η γενική δομή του προτεινόμενου NSGA-II VPS δείχνεται στην Εικόνα 6. Σε κάθε γενιά, ένα διαφορετικό μέγεθος του πληθυσμού αξιολογείται με βάση τα ελάχιστα

και τα μέγιστα όρια του μεγέθους του πληθυσμού. Κάθε άτομο στον πληθυσμό έχει τη δική του διάρκεια ζωής του. Ένα άτομο παραμένει στο σύνολο του πληθυσμού κατά τη διάρκεια της ζωής του και θα αφαιρεθεί όταν η ηλικία του υπερβαίνει τη διάρκεια της ζωής του. Ο διαδικασία του ζευγαρώματος για την επιλογή των γονέων βασίζεται στην μέθοδο Επιλογής με Τροχό Ρουλέτας (Roulette Wheel Selection - RWS) με σταθερά ποσοστά crossover και τα μετάλλαξης. Η NSGA-II VPS διαδικασία απεικονίζεται στην Εικόνα 7, πράγμα που αποδεικνύει ότι ο αλγόριθμος αυτός δουλεύει μέσα σε μια γενιά (δηλαδή εντός t και $t + 1$ γενιάς). Από τότε που την υπολογιστική ταχύτητα για Γενετικούς Αλγόριθμους συνήθως εξαρτάται από το μέγεθος του πληθυσμού, ένα αυτό-ρυθμιζόμενο μέγεθος του πληθυσμού εφαρμόζεται ανάμεσα στα ελάχιστα και μέγιστα όρια μέσα στον NSGA-II.

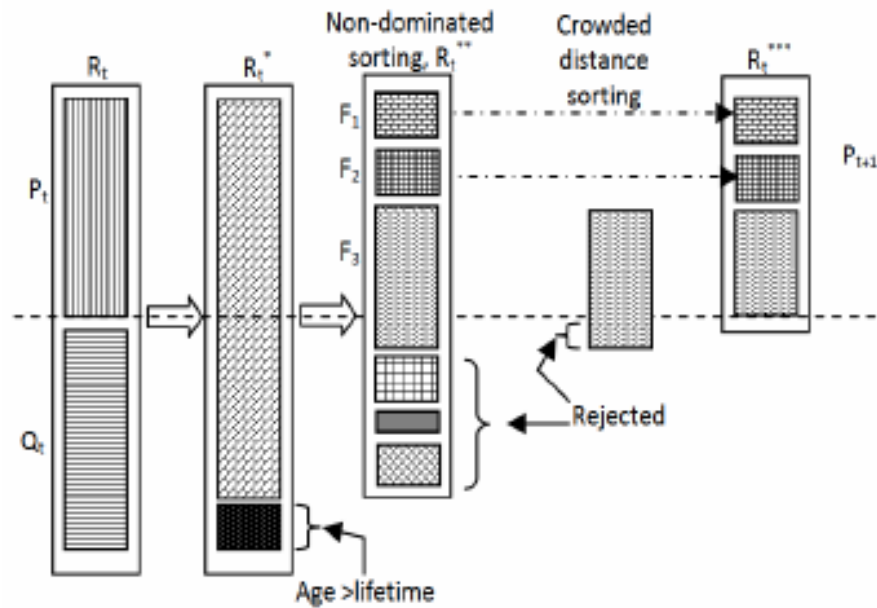
```

Parameters:
Pt: Parent population
Qt: Offspring population
N: Maximum population size
H: Size of archive pool
PAt: Population Archive with size H
tmax: Maximum number of generation
Reset generation counter , t=0
MinLT: Minimum lifetime
MaxLT: Maximum lifetime
Reset age for each individuals, age=0

Begin
Generate initial population Pt.
Calculate lifetime for each individuals in Pt
While (t < tmax)
{
    Increase the age for each individual by 1;
    Update archive PAt [Best individuals from Pt];
    Qt =Mating Pool [generate new offspring based on crossover and
    mutation rates while satisfying hard constraints];
    Evaluate the violations of soft constraints for the new offspring;
    Calculate the lifetime for each offspring in Qt with minimum
    lifetime (MinLT) =1
    Rt = Pt ∪ Qt [Combine parents and offspring's solutions].
    Remove all individual from Rt if the age of the individual is
    greater than their life time based on ( MaxLT ) to obtain Rt*;
    Perform nondominated sorting to Rt* to obtain Rt**;
    Perform sorted crowded distance on Rt** to obtain Rt***.
    Pt+1 = Rt***
    Increase the generation counter, t=t+1.
}

```

Εικόνα 6: Ο NSGA-II με μεταβλητό μέγεθος πληθυσμού (NSGA-II VPS)



Εικόνα 7: Η διαδικασία του NSGA-II VPS

Πρώτον, ο NSGA-II-VPS δημιουργεί έναν νέο απόγονο βασισμένο σε τιμές διασταύρωσης (crossover) και μετάλλαξης να αποκτήσει Q_t . Η ποιότητα του νέου απογόνου αξιολογείται με βάση την αντικειμενική συνάρτηση όπως περιγράφεται στην παράγραφο 2. Όλα τα άτομα συνδυάζονται ($P_t \cup Q_t$), και τα άτομα που υπερβαίνουν τη διάρκεια ζωής τους αφαιρούνται από το σύνολο του ζευγαρώματος για να αποκτήσουν R_t . Αυτή η διαδικασία οδηγεί στην ανανέωση του συνόλου ζευγαρώματος με νέα άτομα να ενισχύουν την ικανότητα της εξερευνητικής διαδικασίας. Η αφαίρεση ατόμων από το σύνολο του ζευγαρώματος γίνεται υπό έναν όρο ελάχιστου μεγέθους πληθυσμού (δηλαδή που ελέγχεται από ένα ελάχιστο δεσμευμένο μέγεθος πληθυσμού). Μία μη κυριαρχούμενη διαλογή με βάση την ποιότητα της λύσης εφαρμόζεται για να ληφθεί R_t και μια ταξινομημένη συνωστισμένη απόσταση γίνεται στο R_t . Το σύνολο του ζευγαρώματος που είναι να εξεταστεί στην επόμενη γενιά βασίζεται στα άτομα στην R_t . Η επιλογή βασίζεται σε μη κυριαρχούμενη σειρά ταξινόμησης, δηλαδή εάν δύο λύσεις έχουν την ίδια ποιότητα, τότε η επιλογή ταξινόμησης συνωστισμένης απόστασης χρησιμοποιείται για να επιλέξει την καλύτερη. Ο αρχείο πληθυσμού με το

μέγεθος H , χρησιμοποιείται για να κρατήσει τα καλύτερα άτομα μέσω μιας διαδικασίας βελτιστοποίησης.

B. Διαδικασία Διασταύρωσης και Μετάλλαξης (Crossover and Mutation process)

Η διασταύρωση και η μετάλλαξη θεωρούνται η καρδιά ενός γενετικού αλγόριθμου. Στην εργασία αυτή, κάθε χρωμόσωμα αντιπροσωπεύει ένα εφικτό χρονοδιάγραμμα. Η αναπαράσταση ενός χρονοδιαγράμματος απεικονίζεται στην Εικόνα 8, η οποία θεωρείται ένας απλός τρόπος να εκτελεστεί η διασταύρωση και η μετάλλαξη. Για παράδειγμα, οι e_{14} , e_{17} , e_1 προγραμματίζονται στην χρονοθυρίδα T_1 στις αίθουσες R_1 , R_2 και R_{11} , αντίστοιχα. Αυτό αντιπροσωπεύει ένα γονίδιο του χρωμοσώματος.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
R_1	e_{14}			e_{19}	e_7		e_{16}	e_{11}	e_{12}
R_2		e_6	e_3	e_4					e_{18}
R_3		e_7		e_{15}				e_{20}	
.			e_5					e_{21}	
.	e_{17}								
.		e_2	e_9	e_8		e_{13}			
R_{11}	e_1			e_{10}					

Εικόνα 8: Αναπαράσταση Χρωμοσώματος

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
R_1	e_{18}				e_{10}	e_{17}			e_{13}
R_2		e_6	e_3	e_4	e_1			e_{19}	
R_3	e_{14}	e_7							e_{12}
.		e_2	e_5	e_{12}		e_9		e_{16}	e_1
.							e_{15}		
.	e_{20}		e_8						
R_{11}	e_{21}				e_{11}				

Parent 1

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
R_1	e_{14}				e_{19}	e_7		e_{16}	e_{11}
R_2		e_6	e_3	e_4	e_1				e_{18}
R_3		e_7		e_{15}			e_{21}	e_{20}	
.			e_5						
.									
.		e_2	e_9	e_8		e_{13}			
R_{11}	e_1			e_{10}					e_{17}

Parent 2

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
R_1	e_{18}	e_{11}			e_{10}				e_{13}
R_2		e_6	e_3	e_4	e_1			e_{19}	
R_3	e_{14}	e_7							e_{12}
.		e_2	e_5	e_{12}		e_9		e_{16}	e_1
.							e_{15}		
.	e_{20}		e_8						
R_{11}	e_{21}	e_{17}			e_{11}				

Offspring 1

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
R_1	e_{14}				e_{19}	e_7		e_{16}	e_{11}
R_2		e_6	e_3	e_4	e_1				e_{18}
R_3		e_7		e_{15}			e_{21}	e_{20}	
.			e_5						
.									
.		e_2	e_9	e_8		e_{13}			
R_{11}	e_1			e_{10}					e_{17}

Offspring 2

Εικόνα 9: Αναπαράσταση Χρωμοσώματος μετά την Διασταύρωση

Η λειτουργία διασταύρωσης όπως απεικονίζεται στην Εικόνα 9. Ο σκιασμένες χρονοθυρίδες, δηλ. T_2 και T_8 αντιπροσωπεύουν τις επιλεγμένες χρονοθυρίδες για τη λειτουργία διασταύρωσης (crossover). Οι χρονοθυρίδες επιλέγονται με βάση ένα ποσοστό crossover χρησιμοποιώντας τη μέθοδο επιλογής με τροχό ρουλέτας (RWS). Αυτός ο μηχανισμός crossover πραγματοποιείται υπό δύο προϋποθέσεις για την εξασφάλιση της εφικτής παραγωγής απογόνων:

- 1) Η διάλεξη μπορεί να μετακινηθεί μόνο εάν η αντίστοιχη χρονοθυρίδα είναι άδεια. Για παράδειγμα, η e_6 μπορεί να μετακινηθεί από τη χρονοθυρίδα T_2 στον Γονέα 1 στην χρονοθυρίδα T_8 στον Απόγονο 2 (βλέπε Εικόνα 9).
- 2) Δεν υπάρχουν συγκρούσεις μεταξύ της μετακινηθείσας διάλεξης και των προγραμματισμένων διαλέξεων. Για παράδειγμα, η μετακινημένη διάλεξη (e_2) μπορεί να μετακινηθεί από τη χρονοθυρίδα T_2 στην T_8 χωρίς συγκρούσεις με τα προγραμματισμένα μαθήματα (e_{11} , e_{17} και e_{20}), όπως φαίνεται στην Εικόνα 9.

Η μετάλλαξη χρησιμοποιείται για να ενισχύσει την απόδοση της λειτουργίας crossover στο να επιτρέπει να διερευνηθεί ένας μεγάλος χώρος αναζήτησης. Η μετάλλαξη γίνεται με την τυχαία επιλογή γειτονικών δομών (**NH1** ή **NH2**) με βάση έναν ρυθμό μετάλλαξης.

C. Γειτονικές Δομές (Neighbourhood Structures)

Σε αυτή την εργασία, έχουν εφαρμοστεί δύο τύποι γειτονικών δομών έχουν στα πειράματά μας, δηλαδή:

NH1: Επιλογή δύο διαλέξεων τυχαία και ανταλλαγή χρονοθυρίδων.

NH2: Επιλογή μίας διάλεξης στην τύχη και μετατόπιση σε μια νέα τυχαία εφικτή χρονοθυρίδα.

Υπολογισμοί Διάρκειας Ζωής (Lifetime Calculations)

Κατά τη γέννηση ενός ατόμου, η διάρκεια ζωής αξιολογείται. Τρεις στρατηγικές είναι προτεινόμενες: αναλογική στρατηγική, γραμμική στρατηγική και διπλό-γραμμική στρατηγική. Στον NSGA-II VPS αλγόριθμο, επιλέγουμε την διπλό-γραμμική στρατηγική, διότι αυτή η στρατηγική είναι σε θέση να ξεπεράσει τα μειονεκτήματα των αναλογικών και γραμμικών στρατηγικές. Η διπλό-γραμμική στρατηγική υπολογίζεται ως εξής:

$$\begin{cases} \text{MinLT} + \eta \frac{\text{fitness}[i] - \text{MinFit}}{\text{AvgFit} - \text{MinFit}} & \text{if } \text{Fitness}[i] \geq \text{Avgfit} \\ \frac{1}{2}(\text{MinLT} + \text{MaxLT}) + \eta \frac{\text{fitness}[i] - \text{MinFit}}{\text{AvgFit} - \text{MinFit}} & \text{if } \text{Fitness}[i] < \text{Avgfit} \end{cases}$$

Όπου :

$$\eta = \text{MaxLT} - \text{MinLT}.$$

Ο κύριος λόγος της χρήσης μιας διάρκειας ζωής για κάθε άτομο στο σύνολο του πληθυσμού είναι για να εξασφαλιστεί ότι μόνο οι λύσεις καλής ποιότητας θα διατηρούνται πάντα στο σύνολο του πληθυσμού. Οι λύσεις όπου η ποιότητα δεν έχει βελτιωθεί κατά τη διάρκεια της διαδικασίας βελτιστοποίησης θα απομακρύνονται από τον πληθυσμό πίσω. Εκτιμάται ότι αυτό θα βοηθήσει τον αλγόριθμο να εξερευνά καλύτερα το χώρο της λύσης, καθ' όλη τη διαδικασία βελτιστοποίησης.

3. Ένας Μεταευρετικός Αλγόριθμος Βασισμένος στο Χρόνο (A time-dependent metaheuristic algorithm .)

1. Περιγραφή του προβλήματος και συναρτήσεις κόστους

Ένα παράδειγμα προβλήματος περιλαμβάνει τις ακόλουθες πληροφορίες:

- Ένα σύνολο από n συμβάντα που πρόκειται να προγραμματιστούν σε 45 χρονοθυρίδες (να ερμηνευθεί ως εννέα χρονοθυρίδες 1-ώρας την ημέρα, για πέντε ημέρες).
- Ένα σύνολο από m αίθουσες όπου οι εκδηλώσεις θα λάβουν χώρα, που η καθεμία έχει μια συγκεκριμένη χωρητικότητα.
- Ένα σύνολο από *χαρακτηριστικά* αιθουσών που απαιτούνται από τα συμβάντα και ικανοποιούνται από τις αίθουσες.
- Ένα σύνολο από s φοιτητές που φοιτούν σε διάφορους συνδυασμούς των συμβάντων.
- Ένα σύνολο από διαθέσιμες χρονοθυρίδες για κάθε ένα από τα συμβάντα n (δηλαδή, δεν είναι όλα τα συμβάντα διαθέσιμα για να προγραμματιστούν σε όλες τις χρονοθυρίδες).
- Ένα σύνολο από απαιτήσεις προτεραιότητας δηλώνοντας ότι ορισμένα γεγονότα θα πρέπει να συμβούν πριν / μετά από άλλα στο χρονοδιάγραμμα.

Λαμβάνοντας υπόψη αυτές τις πληροφορίες, ο στόχος είναι να εκχωρηθούν όλα τα συμβάντα n σε μία αίθουσα και μια χρονοθυρίδα, ενώ υπακούουν τους ακόλουθους πέντε αυστηρούς περιορισμούς:

HC_1 : Κανένας φοιτητής θα πρέπει να υποχρεωθεί να παρακολουθήσει περισσότερα από ένα συμβάντα σε μια συγκεκριμένη χρονοθυρίδα.

HC_2 : Κάθε συμβάν θα πρέπει να ανατεθεί σε μία αίθουσα που έχει αρκετές θέσεις για όλους τους συμμετέχοντες φοιτητές και το οποίο πληροί όλα τα χαρακτηριστικά αιθουσών που απαιτούνται από το συμβάν.

HC_3 : Μόνο ένα συμβάν θα πρέπει να ανατεθεί σε κάθε αίθουσα, σε κάθε χρονική στιγμή (δηλαδή όχι διπλές εκχωρήσεις αιθουσών).

HC₄: Τα συμβάντα θα πρέπει να ανατεθούν σε χρονοθυρίδες που έχουν χαρακτηριστεί ως "διαθέσιμες" για αυτά τα συμβάντα.

HC₅: Όπου αναφέρονται, τα συμβάντα θα πρέπει να παρουσιαστούν να προκύψουν με τη σωστή σειρά στην εβδομάδα.

Εκτός από αυτούς τους πέντε αυστηρούς περιορισμούς, είναι επίσης επιθυμητό να ικανοποιούνται οι ακόλουθοι τρεις ελαστικοί περιορισμοί:

SC₁: Οι φοιτητές δεν θα πρέπει να προγραμματιστούν να παραστεί σε ένα συμβάν κατά την τελευταία χρονοθυρίδα της ημέρας (που σημαίνει, χρονοθυρίδες 9, 18, 27, 36, ή 45).

SC₂: Οι φοιτητές δεν θα πρέπει να παρακολουθήσουν συμβάντα σε τρεις ή περισσότερες διαδοχικές χρονοθυρίδες που προκύπτουν την ίδια ημέρα.

SC₃: Οι φοιτητές δεν θα πρέπει να υποχρεούνται να παρακολουθήσουν μόνο ένα συμβάν σε μια συγκεκριμένη ημέρα.

Όπως είναι τυπικό στην έρευνα χρονοπρογραμματισμού, για αυτό το πρόβλημα η ικανοποίηση των αυστηρών περιορισμών θεωρείται ότι είναι πιο σημαντική από την ικανοποίηση των ελαστικών περιορισμών. Εξαιτίας αυτού, οι κανόνες του διαγωνισμού δηλώνουν ότι μια υποψήφια λύση θα πρέπει να αξιολογείται σύμφωνα σε δύο ξεχωριστές τιμές: η Απόσταση από το Εφικτό (*Distance to Feasibility*) και το Ήπιο Κόστος (*Soft Cost*). Η Απόσταση από το Εφικτό χρησιμοποιείται επειδή -σε αντίθεση με το πρώτο διαγωνισμό- είναι αναγκαίο να επιτραπεί για το γεγονός ότι ένας αλγόριθμος μπορεί να μην είναι σε θέση να αναθέσει το σύνολο των n συμβάντων σε χρονοθυρίδες ενώ υπακούουν τους αυστηρούς περιορισμούς. Γι' αυτόν τον διαγωνισμό, ως εκ τούτου υπάρχει το δικαίωμα να επιτραπεί κάποια συμβάντα να παραμένουν χωρίς θέση, προκειμένου να διασφαλιστεί ότι κανείς από τους αυστηρούς περιορισμούς δεν παραβιάζεται. (Οι κανόνες του ITC2007 ορίζουν ότι, αν

ένας αλγόριθμος παράγει μια λύση που περιέχει τυχόν παραβιάσεις αυστηρών περιορισμών, τότε θεωρείται άκυρη και θα πρέπει να αποκλειστεί από τον διαγωνισμό.) Η Απόσταση από το Εφικτό έτσι, υπολογίζεται λαμβάνοντας υπόψη τα γεγονότα που δεν διατίθενται στο χρονοδιάγραμμα. Ωστόσο, δεν είναι ο αριθμός των γεγονότων που λαμβάνονται υπόψη εδώ, αλλά είναι ο συνολικός αριθμός των φοιτητών που παρακολουθούν κάθε ένα από τα μη τοποθετημένα συμβάντα, αντανακλώντας την πραγματική κατάσταση όπου υπάρχει ενδιαφέρον να ικανοποιηθούν όσο το δυνατόν περισσότερες ανάγκες πολλών ανθρώπων μέσα σε ένα χρονοδιάγραμμα. Έτσι, αν, για παράδειγμα, μία συγκεκριμένη λύση χρονοδιαγράμματος έχει τρία συμβάντα, που δεν έχουν τοποθετηθεί, καθώς και ο αριθμός των φοιτητών που παρακολουθούν κάθε ένα από αυτά είναι 10, 5, και 8, τότε η Απόσταση από το Εφικτό είναι απλά $(10 + 8 + 5) = 25$. Σημείωση ότι αν όλα τα γεγονότα εισάγονται σε ένα χρονοδιάγραμμα (ενώ υπακούουν τους αυστηρούς περιορισμούς), τότε του Απόσταση από το Εφικτό είναι μηδέν.

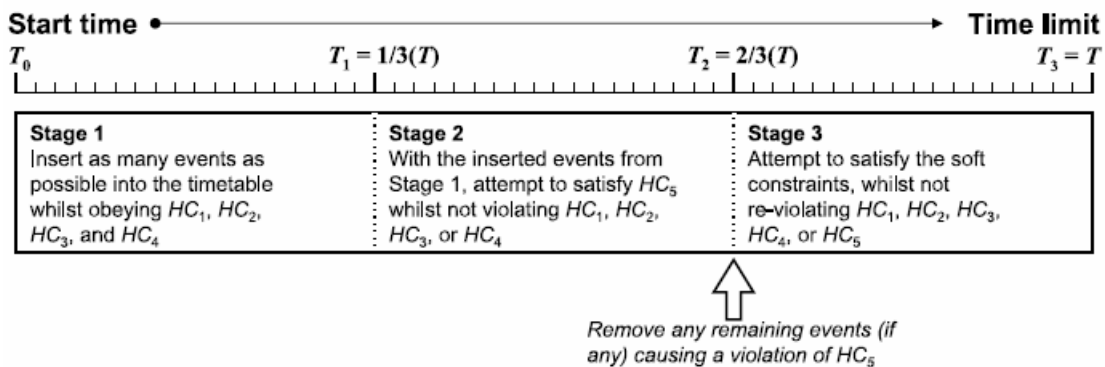
Η δεύτερη τιμή που χρησιμοποιείται για την αξιολόγηση του χρονοδιαγράμματος είναι το Ήπιο Κόστος (Soft Cost), το οποίο υπολογίζεται από μετρώντας απλά τον αριθμό των παραβιάσεων ελαστικών περιορισμών σε ένα χρονοδιάγραμμα. Για τον SC_1 , αν ένας μαθητής έχει προγραμματιστεί να παρευρεθεί σε ένα συμβάν κατά την τελευταία χρονοθυρίδα της ημέρας, τότε αυτό οδηγεί σε έναν βαθμό ποινής. (Φυσικά, αν υπάρχουν x μαθητές σε αυτή την διάλεξη, θεωρούνται ως x βαθμοί ποινής.) Για τον SC_2 αν ένας φοιτητής έχει τρία συμβάντα στη σειρά αυτό μετρείται ως ένας βαθμός ποινής. Εάν ένας φοιτητής έχει τέσσερα συμβάντα στη σειρά αυτό μετρείται ως δύο, και ούτω καθεξής. Σημειώστε ότι τα παρακείμενα γεγονότα που προκύπτουν πάνω από δύο ξεχωριστές ημέρες δεν υπολογίζονται ως παραβίαση. Τέλος, για τον SC_3 κάθε φορά που ένας μαθητής παρακολουθεί μόνο ένα συμβάν σε μια μέρα, αυτό οδηγεί σε έναν βαθμό ποινής. Το Ήπιο Κόστος είναι απλά το σύνολο αυτών των τριών τιμών.

Δεδομένων αυτών των δύο τιμών, η ακόλουθη διαδικασία χρησιμοποιείται για να συγκρίνει λύσεις. Πρώτον, λαμβάνονται υπόψη οι Αποστάσεις από το Εφικτό των λύσεων, και η λύση με τη χαμηλότερη τιμή θεωρείται νικητής. Εντούτοις, όταν δύο ή περισσότερες λύσεις είναι ίσες από την άποψη αυτή, νικητής θεωρείται η λύση μεταξύ αυτών που έχει το χαμηλότερο ήπιο κόστος.

2. Περιγραφή του Αλγορίθμου

Σε αυτήν την προσέγγιση η στρατηγική είναι να αντιμετωπιστεί το πρόβλημα σε τρία διαφορετικά στάδια, το καθένα με ένα αυστηρό όριο χρόνου, T_1 , T_2 , και T_3 αντίστοιχα (τέτοια ώστε το πλήρες χρονικό όριο είναι $T = T_1 + T_2 + T_3$). Αν ένα συγκεκριμένο στάδιο ολοκληρώνεται πριν από τη συμπλήρωση του χρονικού ορίου του, τότε ο χρόνος που απομένει περνά στο επόμενο στάδιο. Εάν το Στάδιο 3 ολοκληρώνεται νωρίς, τότε ο αλγόριθμος επίσης σταματά νωρίς. Για καθοδήγηση, μια περιγραφή των κύριων στόχων κάθε σταδίου δίνεται στην Εικόνα 10. Όπως απεικονίζεται, η ιδέα είναι να οργανωθούν οι περιορισμοί σε τρία διαφορετικά επίπεδα σημασίας. Σε κάθε διαδοχικό στάδιο, οι παραβιάσεις των περιορισμών που ικανοποιούνται σε προηγούμενο στάδιο μετά δεν επιτρέπονται. Στο πρώτο στάδιο, γίνονται προσπάθειες για να ικανοποιηθούν οι αυστηροί περιορισμοί 1 - 4 χρησιμοποιώντας εξειδικευμένες διαδικασίες. Αν και αυτές οι μέθοδοι είναι γενικά αποτελεσματικές, δεν δείχνουν ωστόσο, άμεσα εφαρμόσιμες στον απομένον αυστηρό περιορισμό HC_5 . Έτσι το δεύτερο στάδιο του αλγορίθμου ασχολείται με την αφαίρεση των παραβιάσεων του περιορισμού αυτού. Τέλος, στο Στάδιο 3, ο αλγόριθμος επικεντρώνεται στην αφαίρεση όσο το δυνατόν περισσότερες παραβιάσεις ελαστικών περιορισμών από το τρέχων χρονοδιάγραμμα. Σε αυτό το σημείο αγνοούνται τυχόν μη τοποθετημένα γεγονότα.

Σε Στάδια 2 και 3 αυτού του αλγορίθμου, η βελτιστοποίηση πραγματοποιείται χρησιμοποιώντας προσομοιωμένη απόπηση (annealing). Επειδή ο αλγόριθμος αυτός λειτουργεί σύμφωνα με τα χρονικά όρια, όταν εφαρμόζεται αυτό το metaheuristic είναι χρήσιμο να υπολογιστούν χρονοδιαγράμματα ψύξεως που λαμβάνουν υπόψη τον διαθέσιμο χρόνο υπολογισμού. Ο σκοπός συνεπώς είναι να επιτραπεί ο αλγόριθμος να εκτελέσει μια πιο αργή ψύξη (και ως εκ τούτου μια ευρύτερη, πιο σφαιρική αναζήτηση), όταν παρουσιάζεται με ένα γενναιόδωρο ποσό χρόνου εκτέλεσης, και μία ταχύτερη ψύξη (με μια πιο εντατική, άπληστη αναζήτηση), όταν μόνο μικρές ποσότητες χρόνου είναι διαθέσιμες. Η μέθοδος για τον υπολογισμό αυτών των παραμέτρων περιγράφεται στην υποενότητα 2.3.



Εικόνα 10: Υψηλού επιπέδου περιγραφή του αλγορίθμου τριών σταδίων. Εδώ, το T αντιπροσωπεύει το χρονικό όριο που ορίζεται από το πρόγραμμα συγκριτικής αξιολόγησης του διαγωνισμού.

Σημειώστε ότι σε όλα τα πειράματα που περιγράφονται εδώ, 1/3 του διαθέσιμου χρονικού ορίου T κατανέμεται σε κάθε στάδιο. Σε προκαταρκτικές δοκιμές δοκιμάστηκε επίσης ένα σύστημα που επέτρεπε σε κάθε στάδιο να τρέχει μέχρι το σύνολο των στόχων του ικανοποιηθεί (ή έως ότου επιτευχθεί το T), αλλά αυτό συχνά αποδεικνύεται να δίνει χειρότερη απόδοση σε περιπτώσεις όπου το εφικτό ήταν δύσκολο να επιτευχθεί καθώς αυτό θα οδηγούσε να μην ανατεθεί καθόλου χρόνος στο Στάδιο 3. Επίσης δοκιμάστηκε με διαφορετικές κατανομές χρόνου, όπως η χορήγηση στα Στάδια 1 και 2 ένα σύνολο 9/10 του

διαθέσιμου χρόνου εκτέλεσης, αλλά στο σύνολο βρέθηκε ότι η 1/3 κατανομή έδωσε σε κάθε στάδιο του αλγορίθμου επαρκή χρόνο για την ολοκλήρωση των στόχων.

Στη επόμενη υποενότητα, θα συζητηθούν κάποια θέματα κωδικοποίησης και προεπεξεργασίας που είναι σχετικά στο σχεδιασμό αυτού του αλγορίθμου. Οι υποενότητες 2.2, 2.3, 2.4 και στη συνέχεια περιγράφουν τα Στάδια 1, 2, και 3 του αλγορίθμου αντίστοιχα.

2.1 Θέματα Κωδικοποίησης και Προεπεξεργασίας.

Για αυτή την προσέγγιση ένα χρονοδιάγραμμα κωδικοποιείται χρησιμοποιώντας ένα καλούπι (δηλ. πλέγμα) δύο-διαστάσεων ($r \times 45$) στο οποίο οι γραμμές αντιπροσωπεύουν τις αίθουσες και οι στήλες αντιπροσωπεύουν τις χρονοθυρίδες. Μέσα από αυτή την εργασία αναφέρεται σε αυτό το πλέγμα χρονοδιαγράμματος ως tt και χρησιμοποιείται ο συμβολισμός $tt[i, j]$ για να υποδηλώσει το περιεχόμενο στην τοποθεσία (i, j) . Κάθε κελί σε αυτό το πλέγμα (π.χ. θέση στο πρόγραμμα) μπορεί να είναι κενό ή να καταλαμβάνεται από ακριβώς ένα συμβάν. Σημειώστε ότι αυτό το τελευταίο χαρακτηριστικό σημαίνει ότι είναι αδύνατο να διπλό-καταλειφθεί μία αίθουσα, που επιτρέπει να αγνοηθεί ο HC_3 .

Είναι επίσης χρήσιμο να πραγματοποιηθούν κάποια βήματα προεπεξεργασίας πριν την εκτέλεση του κυρίως σώματος του αλγορίθμου. Πρώτον, δύο επιπλέον πλέγματα κατασκευάζονται: το πλέγμα *event-room*, και το πλέγμα *conflicts*. Το πλέγμα *event-room* είναι διαστάσεων $n \times m$ και χρησιμοποιείται για να υποδηλώσει ποιες αίθουσες είναι κατάλληλες για ποια γεγονότα. Αυτό μπορεί εύκολα να υπολογιστεί για ένα συμβάν i από προσδιορίζοντας απλά ποιες αίθουσες ικανοποιούν τόσο την χωρητικότητα όσο και τα χαρακτηριστικά που απαιτούνται από το i . Το πλέγμα *conflicts* $n \times n$, εν τω μεταξύ, είναι πολύ παρόμοιο με το τυπικό πλέγμα γειτνίασης που χρησιμοποιείται για την αναπαράσταση

γραφημάτων και αναφέρει ποια ζεύγη των συμβάντων συγκρούονται (δηλαδή δεν μπορεί να ανατεθούν σε ίδια χρονοθυρίδα). Έτσι, αν δύο γεγονότα i και j έχουν ένα ή περισσότερους φοιτητές από κοινού, ή εάν και τα δύο i και j μπορούν να ανατεθούν μόνο στην ίδια αίθουσα r , τότε είναι προφανές ότι τα συμβάντα δεν μπορεί ποτέ να ανατεθούν στην ίδια χρονοθυρίδα, και έτσι τα στοιχεία (i, j) και (j, i) στο πλέγμα *conflicts* μπορούν να χαρακτηριστούν ως πραγματικά.

Η τελική πράξη της προεπεξεργασίας ασχολείται με τον αυστηρό περιορισμό HC_4 . Πρώτα απ'όλα, σημειώστε ότι αν υπάρχει ένας περιορισμός "το συμβάν i πρέπει να γίνει πριν από το συμβάν j ", τότε αυτό αυτόματα θα συνεπάγεται τον περιορισμό "το συμβάν j πρέπει να γίνει μετά το συμβάν i ". Για την προσέγγιση αυτή, αυτό σημαίνει ότι όλες οι εμφανίσεις του HC_5 μπορεί εύκολα να αποθηκευτούν σε συμπαγή τρόπο χρησιμοποιώντας ένα πίνακα A με n λίστες, όπου κάθε λίστα $A[i]$ περιέχει τα συμβάντα που πρέπει να προγραμματιστούν μετά το συμβάν i στο χρονοδιάγραμμα. (Οι "πριν" περιορισμοί δεν χρειάζεται να ληφθούν υπόψη). Δεύτερον, μπορεί επίσης να προβεί σε περαιτέρω προσθήκες στο A σημειώνοντας ότι ο αυστηρός περιορισμός HC_5 είναι μεταβατικός (δηλαδή εάν το συμβάν i πρέπει να συμβεί πριν από το συμβάν j , και από το συμβάν j πρέπει να συμβεί πριν από το συμβάν k , τότε αυτό σημαίνει ότι το συμβάν i πρέπει επίσης να συμβεί πριν από το συμβάν k).

2.2 Περιγραφή του Αλγορίθμου: Στάδιο 1

Στο στάδιο 1, ο στόχος είναι να προστεθούν όσο το δυνατόν περισσότερα n συμβάντα εντός του χρονοδιαγράμματος χωρίς να παραβιάζουν τους πρώτους τέσσερις αυστηρούς περιορισμούς. Μια ακριβής περιγραφή ψευδό-κώδικα παρουσιάζεται στην Εικόνα 11. Όπως φαίνεται, αυτό το στάδιο παίρνει ως ορίσματα το άδειο χρονοδιάγραμμα tt , ένα όριο επανάληψης I , και μια λίστα με τα τρέχοντα μη τοποθετημένα συμβάντα U ($|U| = n$). Τα συμβάντα λαμβάνονται τότε ένα-ένα από το U και εισάγονται μέσα σε κατάλληλες θέσεις στο

tt. Προκειμένου να δοκιμάσει και να μεγιστοποιήσει τον αριθμό των συμβάντων που έχουν εισαχθεί, ο ευρετικός κανόνας h_1 χρησιμοποιείται για την επιλογή του επόμενου συμβάντος, με τους δεσμούς να έχουν σπάσει χρησιμοποιώντας τον h_2 , και περαιτέρω δεσμούς με τον h_3 (βλ. Πίνακας 6). Σημειώστε ότι αυτά τα συγκεκριμένα ευρετικά είναι παρόμοια με εκείνα που χρησιμοποιούνται στον αλγόριθμο του Dsaturn για χρωματισμό γράφων, αν και σε αυτή την περίπτωση ο h_1 λαμβάνει επίσης υπόψη το θέμα της κατανομής των αιθουσών. Ο κανόνας h_1 επιλέγει ως εκ τούτου τα συμβάντα με βάση την κατάσταση του τρέχοντος μερικού χρονοδιαγράμματος *tt*, και ιεραρχεί αυτά με τις λιγότερο εφικτές επιλογές που απομένουν. Εν τω μεταξύ, ο h_2 κανόνας ιεραρχεί αυτά τα συμβάντα που έχουν τον υψηλότερο αριθμό συγκρούσεων που, ως ένας γενικός κανόνας, είναι συχνά τα πιο προβληματικά συμβάντα για να εισαχθούν. Σημειώστε ότι τα συμβάντα χωρίς απομένουσες επιλογές τοποθέτησης αγνοούνται σε αυτό το σημείο. Τέλος, για την επιλογή μια θέσης για κάθε συμβάν, χρησιμοποιείται ο κανόνας h_4 , ο οποίος επιλέγει τη θέση αυτού που η απασχόληση θα έχει τη μικρότερη επίδραση στις επιλογές τοποθέτησης στα υπόλοιπα μη τοποθετημένα συμβάντα στο U . Οι δεσμοί αυτού του κανόνα είναι σπασμένοι χρησιμοποιώντας τον h_5 και περαιτέρω δεσμοί με τον h_6 .

Heuristic	Description
h_1	Choose the event with the smallest number of suitable places in <i>tt</i> to which it can be assigned.
h_2	Choose the unplaced event that conflicts with the most other events.
h_3	Choose an event randomly.
h_4	Choose the place that is suitable for the least number of other unplaced events in U .
h_5	Choose the place in the timeslot with the fewest events in.
h_6	Choose a place randomly.
h_7	Choose the event with the least number of students.

Πίνακας 6: Ευρετικοί κανόνες που χρησιμοποιούνται στο Στάδιο 1

Στο τέλος αυτού του σταδίου ανάθεσης, η λίστα U θα είναι άδεια (σε αυτήν την περίπτωση όλα τα γεγονότα έχουν ανατεθεί στο χρονοδιάγραμμα), ή η U θα περιέχει μόνο γεγονότα που δεν έχουν κατάλληλες θέσεις στο tt . Στην τελευταία περίπτωση, η διαδικασία ITERATED-HEURISTIC-SEARCH καλείται, η οποία χρησιμοποιείται για να προσπαθήσει να μεταφέρει περαιτέρω γεγονότα από την U στο tt , διασφαλίζοντας ότι οι αυστηροί περιορισμοί 1-4 δεν παραβιάζονται κατά τη διαδικασία. Αρχικά, η υπό-διαδικασία HEURISTIC-SEARCH καλείται, η οποία λειτουργεί με το να προσπαθεί επανειλημμένα να μετακινήσει γεγονότα από την U σε κενές θέσεις στο tt (γραμμές 3-7). Καθώς γίνεται αυτό όμως, η HEURISTIC-SEARCH, επίσης, ανακατεύει τα συμβάντα εντός του tt , έτσι ώστε οι ελεύθερες θέσεις αλλάζουν θέση (γραμμές 9-15). Το σκεπτικό αυτής της τελευταίας ενέργειας είναι ότι προσφέρει τη δυνατότητα ώστε να προστεθούν περαιτέρω συμβάντα από την U στο tt όταν γυρίζουμε πίσω στη γραμμή (1) της HEURISTIC-SEARCH. Ωστόσο, μολονότι η διαδικασία HEURISTIC-SEARCH είναι αρκετά αποτελεσματική στη μείωση του αριθμού των μη τοποθετημένων συμβάντων, στα αρχικά πειράματα είχε επίσης παρατηρηθεί ότι είναι μόνο σε θέση να το κάνει αυτό για ένα αρκετά σύντομο χρονικό διάστημα, μετά την αδρανοποίηση της διαδικασίας, χωρίς περαιτέρω συμβάντα που μεταφέρονται από την U στο tt . Για να αντιμετωπιστεί αυτό, η ITERATED-HEURISTIC-SEARCH περιλαμβάνει ένα μηχανισμό που προορίζεται για την αναζωογόνηση της διαδικασίας, η οποία επιτυγχάνεται με τη διαδικασία EXTRACT-SOME-EVENTS, η οποία αφαιρεί άλλα συμβάντα από τον tt και τα βάζει σε μία δεύτερη λίστα V . Φυσικά, με την αφαίρεση συμβάντων από το tt , δημιουργούνται επιπλέον ελεύθερες θέσεις που μπορούν να χρησιμοποιηθούν από ορισμένα από τα συμβάντα στην U . Έτσι, τα συμβάντα στην V τοποθετούνται σε μια άκρη προσωρινά, και η HEURISTIC-SEARCH και πάλι εφαρμόζεται με τη χρήση της U και του νέου, κενού χρονοδιαγράμματος. Τέλος, μετά την ολοκλήρωση της δεύτερης φάσης της ευρετικής αναζήτησης, τα γεγονότα στο V προστίθενται στα γεγονότα (αν υπάρχουν) που εξακολουθούν

να διαμένουν στο U και ολόκληρη η διαδικασία ITERATED-HEURISTIC-SEARCH επαναλαμβάνεται.

STAGE-1(tt, U, I)

- (1) **while** (\square events in U with suitable places in tt)
- (2) Select an event $e \in U$ that has suitable places in tt ;
- (3) Choose a suitable place p for e ;
- (4) Move e from U into tt at place p ;
- (5) ITERATED-HEURISTIC-SEARCH(tt, U, I);

ITERATED-HEURISTIC-SEARCH(tt, U, I)

- (1) **while** ($U \neq 0$ **and** (timelimit T_1 not reached))
- (2) HEURISTIC-SEARCH(tt, U, I);
- (3) **if** ($U \neq 0$)
- (4) $V \leftarrow$ EXTRACT-SOME-EVENTS($tt, |U|$);
- (5) HEURISTIC-SEARCH(tt, U, I);
- (6) $U \leftarrow U \square V$;

HEURISTIC-SEARCH(tt, U, I)

- (1) Make a list P of all the unoccupied places in tt ;
- (2) $i \leftarrow 0$;
- (3) **while** ($U \neq 0$ **and** $P \neq 0$ **and** $i < I$)
- (4) **foreach** ($u \in U$ **and** $p \in P$)
- (5) **if** (p is a suitable place in tt to assign u)
- (6) Put u into p in tt ;
- (7) Remove u from U and p from P ;
- (8) **if** ($U \neq 0$ **and** $P \neq 0$)
- (9) **repeat**
- (10) Choose a random event e in tt and $p \in P$;
- (11) **if** (p is suitable place in tt to assign e)
- (12) Move e from its current place to p ;
- (13) Update P to reflect the changes;
- (14) $i \leftarrow i + 1$;
- (15) **until** ($i = I$ or (e has been moved to p))

EXTRACT-SOME-EVENTS(tt, q)

- (1) $V \leftarrow 0$;
- (2) **for** ($i \leftarrow 1$ **to** q)
- (3) Randomly choose two events e and g in tt ;
- (4) Move either e or g (according to h_7) from tt to V ;

Εικόνα 11: Περιγραφή ψευδό-κώδικα του σταδίου 1. Εδώ, το tt αντιπροσωπεύει το $(r \times 45)$ πλέγμα χρονοδιαγράμματος και U και V είναι οι λίστες των μη τοποθετημένων συμβάντων. (Όταν η STAGE-1 καλείται για πρώτη φορά, $|U| = n.$) το I ορίζει το όριο επανάληψης της διαδικασίας HEURISTIC-SEARCH

Εξετάζοντας τον ψευδό-κώδικα της διαδικασίας ITERATED-HEURISTIC-SEARCH, δύο σημαντικά χαρακτηριστικά γίνονται εμφανή. Πρώτον, πρέπει να σημειωθεί ότι αν επιλεγθεί να μεταφερθούν κάποια γεγονότα από το U στην V (γραμμή (4)) και στη συνέχεια προστεθούν τα περιεχόμενα του V στην U (γραμμή (6)), τότε σε ορισμένες περιπτώσεις, ο συνολικός αριθμός των μη τοποθετημένων συμβάντων μπορεί να αυξηθεί, αντίθετα με τους στόχους του Σταδίου 1. Ωστόσο, στην πράξη, εάν μια τέτοια κατάσταση προκύψει, είναι συνήθως προσωρινή, επειδή ο αριθμός των γεγονότων στην U προβλέπεται γενικά να μειωθεί και πάλι, όταν ο αλγόριθμος γυρίζει πίσω στην έναρξη της διαδικασίας.

Το δεύτερο θέμα, εν τω μεταξύ, αφορά τη στρατηγική της εξαγωγής συμβάντος που χρησιμοποιείται στην διαδικασία EXTRACT-SOME-EVENTS. Μια επιλογή που προσφέρεται εδώ είναι να επιλυθούν απλά τυχαία γεγονότα για αφαίρεση. Ωστόσο, στη συγκεκριμένη περίπτωση, φαίνεται λογικό να προωθηθεί η επιλογή για εξάλειψη των μικρότερων συμβάντων από το χρονοδιάγραμμα, λόγω στο γεγονός ότι τα μη τοποθετημένα συμβάντα που περιέχουν λιγότερους οι φοιτητές θα προσελκύσουν μία χαμηλότερη Απόσταση από το Εφικτό, όταν το χρονοδιάγραμμα αξιολογείται. Έτσι, ο ευρετικός κανόνας h_7 (Πίνακας 6) χρησιμοποιείται εδώ.

Ο Πίνακας 7 συνοψίζει την αποτελεσματικότητα του Σταδίου 1 για τις εικοσιτέσσερις περιπτώσεις του προβλήματος. Λεπτομέρειες σχετικά με τα μεγέθη της κάθε περίπτωσης περιλαμβάνονται επίσης εδώ: ο αριθμός των συμβάντων n , οι αίθουσες m και οι φοιτητές s . Με δεκαέξι από τις περιπτώσεις, $\geq 90\%$ των συμβάντων εισάγονται στο χρονοδιάγραμμα από τους κανόνες ευρετικής ανάθεσης, κατά μέσο όρο. Με τις υπόλοιπες οκτώ περιπτώσεις το ποσοστό αυτό είναι κατά μέσο όρο 80. Βλέπουμε, επίσης, ότι μετά την μετέπειτα εφαρμογή της ITERATED-HEURISTIC-SEARCH, $>99\%$ των συμβάντων που εισάγονται εντός του χρονοδιαγράμματος σε όλες τις περιπτώσεις, με εξαίρεση την περίπτωση 22, το μεγαλύτερο πρόβλημα στο σύνολο, όπου έχουν ανατεθεί κατά μέσο όρο μόνο το 95% των συμβάντων.

Instance	n	m	s	Before (%)	After (%)
comp-2007-2-1	400	10	500	86.3 ± 1.2	100 ± 0.0
comp-2007-2-2	400	10	500	83.9 ± 1.2	100 ± 0.0
comp-2007-2-3	200	20	1000	95.3 ± 1.2	100 ± 0.0
comp-2007-2-4	200	20	1000	92.1 ± 1.3	100 ± 0.0
comp-2007-2-5	400	20	300	92.7 ± 1.0	100 ± 0.0
comp-2007-2-6	400	20	300	92.2 ± 0.9	100 ± 0.0
comp-2007-2-7	200	20	500	93.2 ± 1.1	100 ± 0.0
comp-2007-2-8	200	20	500	93.4 ± 1.1	100 ± 0.0
comp-2007-2-9	400	10	500	85.5 ± 1.4	100 ± 0.0
comp-2007-2-10	400	10	500	81.2 ± 1.4	100 ± 0.1
comp-2007-2-11	200	10	1000	94.5 ± 1.3	100 ± 0.0
comp-2007-2-12	200	10	1000	91.3 ± 1.2	100 ± 0.0
comp-2007-2-13	400	20	300	90.0 ± 1.2	100 ± 0.0
comp-2007-2-14	400	20	300	90.7 ± 0.9	100 ± 0.0
comp-2007-2-15	200	10	500	91.5 ± 1.3	100 ± 0.0
comp-2007-2-16	200	10	500	98.0 ± 0.8	100 ± 0.0
comp-2007-2-17	100	10	500	99.9 ± 0.4	100 ± 0.0
comp-2007-2-18	200	10	500	89.4 ± 2.2	100 ± 0.0
comp-2007-2-19	300	10	1000	86.9 ± 1.0	99.4 ± 0.6
comp-2007-2-20	400	10	1000	98.6 ± 0.7	100 ± 0.0
comp-2007-2-21	500	20	300	94.3 ± 1.0	100 ± 0.0
comp-2007-2-22	600	20	500	83.9 ± 0.8	95.0 ± 1.0
comp-2007-2-23	400	20	1000	85.6 ± 1.4	99.8 ± 0.3
comp-2007-2-24	400	20	1000	97.6 ± 0.6	100 ± 0.0

Πίνακας 7: Ποσοστό των συμβάντων που εισάγονται στο χρονοδιάγραμμα πριν και μετά την εφαρμογή της ITERATED-HEURISTICSEARCH. Τα παραπάνω στοιχεία είναι το αποτέλεσμα από 51 εκτελέσεις σε κάθε περίπτωση, μαζί με την τυπική απόκλιση. Οι περιπτώσεις 1-8 είναι οι «αρχικές» περιπτώσεις, 9-16 οι "πρόσφατες" περιπτώσεις, και 17-24 οι «κρυφές» περιπτώσεις.

2.3 Περιγραφή του Αλγορίθμου: Στάδιο 2

Στο Στάδιο 2 του αλγορίθμου, η προσοχή στρέφεται προς την εξάλειψη των παραβιάσεων του απομένοντος αυστηρού περιορισμού HC_5 . Όπως αναφέρθηκε νωρίτερα, αυτό γίνεται με τη χρήση της προσομοιωμένης απόκτησης (SA). Η συνάρτηση κόστους $C(tt)$ που χρησιμοποιείται σε αυτή την φάση είναι:

$$C(tt) = \sum_{i=1}^n \sum_{j=1}^{|A[i]|} f(i, j), \quad (1)$$

where

$$f(i, j) = \begin{cases} \frac{\text{size}(i) + \text{size}(A[i]_j)}{B} & \text{if } (\text{slot}(i) \geq \text{slot}(A[i]_j)) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Εδώ, το A αναφέρεται στον πίνακα των λιστών που περιγράφονται στην Υποενότητα 2.1, το $A [i]_j$ δηλώνει το j -στό στοιχείο στην λίστα $A [i]$, και η υποδοχή (i) υποδεικνύει την χρονοθυρίδα που το συμβάν i ανατίθεται στο χρονοδιάγραμμα tt . Επιπλέον, το μέγεθος (i) υποδεικνύει τον αριθμό των φοιτητών που παρακολουθούν το συμβάν i , και το B είναι μία σταθερά που αντανακλά το συνολικό αριθμό των φοιτητών στα δύο μεγαλύτερα συμβάντα. Το κόστος λειτουργίας C έτσι αντανακλά τον αριθμό των παραβιάσεων του HC_5 στο tt , με κάθε παραβίαση να συνεπάγεται μια τιμή ποινής μεταξύ του μηδενός και του ένα. Σε μια προηγούμενη έκδοση αυτού του αλγορίθμου (Lewis 2008) όλες οι παραβιάσεις του H_5 συνεπάγονται μια τιμή ποινής του ενός, όμως αυτή η τρέχουσα μέθοδος είναι μια βελτίωση, δεδομένου ότι δίνει προτεραιότητα στην ικανοποίηση του HC_5 με μεγαλύτερα συμβάντα, αποδίδοντας προσοχή στο έσχατο μέσο της ποιότητας: την απόσταση από το εφικτό. Προφανώς, ο στόχος σε αυτή τη φάση είναι να παραχθεί μία λύση tt με $C(tt) = 0$.

Ο τελεστής γειτονιάς που χρησιμοποιείται στην παρούσα φάση λειτουργεί με τυχαία επιλογή ενός κελιού $tt [a, b]$ στο χρονοδιάγραμμα. Ένα δεύτερο κελί $tt [c, d]$, τότε επίσης επιλέγεται τυχαία, και τα περιεχόμενα των δύο κελιών αντιστρέφονται. Αν μια τέτοια κίνηση προκαλεί παραβίαση των αυστηρών περιορισμών 1-4, τότε αμέσως απορρίπτεται και γίνεται επαναφορά. Αλλιώς είναι αποδεκτή, και το tt επαναξιολογείται χρησιμοποιώντας το C . Σημειώστε ότι στην επιλογή του δεύτερου κελιού $tt [c, d]$, είναι απαραίτητο να εξασφαλιστεί ότι $tt [a, b] \neq tt [c, d]$. Αυτό εγγυάται (α) ότι οι δύο επιλεγμένα κελιά είναι διακριτά, και (β) ότι τουλάχιστον ένα από τα επιλεγμένα κελιά είναι κατειλημμένο (προφανώς, εάν είτε (α),

είτε (β) δεν ήταν αληθής τότε η αντίστοιχη κίνηση θα οδηγήσει σε ένα ταυτόσημο χρονοδιάγραμμα).

Σημειώστε ότι η εφαρμογή αυτού του τελεστή γειτονιάς έχει αποτέλεσμα σε μία από τις δύο ενέργειες. Η πρώτη συμβαίνει όταν επιλέγονται δύο κατειλημμένα κελιά, που προκαλεί την ανταλλαγή των θέσεων των δύο συσχετιζόμενων συμβάντων στο χρονοδιάγραμμα. Το δεύτερο συμβαίνει όταν ένα κατειλημμένο και ένα κενό κελί επιλέγονται, προκαλώντας την μετακίνηση μόνο ενός συμβάντος σε διαφορετική θέση στην χρονοδιάγραμμα. Λόγω του τρόπου με τον οποίο τα κελιά επιλέγονται σε αυτή την περίπτωση, η πιθανότητα καθεμίας από αυτές τις δράσεις που συμβαίνουν, σχετίζεται άμεσα με την αναλογία των κατειλημμένων κελιών στο χρονοδιάγραμμα: υποθέτοντας (χωρίς βλάβη της γενικότητας) ότι τα γεγονότα n είναι παρόντα σε ένα χρονοδιάγραμμα με $p = 45m$ θέσεις, τότε η πιθανότητα μιας ανταλλαγής είναι:

$$P(\text{swap}) = \frac{n}{p} \times \frac{n-1}{p-1}, \quad (3)$$

(δηλ. η υπό όρους πιθανότητα της επιλογής ενός κατειλημμένου κελίου στο πλέγμα, ακολουθούμενη από ένα άλλο, διαφορετικό κατειλημμένο κελί). Η πιθανότητα εκτέλεσης μια κίνησης, εν τω μεταξύ, είναι:

$$P(\text{move}) = \left(\frac{p-n}{p} \times 1.0 \right) + \left(\frac{n}{p} \times \frac{p-n}{p-1} \right), \quad (4)$$

(δηλ. το άθροισμα της πιθανότητας της επιλογής ενός κενού κελίου που ακολουθείται από ένα κατειλημμένο κελί, και η πιθανότητα επιλογής των κατειλημμένων κελιών που ακολουθούνται από ένα κενό). Σημειώστε ότι αυτές οι πιθανότητες προκύπτουν φυσικά λόγω της επιλεγμένης μεθόδου της επιλογής κελιού και θα καθορίζεται σε κάθε εκτέλεση του

αλγόριθμου. Τονίζονται εδώ για να βοηθήσουν την ανάλυση του αλγόριθμου και σημειώνεται ότι στην πράξη θα ήταν εύκολο να επισυνάπτουν διαφορετικές πιθανότητες για τις ενέργειες ανταλλαγής και μετακίνησης αν ήταν επιθυμητό.

Λαμβάνοντας υπόψη την παραπάνω συνάρτηση κόστους και τον τελεστή γειτονιάς, μια απλή εφαρμογή της SA. χρησιμοποιείται τώρα: ξεκινώντας από μια αρχική "θερμοκρασία" t_0 , κατά τη διάρκεια της εκτέλεσης η μεταβλητή της θερμοκρασίας σιγά-σιγά μειώνεται σύμφωνα με έναν κανόνα ενημέρωσης θερμοκρασίας $t_{i+1} = \alpha t_i$, όπου α ($0 < \alpha < 1$) είναι μία μεταβλητή γνωστή ως "ρυθμός ψύξης". Σε κάθε θερμοκρασία t_i , μία αλυσίδα μήκους n^2 του Markov δημιουργείται στη συνέχεια εκτελώντας εφαρμογές n^2 του τελεστή γειτονιάς. Κάθε κίνηση που αυξάνει το κόστος του χρονοδιαγράμματος είναι τότε αποδεκτή με πιθανότητα $\exp(-\delta/t_i)$, όπου δ είναι η μεταβολή του κόστους που προκαλεί αυτή η κίνηση. Κάθε κίνηση που μειώνει ή αφήνει αμετάβλητο το κόστος, εν τω μεταξύ, είναι αποδεκτή αυτόματα.

Επειδή ο αλγόριθμος αυτός έχει ως στόχο να είναι εξαρτώμενος από τον χρόνο, είναι καλή ιδέα να επιλεγθεί ένας ρυθμός ψύξης α που να επιτρέπει στις αλυσίδες Markov να παράγονται σε όσο το δυνατόν περισσότερες θερμοκρασίες, μεταξύ της αρχικής θερμοκρασίας t_0 και κάποιας τελικής θερμοκρασίας. Για να υπολογιστεί ένας τέτοιος ρυθμός ψύξης ο αλγόριθμος SA τρέχει για πρώτη φορά για το 5% του κατανεμημένου χρόνου του Σταδίου2, και ο αριθμός των αλυσίδων Markov που δημιουργούνται καταγράφεται. Το μοτίβο αυτό χρησιμοποιείται στη συνέχεια για την πρόβλεψη του αριθμού μ των αλυσίδων Markov που δημιουργούνται στο υπόλοιπο 95% του χρόνου. Χρησιμοποιώντας τον μ , μπορεί στη συνέχεια να υπολογιστεί μια τιμή για τον α που εξασφαλίζει ότι η θερμοκρασία θα μειωθεί από την t_0 σε μία συγκεκριμένη τελική θερμοκρασία t_μ σε ακριβώς μ βήματα ως:

$$\alpha = (t_\mu/t_0)^{1/\mu} \quad (5)$$

Στην περίπτωση μας, μετά από άλλες εργασίες, η αρχική θερμοκρασία t_0 προσδιορίζεται αυτόματα με την εκτέλεση ενός μικρού δείγματος από κινήσεις γειτονιάς και στη συνέχεια με τον υπολογισμό της διακύμανσης του κόστους αυτών των κινήσεων. Η τελική θερμοκρασία t_m , εν τω μεταξύ, θα πρέπει να καθορίζεται από το χρήστη.

Τέλος, το Στάδιο 2 ολοκληρώνεται είτε όταν ένα tt χρονοδιάγραμμα βρίσκεται με κόστος $C(tt) = 0$, ή όταν επιτυγχάνεται το χρονικό όριο T_2 . Εάν συμβαίνει το τελευταίο, τότε λαμβάνεται η καλύτερη λύση που βρέθηκε κατά τη διάρκεια αυτού του σταδίου, και τα γεγονότα που φαίνονται να προκαλούν παραβίαση του HC_5 αφαιρούνται ένα-ένα από το tt μέχρι να είναι εντελώς απαλλαγμένο από οποιεσδήποτε παραβιάσεις αυστηρών περιορισμών. Αυτό το προκύπτον χρονοδιάγραμμα στη συνέχεια μετακυλιέται στο Στάδιο 3.

Ο Πίνακας 8 συνοψίζει τα αποτελέσματα του Σταδίου 2 για τις είκοσι τέσσερις περιπτώσεις του διαγωνισμού. Σημειώστε ότι δεν υπάρχει σημαντική συσχέτιση μεταξύ των αρχικών και των τελικών δαπανών των χρονοδιαγραμμάτων. Συνολικά, δεκαέξι από τις περιπτώσεις διαθέτουν τελικό κόστος με μέσες τιμές και τυπικές αποκλίσεις κοντά στο μηδέν, υποδεικνύοντας ότι η διαδικασία είναι σε θέση να ολοκληρώσει επιτυχώς τους στόχους της στην πλειονότητα των εκτελέσεων. Φαίνεται ότι οι περιπτώσεις που προκάλεσαν δυσκολίες με την ανάθεση heuristics στο Στάδιο 1, δηλαδή οι περιπτώσεις 1, 2, 9, 10, 19, 22, και 23 επίσης αποδεικνύονται προβληματικές εδώ, και τα συμβάντα που συνήθως πρέπει να αφαιρεθούν από αυτά τα χρονοδιαγράμματα για να τα ελευθερώσει από τις υπόλοιπες παραβιάσεις του HC_5 . Αυτή είναι επίσης η περίπτωση με το παράδειγμα 24, όπου κατά μέσο όρο 1,8 συμβάντα πρέπει να αφαιρεθούν. Ένας πιθανός λόγος για τις χαμηλές επιδόσεις σε αυτές τις οκτώ περιπτώσεις είναι ότι οι υποψήφιες λύσεις σε ορισμένα από αυτά τα προβλήματα έχουν υψηλές αναλογίες από κατελιμμένα κελιά (>66% των διαθέσιμων θέσεων με περιπτώσεις 1, 2, 9, 10, 19, και 22), επομένως σύμφωνα με το (3) ένα μεγαλύτερο ποσοστό των προτεινόμενων γειτονικών κινήσεις θα είναι αντιμεταθέσεις, οι οποίες είναι πιο

πιθανό να απορριφτούν από και τα δύο γεγονότα που εμπλέκονται θα πρέπει να ανατεθούν σε κατάλληλες θέσεις.

Υπάρχουν εξαιρέσεις σε αυτή την παρατήρηση όμως: οι περιπτώσεις 23 και 24, για παράδειγμα, διαθέτουν σχετικά χαμηλά ποσοστά πληρότητας του 44%, αλλά εξακολουθούν να δείχνουν μια σημαντική έλλειψη στην απόδοση σε αυτό το στάδιο. Ωστόσο, όπως θα δούμε στην Εικόνα 12 αργότερα, όλες οι οκτώ περιπτώσεις που προκαλούν δυσκολίες εδώ επίσης έχουν την τάση να διαθέτουν ένα αρκετά υψηλό ποσοστό απόρριψης των προτεινόμενων γειτονικών κινήσεων, που φαίνεται να κάνουν κινήσεις στο χώρο αναζήτησης πιο περιορισμένα, κάνοντας τις βελτιώσεις στην εξερεύνηση και στο κόστος που προκύπτουν πιο δύσκολο να επιτευχθούν.

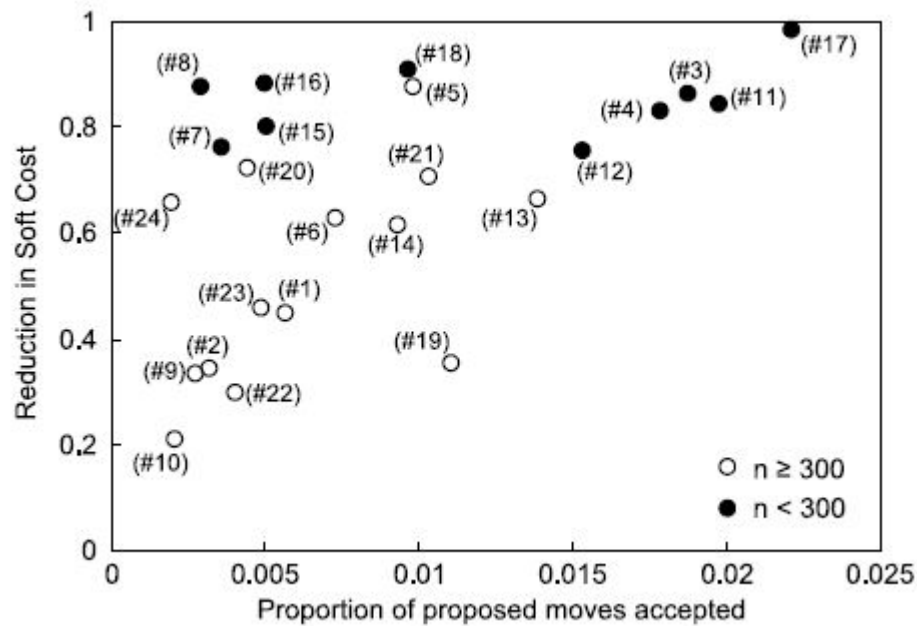
Instance	Start	End	Removed
comp-2007-2-1	14.9 ± 3.0	0.5 ± 0.7	0.5 ± 0.8
comp-2007-2-2	13.8 ± 3.1	1.6 ± 1.6	1.8 ± 1.8
comp-2007-2-3	7.4 ± 1.4	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-4	8.1 ± 1.7	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-5	57.1 ± 7.6	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-6	52.9 ± 4.1	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-7	6.4 ± 1.8	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-8	7.7 ± 2.1	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-9	16.9 ± 2.6	2.8 ± 2.1	3.0 ± 2.1
comp-2007-2-10	16.0 ± 2.4	4.3 ± 2.6	4.6 ± 2.8
comp-2007-2-11	6.0 ± 1.6	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-12	7.4 ± 2.5	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-13	52.0 ± 5.9	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-14	52.6 ± 5.2	0.0 ± 0.2	0.0 ± 0.2
comp-2007-2-15	7.7 ± 2.0	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-16	10.9 ± 2.1	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-17	3.1 ± 1.0	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-18	7.3 ± 1.8	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-19	11.9 ± 2.3	4.0 ± 2.1	4.4 ± 2.3
comp-2007-2-20	16.7 ± 3.0	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-21	70.3 ± 6.1	0.0 ± 0.0	0.0 ± 0.0
comp-2007-2-22	77.0 ± 8.5	32.4 ± 10.4	31.1 ± 9.0
comp-2007-2-23	18.0 ± 3.5	4.3 ± 2.4	4.5 ± 2.3
comp-2007-2-24	116.2 ± 10.6	1.8 ± 2.4	1.8 ± 2.1

Πίνακας 8: Κόστος του χρονοδιαγράμματος (με την διεργασία κόστους C) στην αρχή και στο τέλος του Σταδίου 2, και ο αριθμός των επιπλέον συμβάντων που αφαιρούνται προκειμένου να ελευθερώσει τα χρονοδιαγράμματα των υπόλοιπων παραβιάσεων αυστηρών περιορισμών. Τα παρουσιαζόμενα στοιχεία είναι οι μέσοι όροι από 51 εκτελέσεις σε κάθε περίπτωση, μαζί με την τυπική απόκλιση.

2.4 Περιγραφή του Αλγορίθμου: Στάδιο 3

Στο στάδιο 3, η προσοχή στρέφεται προς την ικανοποίηση των ελαστικών περιορισμών του προβλήματος. Όταν αυτό το τμήμα του αλγορίθμου καλείται, θα έχει προκύψει μία από τις δύο περιπτώσεις: είτε ένα έγκυρο χρονοδιάγραμμα με Απόσταση από το Εφικτό μηδέν θα έχει παραχθεί ή, αφού περάσουν τα 2/3 του διαθέσιμου χρόνου λειτουργίας που προσπαθεί να ασχοληθεί με τους αυστηρούς περιορισμούς, θα έχει εγκατασταθεί για ένα χρονοδιάγραμμα που έχει κάποια μη τοποθετημένα συμβάντα. Στην τελευταία περίπτωση, αυτές τα μη τοποθετημένα συμβάντα δεν λαμβάνονται υπόψη περαιτέρω - δηλ. εφόσον είναι αποβάλλονται αποτελεσματικά από το πρόβλημα.

Η εφαρμογή της SA που χρησιμοποιείται εδώ είναι και πάλι απλή, με το μοτίβο ψύξης να υπολογίζεται με τον ίδιο τρόπο όπως το Στάδιο 2. Ο τελεστής γειτονιάς είναι επίσης ο ίδιος με Στάδιο 2, αν και σε αυτή την περίπτωση, οι κινήσεις που προκαλούν μια παράβαση του αυστηρού περιορισμού HC_5 (εκτός από τις προηγούμενες τέσσερις) επίσης επαναφέρονται αμέσως. Τέλος, η συνάρτηση κόστους που χρησιμοποιείται εδώ είναι απλά το Soft Cost (βλ. υποενότητα 2), η οποία είναι κατάλληλη, λόγω του γεγονότος ότι δεν επιτρέπονται παραβιάσεις αυστηρών περιορισμών στο χρονοδιάγραμμα από αυτό το σημείο και μετά.



Εικόνα 12: Διάγραμμα διασποράς δείχνει τη σχέση μεταξύ του ποσοστού των αποδεκτών κινήσεων γειτονιάς στο Στάδιο 3, και η προκύπτουσα μείωση του Soft Cost (που εκφράζεται ως αναλογία). Κάθε σημείο στην γραφική παράσταση είναι κατά μέσο όρο σε όλες τις διαδρομές με ένα συγκεκριμένο παράδειγμα, όταν επιτεύχθηκε η Απόσταση από το Εφικτό να ισούται με μηδέν, με την εξαίρεση το παράδειγμα 22 όπου η εκτέλεση με την χαμηλότερη Απόσταση από το Εφικτό από τις 51 εκτελέσεις χρησιμοποιήθηκε.

4. Λύνοντας το πρόβλημα κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών με βελτιστοποίηση αποικίας μυρμηγκιών (Ant Colony Optimization)

1. Ο Αλγόριθμος

Η προσέγγισή μας βασίζεται στην βελτιστοποίηση αποικίας μυρμηγκιών (ACO) και μπορεί να χαρακτηριστεί πιο συγκεκριμένα ως σύστημα μυρμηγκιών (AS). Σε αλγόριθμους ACO τα τεχνητά μυρμήγκια κατασκευάζουν διαδοχικά λύσεις με βάση την παγκόσμια πληροφορία (φερομόνες) και τοπικές πληροφορίες (π.χ. κάποιο άπληστο κριτήριο). Οι φερομόνες συνεπώς δρουν ως ένα πιθανοθεωρητικό μοντέλο για την κατασκευή λύσης και διαρκώς ενισχύονται από τα μυρμήγκια που έχουν κατασκευάσει λύσεις υψηλής ποιότητας. Η εξάτμιση φερομόνης εξουδετερώνει την πρόωμη σύγκλιση επιτρέποντας την εξερεύνηση του χώρου αναζήτησης για μια μεγαλύτερη διάρκεια. Μια διαδικασία γενικής βελτιστοποίησης αποικίας μυρμηγκιών αποτελείται από τρία βήματα που επαναλαμβάνονται. Τα βήματα είναι κατασκευή λύση από τα τεχνητά μυρμήγκια, ενημέρωση φερομόνης και προαιρετικά ενέργειες όπως μια τοπική αναζήτηση.

Πιο συγκεκριμένα, το AS μπορεί να περιγραφεί με τον αλγόριθμο TimeTabling-AS(). Η κύρια εργασία για το σχεδιασμό ενός ACO αλγόριθμου είναι να επινοήσει τις φερομονικές δομές και τους κανόνες ενημέρωσης, και να βρει αποτελεσματικές τοπικές μεθόδους που είναι σε θέση να οδηγήσει τη διαδικασία προς υποσχόμενες περιοχές του χώρου αναζήτησης. Τα αντίστοιχα τμήματα του αλγορίθμου μας περιγράφονται λεπτομερώς στα ακόλουθα τμήματα.

Πληροφορίες Φερομόνης

Στον αλγόριθμο μας τα μυρμήγκια αντιστοιχίζουν συμβάντα σε χρονοθυρίδες και αίθουσες με βάση δύο είδη φερομόνης που συμβολίζονται με τ_{ij}^r και $\tau_{ik}^r * \tau_{ij}^s$ είναι ένα πλέγμα $n \times 45$

που εκπροσωπεί τις σχετικές πιθανότητες για ανάθεση ενός συμβάντος i σε μία χρονοθυρίδα j . Ομοίως, το τ_{ik}^r είναι ένα πλέγμα $n \times r$ που αντιπροσωπεύει τις σχετικές πιθανότητες ανάθεσης μιας εκδήλωσης στο i σε αίθουσα k . Η απόφαση για την αποθήκευση πληροφοριών φερομόνης με αυτόν τον τρόπο είναι ένα βασικό-χαρακτηριστικό του αλγορίθμου, καθώς αποφεύγει την χρήση μίας πολύ μεγαλύτερης δομής δεδομένων που υπονοείται από μία πιο παραδοσιακή κωδικοποίηση χρησιμοποιώντας ατομικές τιμές φερομόνης για όλους τους συνδυασμούς θυρίδα/αίθουσα/συμβάν. Από την άλλη πλευρά, περιέχει περισσότερες πληροφορίες από την αποκλειστική χρήση των φερομόνων συμβάν-χρονοθυρίδα.

Algorithm TimeTabling-AS()

```

1 while time limit not yet reached do
2   for each ant  $k = 1, \dots, m$  do
3     create random permutation  $\pi^e$  of the events
4     for each event in order  $\pi^e$  do
5       assign event based on pheromones
6     end
7   end
8   locally improve each constructed solution
9   for each solution with a better than average score do
10    pheromone amplification for assignments appearing in solution
11  end
12  pheromone evaporation
13 end

```

Προφανώς, τα διαφορετικά πλέγματα φερομόνης τ^s και τ^r δεν είναι τόσο εκφραστικά όπως ένας τρίτης τάξης τανυστής (tensor) φερομόνης (τ_{ijk}), που καλύπτει όλους τους συνδυασμούς των συμβάντων i , χρονοθυρίδων j , και αιθουσών k θα μπορούσε να είναι. Από την άλλη πλευρά ο (τ_{ijk}) μπορεί να αναμένεται να είναι αραιός, δηλαδή υπάρχουν λίγα στοιχεία διαφορετικά από το μηδέν. Έτσι, είναι πιθανό ότι τα στοιχεία τ_{ijk} μπορεί να είναι επαρκώς προσεγγίζοντα από $\tau_{ijk} \approx \tau_{ij}^s \times \tau_{ik}^r$. Επιπλέον, δεν περιμένουμε ιδιαίτερα ισχυρές αμοιβαίες εξαρτήσεις των σχέσεων συμβάν-αίθουσας και συμβάν-χρονοθυρίδας. Ας υποθέσουμε ότι κάποιο μάθημα E απαιτείται να πραγματοποιηθεί σε ένα υποσύνολο των αιθουσών συμπεριλαμβανομένων την αίθουσα R_l και τους φοιτητές που συμμετέχουν αυτό το μάθημα,

καθώς και ορισμένοι περιορισμοί προτεραιότητας απαιτούν το συμβάν που θα προγραμματιστεί νωρίς την πρώτη ημέρα, ας πούμε σε μία από τις θυρίδες S_1, \dots, S_m . Δεν υπάρχει προφανής ανάγκη να εκφραστεί η απαίτηση να εκχωρηθεί το E στο R_l , όταν χρησιμοποιείται το S_l για παράδειγμα, καθώς όλες οι αίθουσες ικανοποιούν τις απαιτήσεις του E θα είναι τυπικά είναι εξίσου καλό σε αυτή την κατάσταση μία τρέχουσα μερική λύση. Οι σκέψεις αυτές οδηγούν κατευθείαν στο συμπέρασμα ότι η ανάθεση σε μία αίθουσα είναι τυπικά λιγότερο κρίσιμη από την ανάθεση σε μια χρονοθυρίδα, η οποία υποστηρίζεται επίσης από τα πειράματα. Εάν υπάρχουν αμοιβαίες εξαρτήσεις, αυτά αντιμετωπίζονται εμμέσως από την διαδικασία κατασκευής της λύσης.

Time	No. of matrices	DTF	SCP	σ_{DTF}	σ_{SCP}	P(DTF = 0)
5 min	1	54	516	121	600	82
	2	0	215	82	406	92
1 h	1	0	120	5.6	172	99
	2	0	38	0	58	100

Πίνακας 9: Σύγκριση των διαφορετικών αναπαραστάσεων φερομόνης. Για τις δύο παραστάσεις και δύο προθεσμίες ο πίνακας περιέχει τη μέση τιμή DTF και SCP, την τυπική απόκλιση για DTF και SCP αντίστοιχα και την πιθανότητα να βρει μια εφικτή λύση. Αποτελέσματα ελήφθησαν από 50 εκτελέσεις σε κάθε μία από τις περιπτώσεις του ITC2007.

Algorithm: getNextPermItem(j)

```

1 if  $j > pos$  then
2   for  $j' = pos, \dots, j$  do
3     if  $\sigma > \xi$  then
4        $rnd \leftarrow \sigma * \text{random number}$ 
5        $q \leftarrow pos$ 
6        $\zeta \leftarrow 0$ 
7       while  $\zeta < rnd$  and  $q < 45$  do
8          $\zeta \leftarrow \zeta + w_{q-1}^s$ 
9          $q \leftarrow q + 1$ 
10      end
11      $\sigma \leftarrow \sigma - w_{q-1}^s$ 
12     swap( $q - 1, j'$ )

```

```

13         else
14             // the remaining weights are all zero
15             arbitrarily choose one of the remaining
16             indices rnd
17             swap(rnd, j')
18         end
19         pos ← pos + 1
20     end
21 end
22 return  $\pi_j^s$ 

```

Για να δοκιμαστεί η υπόθεση ότι η προσέγγιση $\tau_{ijk} \approx \tau_{ij}^s \times \tau_{ik}^r$ είναι επαρκής, υλοποιήσαμε και τις δύο παραστάσεις φερομόνης: τ_{ijk} και $\tau_{ij} + \tau_{ik}$. Εάν η προσέγγιση είναι επαρκής τότε η ποιότητα λύσης για την προσεγγισμένη αναπαράσταση δεν πρέπει να είναι χειρότερη από την πλήρη αναπαράσταση. Τα πειράματά μας υποστηρίζουν αυτή την υπόθεση, και τα αποτελέσματα καταγράφονται στον Πίνακα 9. Οι δύο αναπαραστάσεις πλέγματος εκτελούνται όντως καλύτερα από την πλήρη αναπαράσταση, επειδή ο αλγόριθμος εκτελεί από δύο έως και πέντε φορές περισσότερες επαναλήψεις ανά μονάδα χρόνου. Ο λόγος για αυτό είναι ότι το ενιαίο πλέγμα είναι πολύ μεγαλύτερο από τα συνδυασμένα δύο πλέγματα και έτσι η κατασκευή των υποψήφιων λύσεων και η εξάτμιση φερομόνης είναι πιο δαπανηρή για να υπολογιστεί. Έτσι, η αύξηση της αποτελεσματικότητας αντισταθμίζει την έλλειψη εκφραστικότητας.

Κατασκευή Λύσης

Η κατασκευή λύση θεωρεί ότι τα συμβάντα σε μια ενιαία τυχαία σειρά και εκχωρεί κάθε συμβάν σε μία αίθουσα και μία εφικτή χρονοθυρίδα με ένα άπληστο τυχαίο τρόπο (αν είναι δυνατόν) θεωρώντας την πληροφορία φερομόνης.

Τυπικά οι αλγόριθμοι ACO λαμβάνουν υπόψη επίσης τοπικά ευρετικές πληροφορίες εκτός από τις πληροφορίες φερομόνης. Αυτές οι πληροφορίες μερικές φορές ονομάζονται λειτουργία προβολής, ειδικά στην περίπτωση του προβλήματος του περιοδεύοντος πωλητή.

Δεν χρησιμοποιούμε καθόλου τοπικά ευρετικές πληροφορίες, επειδή αυτό αυξάνει την τυχαιότητα του αλγορίθμου, η οποία είναι σημαντικό για τη συμπεριφορά του αλγορίθμου στην παρουσία ενός σημαντικού σκελετού λύσης.

Για κάθε συμβάν τυχαιοποιημένα σταθμισμένες παραλλαγές των διαθέσιμων χρονοθυρίδων και αίθουσών (π^s και π^r , αντίστοιχα) παράγονται με αντιγραφή των σειρών του πλέγματος φερομόνης τ_i^s , τ_i^r σε διανύσματα w^s και w^r και τον υπολογισμό των σταθμισμένων τυχαίων παραλλαγών π^s και π^r έτσι ώστε οι εγγραφές με υψηλότερες τιμές να εμφανίζονται πιο πολύ στην αρχή. Με αυτό τον τρόπο οι θυρίδες (αίθουσες) με υψηλότερες τιμές φερομόνης για το τρέχον συμβάν είναι πιο πιθανό να εμφανιστούν νωρίτερα από ό,τι οι θυρίδες (αίθουσες) με χαμηλές τιμές φερομόνης. Αυτό μπορεί να συγκριθεί με την επιλογή ανάλογης καταλληλότητας σε γενετικούς αλγόριθμους όπως για κάθε θέση είναι επιλεγμένο ένα στοιχείο (από τα υπόλοιπα) με μία πιθανότητα ανάλογη με τις αντίστοιχες φερομόνες.

Το μυρμήγκι προσπαθεί στη συνέχεια να εκχωρήσει το τρέχον συμβάν σε ένα συνδυασμό θυρίδας/αίθουσας στη σειρά τους για τα π^s και π^r , αντίστοιχα. Η πρώτη δυνατή ανάθεση που δεν παραβιάζει κανένα αυστηρό περιορισμό σημαίνει πως η τρέχουσα μερική λύση είναι αποδεκτή. Για να εξασφαλιστεί ότι και τα δύο είδη της φερομόνης λογίζονται με ισορροπημένο τρόπο, οι συνδυασμοί θυρίδας/αίθουσας θεωρούνται στην ακόλουθη σειρά: $(\pi_1^s, \pi_1^r), (\pi_1^s, \pi_2^r), (\pi_2^s, \pi_1^r), (\pi_1^s, \pi_3^r), (\pi_2^s, \pi_2^r), (\pi_3^s, \pi_1^r), \dots, (\pi_{45}^s, \pi_r^r)$. (Εικόνα 12.)

Για να επιταχυνθεί η διαδικασία αυτή οι τυχαία σταθμισμένες παραλλαγές δεν δημιουργήθηκαν εξ ολοκλήρου εκ των προτέρων, αλλά όπως τα αιτούμενα στοιχεία υπολογίζονται με τη ζήτηση. Ο αλγόριθμος `getNextPermItem(j)` εκτελεί αυτό το έργο για τις θυρίδες και οι φερομόνες του δωματίου αντιμετωπίζονται ανάλογα. Ο αλγόριθμος επιστρέφει το ζητούμενο j -οστό στοιχείο από την τυχαία σταθμισμένη μετάθεση π^s . Το γενικό διάνυσμα w^s υποτίθεται να γεμιστεί με τις αντίστοιχες σειρές των πλεγμάτων φερομόνης τ^s για το

συμβάν i που είναι υπό εξέταση. Πριν η μέθοδος εκτελεστεί για πρώτη φορά, ας υποθέσουμε

$$\sigma \rightarrow \sum_{i=1}^{45} w_i^s .$$

ότι

Η ακέραια μεταβλητή pos αποθηκεύει το δείκτη στον οποίο η σταθμισμένη μετάθεση έχει ήδη δημιουργηθεί. Εάν η j είναι μικρότερη από την pos δεν απαιτείται κανένας περαιτέρω υπολογισμός. Διαφορετικά τα υπόλοιπα στοιχεία υπολογίζονται (γραμμές 3-20). Η διεργασία $swap(i, j)$ ανταλλάσει τα στοιχεία π_i^s, π_j^s και w_i^s, w_j^s αντιστοίχως. Στην ειδική περίπτωση που όλα τα υπόλοιπα βάρη είναι μηδέν, ή κοντά στο μηδέν μέχρι κάποια μικρή ανοχή ϵ κάποιο αυθαίρετο στοιχείο επιλέγεται (γραμμή 13-17).

Σημειώστε ότι για την προετοιμασία για το επόμενο μυρμήγκι μόνο η pos πρέπει να επαναφερθεί. Οι w^s και π^s δεν χρειάζεται να επαναρχικοποιηθούν για κάθε μυρμήγκι. Επιπλέον, για όλα εκτός από τα πρώτα μυρμήγκια η μέθοδος έχει πολύ καλύτερη απόδοση, καθώς οι w^s και π^s είναι ήδη περίπου ταξινομημένες εκ των προτέρων. Το αποτέλεσμα αυτής της προδιαλογής είναι ότι ο πιο εσωτερικός βρόγχος θα, κατά μέσο όρο, τρέχει λιγότερες φορές αφού το ξ συσσωρεύεται γρηγορότερα. Αυτό το φαινόμενο είναι ιδιαίτερα έντονο όταν η διανομή φερομόνης έχει διευθετηθεί μετά μερικές δεκάδες επαναλήψεις από τότε που συνήθως υπάρχουν μόνο μερικά μεγάλα βάρη στην w_q^s με η πλειονότητα των βαρών είναι κοντά στο μηδέν.

Ενημέρωση Φερομόνης

Μετά από κάθε επανάληψη μόνο το σύνολο των μυρμηγκιών των οποίων οι λύσεις είναι δεσμευμένες για την χαμηλότερη βαθμολογία DTF λαμβάνονται υπόψη για την ενημέρωση φερομόνης. Για αυτό το σύνολο ο μέσος όρος του σκορ του SCP υπολογίζεται και όλα τα μυρμήγκια που είναι και τα δύο μέλη του συνόλου και το σκορ SCP είναι κάτω από το μέσο

όρο προσθέτει μία ποσότητα φερομόνης ανάλογη με την ποιότητα της λύσης για τις εκτελούμενες αναθέσεις συμβάν/θυρίδα και συμβάν/αίθουσα.

Στο αρχικό στάδιο αυτά είναι πολύ λίγα μυρμηγκία, τυπικά μόνο ένα, αφού το σκορ DTF θα είναι διαφορετικό για τα περισσότερα μυρμηγκία. Στα μεταγενέστερα στάδια, καθώς όλο και περισσότερα μυρμηγκία παράγουν εφικτές λύσεις, δηλαδή ένα σκορ DTF ίσον με μηδέν, ο αριθμός των μυρμηγκιών που εναποθέτουν φερομόνη αυξάνεται και μπορεί να φτάσει μέχρι το ήμισυ του συνολικού αριθμού των μυρμηγκιών.

Η ενημέρωση φερομόνης εκτελείται σύμφωνα με αυτόν τον τύπο:

$$\tau_{ij}^s \leftarrow \max(0, \tau_{ij}^s + \Delta\tau_{ij} - \Delta\tilde{\tau}_{ij}). \quad (1)$$

το τ_{ij}^s είναι η φερομόνη η οποία είναι αποθηκευμένη στο πλέγμα φερομόνης. Το $\Delta\tau_{ij}$ είναι η ποσότητα φερομόνης που προστέθηκε από τη λύση και το $\Delta\tilde{\tau}_{ij}$ είναι ένας όρος ποινής για τα στοιχεία λύσης που παραβιάζουν ελαστικούς περιορισμούς. Ο μέγιστος υπολογισμός αποφεύγει αρνητικές τιμές φερομόνης που διαφορετικά μπορεί να εμφανιστούν λόγω της ποινικοποίησης του SCP.

Για τις αναθέσεις συμβάντος / θυρίδας, αυτό γίνεται αναλυτικά ως εξής:

$$\Delta\tau_{ij} = \begin{cases} f \cdot g & \text{if event } i \text{ and slot } j \text{ is part of the solution} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$f = \begin{cases} \frac{100}{\#unplaced} & \text{if there are unplaced events} \\ 200 & \text{otherwise} \end{cases} \quad (3)$$

$$g = \begin{cases} \frac{1000}{SCP} & SCP \neq 0 \\ 2000 & \text{otherwise} \end{cases} \quad (4)$$

Οι αριθμητικές σταθερές έχουν προσδιοριστεί με προκαταρκτικά πειράματα. Οι αριθμητές επιλέγονται έτσι ώστε για τις περισσότερες λύσεις f και g είναι μεγαλύτεροι από ένα, δηλαδή στις περιπτώσεις του δοκιμαζόμενου προβλήματος υπάρχουν λιγότερα από 100 καταταγμένα

γεγονότα και το SCP είναι συνήθως κάτω από 1000. Οι σταθερές που χρησιμοποιούνται όταν ο παρονομαστής γίνεται μηδέν επιλέγονται να είναι το διπλάσιο του παρονομαστή, ο οποίος έχει σχεδιαστεί για την ενίσχυση των καλών αναθέσεων κάπως. Παρακαλώ σημειώστε ότι η απόλυτες τιμές των σταθερών στους παραπάνω τύπους δεν είναι σημαντικές, μόνον το σχετικό τους μέγεθος είναι σημαντικό.

Αν μια ανάθεση έχει βρεθεί να προκαλέσει παραβίαση των ελαστικών περιορισμών, οι εμπλεκόμενες αναθέσεις τιμωρούνται αναλόγως

$$\Delta \tilde{\tau}_{ij} \leftarrow (1 - (1 - \gamma)^{\text{SCP}(i)}) \cdot f \cdot g, \quad (5)$$

όπου το SCP (i) υποδηλώνει την ποινή ελαστικού περιορισμού που επάγεται από το συμβάν i και το γ συμβολίζει τον παράγοντα της ποινής. Χρησιμοποιήθηκαν παράγοντες ποινή μεταξύ του 0,05 (χαμηλή ποινή) και 0,5 (υψηλή ποινή) με το 0,3 να είναι ένας καλός συμβιβασμός.

Σημειώστε ότι ενώ το $\Delta \tau_{ij}$, δηλαδή η απόθεση φερομόνης, είναι σταθερή για ολόκληρη τη λύση, η ποινή $\Delta \tilde{\tau}_{ij}$ είναι συγκεκριμένης ανάθεσης. Έτσι, διαφορετικά στοιχεία της λύσης μπορεί να λάβουν διάφορες ενημερώσεις φερομόνης.

Τέλος, η εξάτμιση φερομόνης ακολουθεί την πρότυπη AS μέθοδο, η οποία είναι

$$\tau_{ij}^s \leftarrow (1 - \rho) \tau_{ij}^s \quad (6)$$

με το ρ να είναι ο συντελεστής εξάτμισης φερομόνης. Εδώ χρησιμοποιούνται τιμές από 0,1 έως 0,3 για το ρ .

Η ενημέρωση φερομόνης για τις φερομόνες συμβάντος-αίθουσας τ_{ik}^f γίνεται κατ'αναλογία. Στην περίπτωση της στασιμότητας, δηλαδή καμία μέχρι τώρα καλύτερη λύση δεν έχει βρεθεί κατά τη διάρκεια των τελευταίων 500 επαναλήψεων, οι τιμές φερομόνης κανονικοποιούνται.

Η διαδικασία κανονικοποίησης εκτελεί μία γραμμική κλιμάκωση, έτσι ώστε η μέση τιμή των αρχικών φερομονών παραμένει η ίδια και η απόκλιση από τον μέσο όρο είναι σχετικά μικρή, π.χ. περίπου δέκα τοις εκατό. Υποθέτοντας ότι η ελάχιστη τιμή φερομόνης είναι μηδέν αυτή κλιμάκωση επιτυγχάνεται θέτοντας

$$\tau_{ij}^n = (\tau_{ij} - \bar{\tau}) \cdot \frac{\bar{\tau}\omega}{\max(\tau_{ij})} + \bar{\tau} \quad (7)$$

όπου το τ_{ij}^n υποδηλώνει την κανονικοποιημένη τιμή φερομόνης, το $\bar{\tau}$ υποδηλώνει τη μέση τιμή φερομόνης και το ω είναι η επιθυμητή εξάπλωση γύρω από το μέσο όρο. Λόγω της γραμμικότητας του τελεστή προσδοκίας, ο μέσος όρος των κλιμακούμενων τιμών φερομόνης είναι ίδιος όσο η μέση τιμή των αρχικών τιμών.

Μέθοδος Βελτίωσης

Συχνά, οι ACO προσεγγίσεις επωφελούνται σημαντικά με το να συμπεριλάβουν μία διαδικασία τοπικής αναζήτησης για τη βελτίωση των υποψηφίων λύσεων που προέρχονται από τα μυρμήγκια. Στον αλγόριθμο μας χρησιμοποιούμε ένα ευρετικό βελτίωσης αλυσίδας, καλύτερης πρώτης εκτίναξης. Μια αλυσίδα ξεκινά με την μετακίνηση ενός συμβάντος που προκαλεί παραβιάσεις ελαστικών περιορισμών σε διαφορετική χρονοθυρίδα, αν αυτό μπορεί να επιτευχθεί χωρίς να παραβιάζονται αυστηροί περιορισμοί και καθώς αφαιρείται το πολύ ένα άλλο συμβάν από τη λύση. Αν ένα συμβάν απαιτείται να αφαιρεθεί, η αλυσίδα συνεχίζεται με την προσπάθεια να τοποθετήσει αυτό το συμβάν με τους κανόνες που περιγράφονται παραπάνω. Εάν υπάρχουν περισσότεροι από ένας τρόπος για να συνεχίσει η αλυσίδα, εναλλακτικές λύσεις οι οποίες μειώνουν την ποινή ελαστικού περιορισμού περισσότερο δοκιμάζονται πρώτα. Αν ένα γεγονός μπορεί να τοποθετηθεί χωρίς να απορριφθεί κάποιο άλλο, η αλυσίδα έχει τελειώσει και είναι αποδεκτή αν μειώνει την συνολική ποινή ελαστικού περιορισμού. Εάν μη αποδεκτές αλυσίδες μπορούν να βρεθούν

στο μέγιστο μήκος της αλυσίδας, η αναζήτηση ματαιώνεται. Εφαρμόζουμε αυτή τη διαδικασία όταν ένα μυρμήγκι βρήκε μια νέα καλύτερη λύση και χρησιμοποίησε ένα μέγιστο μήκος αλυσίδας από 16 κινήσεις.

Παραλληλοποίηση

Οι ACO αλγόριθμοι είναι εύκολα παραλληλήσιμοι, δεδομένου ότι κάθε μυρμήγκι κατασκευάζει λύση του ανεξάρτητα. Μόνο η αξιολόγηση λύσης και η ενημέρωση φερομόνης πρέπει να είναι συγχρονισμένες αφού όλα τα μυρμήγκια έχουν ολοκληρώσει την κατασκευή της λύσης τους. Δοκιμάστηκε μια παράλληλη υλοποίηση του αλγορίθμου όπου οι παρτίδες των μυρμήγκιων τρέχουν σε πολλαπλά νήματα. Για την εφαρμογή αυτή δεν χρησιμοποιήθηκε η ευρετική βελτίωση περιγράφεται στην υποενότητα 1.4. Αντ' αυτού χρησιμοποιείται ένας προσομοιωμένος (SA) αλγόριθμος απόκτησης. Το ευρετικό που είναι βασισμένο στην αλυσίδα εκτίναξης έχει σχεδιαστεί για να δώσει μια καλύτερη "γρήγορη λύση", και ως εκ τούτου εξαντλεί μάλλον γρήγορα τον περιορισμένο χώρο αναζήτησης του. Δεν είναι επίσης ικανό να διαφύγει τα τοπικά ελάχιστα, και ως εκ τούτου δεν είναι κατάλληλο για να λειτουργεί παράλληλα με τον αλγόριθμο ACO. Έτσι, η μετάβαση σε SA.

Ο τελεστής γειτονιάς επιλέγει τυχαία δύο συνδυασμούς από χρονοθυρίδες και αίθουσες, που ο καθένας προσδιορίζει μια θυρίδα στο χρονοδιάγραμμα. Οι χρονοθυρίδες επιλέγονται έτσι ώστε οι δύο επιλεγμένες χρονοθυρίδες δεν είναι πανομοιότυπες. Το περιεχόμενο που σχετίζεται με αυτές τις σχισμές τότε ανταλλάσσεται. Υπάρχουν τρεις περιπτώσεις, η πρώτη είναι ότι και οι δύο θυρίδες είναι κενές, όπου υπόθεση αυτή δεν λαμβάνεται υπόψη. Η δεύτερη υπόθεση είναι ότι μία από τις θυρίδες είναι κενή και η άλλη έχει ένα σχετικό συμβάν. Αυτό έχει ως αποτέλεσμα, το συμβάν να μετακινηθεί. Και, τέλος, η τρίτη περίπτωση είναι ότι και οι δύο θυρίδες έχουν σχετικά συμβάντα, η οποία οδηγεί στην ανταλλαγή αυτών των συμβάντων. Οι κινήσεις που οδηγούν σε παραβιάσεις αυστηρών περιορισμών απορρίπτονται

αυτόματα, και κινήσεις που βελτιώνουν το SCP είτε το αφήνουν αμετάβλητο γίνονται αυτόματα αποδεκτές. Οι κινήσεις που μειώνουν το SCP είναι αποδεκτές με πιθανότητα $e^{-\Delta SCP/T_i}$, όπου T_i είναι η τρέχουσα θερμοκρασία. Είναι ρυθμισμένη αρχικά σε μια σταθερή τιμή και T_0 μειώσε κάθε τρίτη επανάληψη του ACO σύμφωνα προς το γεωμετρικό διάγραμμα ψύξεως $T_{i+1} = \delta T_i$.

Ο SA διεξάγεται σε ξεχωριστό νήμα παράλληλα με τον ACO. Αυτό το νήμα σταματά μετά από κάθε επανάληψη του ACO, όπου συγκρίνονται οι λύσεις και από τους δύο αλγορίθμους. Αν μια βελτιωμένη λύση βρίσκεται από τον SA περιλαμβάνεται στην ενημέρωση φερομόνης μια φορά, αν μια βελτιωμένη λύση βρίσκεται από τον ACO, ο SA ξαναρχίζει με αυτήν την βελτιωμένη λύση.

5. Ένα Elitist-Ant σύστημα για επίλυση προβλημάτων κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem)

1. Ο Αλγόριθμος

Γενικώς, ο Gambardella μεταφέρει τη συμπεριφορά των πραγματικών μυρμηγκιών σε έναν αλγόριθμο υβριδικού συστήματος αποικίας μυρμηγκιών με το να δημιουργεί μια αναλογία μεταξύ της αναζήτησης πραγματικών μυρμηγκιών και το σύνολο των εφικτών λύσεων στο πρόβλημα. Το ποσό των τροφίμων σε μια πηγή και το συνάρτηση καταλληλότητας, το μονοπάτι φερομόνης και μια προσαρμοστική μνήμη. Χρησιμοποίησαν μονοπάτια φερομόνης για εξερεύνηση και εκμετάλλευση. Στην αρχική περίπτωση, η φερομόνη επιλέγει το στοιχείο που χρησιμοποιείται για την κατασκευή μίας λύσης πιθανοκρατικά. Στην τελευταία, η φερομόνη

επιλέγει το στοιχείο που μεγιστοποιεί ένα μείγμα από τιμές από ίχνη φερομόνης και μερικά αντικειμενικές συναρτήσεις αξιολόγησης.

Σε αυτή την εργασία, θα γίνει επέκταση ενός αλγόριθμου συστήματος Elitist-Ant που προτείνει ο Gambardella, για την επίλυση του προβλήματος χρονοπρογραμματισμού μαθημάτων, που χρησιμοποιεί τη εντατικοποίηση και διαφοροποίηση των μηχανισμών όπως στον Gambardella. Εδώ χρησιμοποιείται επίσης μια εξωτερική μνήμη με ένα μεταβλητό μέγεθος για την αποθήκευση των ελίτ λύσεων σε κάθε επανάληψη, προκειμένου να διαπιστωθεί αν χρειάζεται ενημέρωση του μονοπατιού φερομόνης ή επανεκκίνηση της αναζήτησης με δημιουργία νέου πληθυσμού από τις ελίτ λύσεις. Με την χρησιμοποίηση και των δύο μηχανισμών, είμαστε σε θέση να διαχειριστούμε την ανώμαλη εξερεύνηση που προκαλείται από απεριόριστους εκ νέου συνδυασμούς. Ο ψευδό κώδικας του αλγόριθμου συστήματος Elitist-Ant παρουσιάζεται στην Εικόνα 13.

Σε αυτή την εργασία, χρησιμοποιήσαμε τους ίδιους τύπους που παρουσιάζονται στο έργο του Mayer σχετικά με το μοντέλο φερομόνης, την αναλογική δειγματοληψία φερομόνης, τις ενημέρωσεις πλέγματος μονοπατιών φερομόνης, την εξάτμιση φερομόνης, και τους κανόνες ενημέρωσης φερομόνης.

Ο αλγόριθμος ξεκινά με την αρχικοποίηση όλων των παραμέτρων όπως στον Πίνακα 9 (Στάδιο 1). Τότε (Στάδιο 2), κάθε μυρμήγκι κατασκευάζει την λύση από το μηδέν (άδεια χρονοδιαγράμματα) χρησιμοποιώντας την επιλογή του τροχό ρουλέτας καθοδηγείται από δύο πλέγματα μονοπατιών φερομόνης (τ_j^1 , τ_j^2) τα οποία παρουσιάζονται ως οι λίστες των διαθέσιμων χρονοθυρίδων και αιθουσών για ένα μάθημα. Χρησιμοποιούμε ένα συνδυασμό του μικρότερου βαθμού κορεσμού και του μεγαλύτερου βαθμού βάρους ευρετικών για να αντιπροσωπεύουν ένα μυρμήγκι για την κατασκευή μιας λύσης. Με άλλα λόγια, τα απρογραμμάτιστα μαθήματα ταξινομούνται με βάση τον μεγαλύτερο αριθμό των εγγραφών

και το μικρότερο αριθμό των διαθέσιμων χρονοθυρίδων. Κατά τη φάση της κατασκευής, εκπροσωπούμε τα πλέγματα φερομόνης τ_j^l και τ_j^r ως ο βάρος της προτιμότερης (με ελάχιστο κόστος ποινής) ανάθεσης του μαθήματος j στην χρονοθυρίδα p , και το μάθημα j στην αίθουσα k με βάση το ιστορικό των προηγούμενων εργασιών (δηλαδή την εμπειρία των προηγούμενων μυρμηγκιών). Όσο για την φάση βελτίωσης, αντιπροσωπεύουμε την φερομόνη ως το βάρος της προτιμότερης (σημαντική βελτίωση μετάθεσης) αλληλουχίας γειτονικών δομών μίας λύσης (μηχανισμός εντατικοποίησης).

Step 1: Initialization phase

While Stopping Criterion is not met do

Step 2: Construction phase

for each ant //solution construction

Assign all courses into feasible timeslots and rooms using roulette wheel selection mechanism guided by pheromone information

end for

Step 3: Improvement phase

while non-improvement criterion is not met, **do** //max number of stagnation iterations $n_{stagnation}$

Locally improve each constructed solution // iterated local search routine

Update size and content of external memory // update size & contents

End while

Step 4: Intensification phase

Apply intensification mechanism

Step 5: Global pheromone update phase

Update pheromone trails for assignments appearing in solution

Step 6: Diversification phase

Evaporate pheromone matrix // diversity control

Apply diversification mechanism

End while

Step 7: Return best ant // best solution

Εικόνα 13: Ο ψευδοκώδικας του αλγορίθμου Elitist-Ant

Αφού όλες οι λύσεις είναι εφικτές, θα τις βελτιωθούν με μία τοπική επαναλαμβανόμενη αναζήτηση (ILS) που καθοδηγείται από το πλέγμα του μονοπατιού φερομόνης (Στάδιο 3). Η ILS υλοποιεί απλά πέντε τύπους κινήσεων διαταραχής (βλ. Πίνακα 10), όπου κάθε κίνηση

εφαρμόζεται για ένα προκαθορισμένο αριθμό επαναλήψεων (χρησιμοποιούμε 100 σε αυτή την περίπτωση) για να προσδιοριστεί η αντοχή της διαταραχής (αυξάνοντας ή μειώνοντας τιμές φερομόνης). Χρησιμοποιούμε απλή καθοδική ευρετική μέθοδο. Μία νέα λύση είναι αποδεκτή, αν είναι καλύτερη από την τρέχουσα λύση. Μετά την εφαρμογή της τοπικής αναζήτησης, ενημερώνουμε τα περιεχόμενα και το μέγεθος της εξωτερικής μνήμης που περιέχει τις ελίτ λύσεις που βρέθηκαν (Στάδιο 3), καθόσον στον Mayer. Μια ελίτ λύση είναι μια λύση που έχει την καλύτερη αξία και το λιγότερο αριθμό των μη προγραμματισμένων μαθημάτων (αν υπάρχουν). Η εξωτερική μνήμη αποθηκεύει τις ελίτ λύσεις που βρίσκονται. Χρησιμοποιείται για να διατηρηθεί μία καλή εμπειρία πληροφοριών αναζήτησης για να καθοδηγήσει την αναζήτηση αποτελεσματικά. Επίσης, χρησιμοποιείται για την κατασκευή νέων λύσεων σε διαδοχικές επαναλήψεις από ελίτ λύσεις (με την εκτέλεση κάποιων διαταραχών στις ελίτ λύσεις) και όχι την κατασκευή τους από το μηδέν. Μετά από κάθε επανάληψη, το μυρμήγκι με το χαμηλότερο αριθμό των μη προγραμματισμένων μαθημάτων (αν υπάρχει) και ένα καλύτερο από το μέσο όρο τιμές καταλληλότητας θα προσθέσει μία ορισμένη ποσότητα φερομόνης για να καθοδηγήσει την μετέπειτα αναζήτηση μυρμηγκιού για καλύτερη λύση.

Στο (Στάδιο 4), ο μηχανισμός εντατικοποίησης χρησιμοποιείται για τη διερεύνηση της γειτονιάς των καλών λύσεων πιο αποτελεσματικά αφού έχει βελτιωθεί η καλύτερη λύση που έχει βρεθεί μέχρι σήμερα. Αυτό γίνεται με τη χρήση μίας τυχαίας καθοδικής ευρετικής, όπου κάποιοι γείτονες από όλες τις γειτονιές (βλ. Πίνακα 10) μίας λύσης παράγονται και η καλύτερη επιλέγεται έως ότου δεν είναι δυνατή παραπάνω βελτίωση. Στη συνέχεια, όλα τα μυρμήγκια ξεκινούν την επόμενη επανάληψη με την καλύτερη διαταραχή που γίνεται για την κατασκευή νέων λύσεων από τις προηγούμενες (όπως στο Βήμα 2 αλλά με την εκτέλεση διαταραχών στις ελίτ λύσεις και όχι την κατασκευή νέων λύσεων από το μηδέν).

Μόνο το καλύτερο μυρμήγκι (ελίτ λύση) ενημερώνει το πλέγμα μονοπατιών φερομόνης (παγκόσμια ενημέρωση) (Στάδιο 5). Επιπλέον, ο καλύτερος γείτονας που εκτελείται (που έχει ως αποτέλεσμα μια ελίτ λύση) από μια καλύτερη λύση στην τρέχουσα επανάληψη, θα πρέπει να χρησιμοποιείται ως η αρχική διαταραχή των λύσεων στην επόμενη επανάληψη στη φάση βελτίωσης. Αυτό γίνεται για να παραχθεί περαιτέρω εντατικοποίηση γύρω από ελπιδοφόρες περιοχές κατά την κατασκευή νέων λύσεων. Σε αυτήν την εργασία, χρησιμοποιούμε πέντε γειτονικές δομές όπως απεικονίζεται στον Πίνακα 10. Οι γειτονικές δομές απλά εξερευνούνται χρησιμοποιώντας μία απλή καθοδική ευρετική (Στάδιο 3). Σε κάθε επανάληψη, μια γειτονιά (επιλέγεται τυχαία) απασχολείται. Αν υπάρχει μια βελτίωση, η γειτονιά θα χρησιμοποιηθεί και πάλι σε διαδοχικές επαναλήψεις έως ότου δεν υπάρξει βελτίωση, τότε μια διαφορετική τυχαία επιλεγμένη γειτονιά (όχι από την προηγούμενη) χρησιμοποιείται. Αυτό το βήμα προχωρά για ένα προκαθορισμένο αριθμό επαναλήψεων.

N_s	Description
N_1	Move a randomly selected course to a random feasible room and timeslot (Socha; 2003).
N_2	Swap timeslots and rooms of two randomly selected courses (with maintaining feasibility) (Socha; 2003).
N_3	Randomly select two timeslots and rooms and swap all courses between these timeslots and rooms so that they are still in feasible timeslot and room pair where the course will not violate any hard constraints (Socha; 2003).
N_4	Select randomly two timeslots, say t_i and t_j (where $j>i$) where the timeslots are ordered t_1, t_2, \dots, t_{45} . Take all courses in t_i and assign them in t_j . Move all courses in t_j and assign them in t_{j-1} . Then assign all courses that were in t_{j-1} to t_{j-2} and so on until all courses that were in t_{i+1} are assigned to t_i and terminate the process (Alvarez-Valdes <i>et al.</i> 2002).
N_5	Move the highest penalty cost course from a random 10% selection of the courses to a random feasible timeslot (Abdullah <i>et al.</i> ; 2007a).

Πίνακας 10: Γειτονιές που ορίζονται από το πρόβλημα.

Ο μηχανισμός της διαφοροποίησης (Στάδιο 6), που ονομάζεται επίσης εξάτμιση φερομόνης, χρησιμοποιείται μετά την εκτέλεση ενός προκαθορισμένου αριθμού από επαναλήψεις μη βελτίωσης της καλύτερης λύση που βρέθηκε μέχρι σήμερα στην τοπική ρουτίνα αναζήτησης.

Αυτό βοηθά στην αποφυγή την πρόωρης σύγκλισης του αλγορίθμου. Ο μηχανισμός περιοδικά (ο μηχανισμός διαφοροποίησης απασχολείται κάθε φορά) σβήνει όλα τα μονοπάτια φερομόνης. Αρχικοποιούνται εκ νέου οι τιμές φερομόνης όταν ο μηχανισμός εντατικοποίησης αποτυγχάνει να βελτιώσει όλες τις λύσεις μυρμηγκιών. Με άλλα λόγια, αν όλες οι γειτονιές στον Πίνακα 10 διερευνηθούν και καμία βελτίωση λύσης δεν έχει γίνει, τότε τα μονοπάτια φερομόνης πρέπει να διαγραφούν για να ακολουθήσουν μια νέα πορεία προς καλύτερες ποιοτικές λύσεις. Κάθε μυρμηγκί θα δημιουργήσει μια νέα λύση από την ελίτ λύση στην εξωτερική μνήμη για όλα τα μυρμηγκία (εκτός από εκείνα με την καλύτερη λύση που βρέθηκε μέχρι τώρα) εκτελώντας κάποιες διαταραχές στις ελίτ λύσεις. Η όλη διαδικασία του αλγορίθμου επαναλαμβάνεται μέχρις ότου πληρείται το κριτήριο διακοπής, είτε η καλύτερη λύση έχει βρεθεί ή ο αριθμός των επαναλήψεων φθάσει στα όριά του.

2. Πειραματικά αποτελέσματα

Όπως συνιστάται από τον Mayer και τα προκαταρκτικά πειράματα, έχουμε ρυθμισμένες παραμέτρους όπως απεικονίζεται στον Πίνακα 11 πιο κάτω. Σε αυτή την εργασία, δοκιμάστηκε ο αλγόριθμος σε γνωστά σημεία αναφοράς για περιπτώσεις κατάρτισης Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών.

Τρέξαμε τον αλγόριθμος 25 φορές σε κάθε περίπτωση για το χρόνο εκτέλεσης που κυμαίνεται μεταξύ 30 δευτερόλεπτα και 12 ώρες για κάθε εκτέλεση ανάλογα με το μέγεθος και την πολυπλοκότητα της περίπτωσης προς απόκτηση της πιθανής καθολικής βέλτιστη λύση (ένα πολύ μεγαλύτερο χρονικό διάστημα από αυτό που επιτρέπεται στον διαγωνισμό).

Τα πειράματα πραγματοποιήθηκαν σε Intel Pentium Core 2 Duo 2,16 GHz επεξεργαστή, 2GB RAM, και υλοποιείται σε Java NetBeans IDE v 6,8.

Parameter	Description and Value
n_{ant}	number of dispatched ants (solutions) per iteration =20
n_{iter}	number of Iterations= 100,000
$n_{stagiter}$	number of non-improvement iterations =100
τ^r and τ^t	pheromone matrices initial values of rooms and timeslots =0.5
ρ	pheromone evaporation rate =0.25 , $\rho \in [0,1]$
α	importance of pheromone controlling (exploration) ratio =1.0, $\alpha \in [1.0,1.1]$
ψ	importance of soft constraints (penalty) =0.3
N_s	number of employed neighborhood structures per solution =5
k_m	initial external memory size =5 (elite solutions)
<i>Elitism</i>	use the best ant (for each iteration) to update global pheromone.

Πίνακας 11: Ρυθμίσεις παραμέτρων για τον αλγόριθμο.

Ο Πίνακας 12 παρουσιάζει τα αποτελέσματα που ελήφθησαν από τον AS αλγόριθμο με βάση τις παραμέτρους που μας παρουσιάζονται στον Πίνακα 11, σε σύγκριση με άλλες προσεγγίσεις ACO, βάσει του πληθυσμού προσεγγίσεις και τα καλύτερα γνωστά αποτελέσματα που λαμβάνονται με άλλες. Τα καλύτερα αποτελέσματα παρουσιάζονται με έντονους χαρακτήρες. Οι στατιστικές μετρήσεις βασίζονται στους ακόλουθους δείκτες απόδοσης κάτω από μια χαλαρή κατάσταση διακοπής, που είναι ο αριθμός των επαναλήψεων: το καλύτερο αποτέλεσμα που λαμβάνεται (*best*), η μέση τιμή (*m*), η τυπική απόκλιση (σ), τα κατώτερα (*Q1*) και το ανώτερα (*Q2*) τεταρτημόρια κατανομών των αποτελεσμάτων και η χειρότερη τιμή κόστους (*w*), που λαμβάνεται από το συνολικό αριθμό των εκτελέσεων για κάθε παράδειγμα. Επιπλέον, τα καλύτερα αποτελέσματα (*best**) που ελήφθησαν από τον αλγόριθμο υπό την κατάσταση διακοπής του διαγωνισμού, η οποία είναι ο περιορισμένος χρόνος ίσος με 474 δευτερόλεπτα. Μερικές από τις μεθόδους που παρουσιάζονται στον Πίνακα 12 μοιράζονται κοινά χαρακτηριστικά με το καθένα άλλα και κάπως με τον αλγόριθμο μας, όπως είναι τα χαρακτηριστικά με βάση τον πληθυσμό, ο μεγάλος αριθμός των επαναλήψεων (κυμαίνεται από 50.000 έως 200.000), οι μεταβλητές

γειτονικές δομές, μηχανισμοί ελέγχου, και χαλαρή κατάσταση διακοπής, στην οποία προσπαθήσαμε να κάνουμε λογικές, δίκαιες και ουσιαστικές συγκρίσεις.

Αποτελέσματα από τον Πίνακα 12 έδειξαν ότι ο Elitist-AS αλγόριθμος είναι ικανός να παράγει εφικτά χρονοδιαγράμματα για όλες τις περιπτώσεις σε κάθε εκτέλεση. Παρά το γεγονός ότι ο χρόνος εκτέλεσης του αλγόριθμου ήταν κάπως μικρός σε όταν λαμβάνονται υπόψη τα χρονικά όρια που επιτρέπονται στον διαγωνισμό, τα αποτελέσματά μας είναι κάπως ανταγωνιστικά σε σχέση με τις προσεγγίσεις του ACO. Φυσικά, αν εκτελεστεί ο αλγόριθμος για μεγαλύτερο χρονικό διάστημα θα είμαστε σίγουρα σε θέση να αποκτήσουμε καλύτερη σχέση κόστους αξίας μίας λύσης ή ακόμη και να αποκτήσουμε την καθολική βέλτιστη λύση. Ως εκ τούτου, έχουμε επέκτεινε το χρόνο εκτέλεσης για να επιτευχθούν καλύτερα αποτελέσματα. Επίσης, μπορεί να αποδειχθεί ότι ο αλγόριθμος είναι σχετικά συνεπής στην παραγωγή βέλτιστων αποτελεσμάτων για όλες τις περιπτώσεις μικρού μεγέθους.

Ο αλγόριθμος μας έχει παράγει εφικτά και καλής ποιότητας αποτελέσματα για όλες τις περιπτώσεις, και η ποιότητα τους ξεπέρασε όλες τις προσεγγίσεις που βασίζονται στον πληθυσμό που απεικονίζονται στον πίνακα για όλες τις περιπτώσεις, εκτός από τις *medium3* και *large* περιπτώσεις. Ο αλγόριθμος λαμβάνει τα καλύτερα αποτελέσματα επικεντρωμένα σε όλες τις περιπτώσεις μεσαίου μεγέθους σε σύγκριση μόνο με τις προσεγγίσεις στον πίνακα.

Επιπλέον, τα αποτελέσματα του Elitist-AS αλγόριθμου που λαμβάνονται στο πλαίσιο της κατάστασης διακοπής του διαγωνισμού φτάνουν το βέλτιστο για όλες τις μικρές περιπτώσεις σε χρόνο μικρότερο από 300 δευτερόλεπτα, καθώς και η περίπτωση των αποτελεσμάτων που λαμβάνονται υπό την χαλαρή κατάσταση διακοπής, ενώ όλα τα αποτελέσματα μεσαίων και μεγάλων περιπτώσεων είναι σχετικά καλύτερα από ό, τι μερικές από τις προσεγγίσεις για την απόκτηση καλής ποιότητας αποτελέσματα μέσα σε συντομότερο υπολογιστικό χρόνο στις περισσότερες περιπτώσεις. Ο αλγόριθμος λαμβάνει καλύτερης ποιότητας αποτελέσματα από

όλες τις εικονογραφημένες προσεγγίσεις βασισμένες σε σύστημα μυρμηγκιών και πληθυσμού στον ανωτέρω πίνακα, σε σχέση με τα καλύτερα γνωστά αποτελέσματα που αναφέρονται στη βιβλιογραφία. Επιπλέον, οι στατιστικές αναγνώσεις από τον Πίνακα 12 αποδεικνύουν ότι ο αλγόριθμος (Elitist-AS) είναι αποτελεσματικός και αξιόπιστος.

Instance	Our Elitist-Ant System algorithm							ACO based approaches		Population-based approaches		
	best*	best	Q1	σ	m	Q2	w	MM AS	DH CA	EGS GA	EM GD	HEA
small1	0	0	0	.79	1	2	2	1	5	0	0	0
small2	0	0	0	.913	1	2	3	3	5	0	0	0
small3	0	0	1	1.262	1	2	4	1	3	0	0	0
small4	0	0	0	.586	0	1	2	1	3	0	0	0
small5	0	0	0	.5	0	1	1	0	0	0	0	0
medium1	190	84	87	6.837	92	97	110	195	176	139	96	221
medium2	223	82	85	8.542	90	98	109	184	154	92	96	147
medium3	259	123	128	11.876	134	146	160	248	191	122	135	246
medium4	127	62	68	10.649	76	83	99	164.5	148	98	79	165
medium5	132	75	82	7.659	87	92	102	219.5	166	116	87	130
large	869	690	719	50.868	750	783	877	851.5	798	615	683	529

Approaches' results:

- MMAS: the Max-Min Ant System in (Socha *et al.*; 2002).
- DHCA: the Die Hard Cooperative Ant behavior algorithm in (Ejaz and Javed; 2007).
- EGSGA: the Extended Guided Search Genetic Algorithm in (Jat and Yang; 2010).
- EMGD: the hybrid of Electromagnetic-Like mechanism with force decay Rate Great Deluge in (Abdullah *et al.*; 2009).
- HEA: the Hybrid Evolutionary Algorithm – GA operators and Randomized Iterative Improvement in (Abdullah *et al.*; 2007b).

Πίνακας 13: Υπολογιστική στατιστικά στοιχεία του αλγορίθμου.

Κεφάλαιο 4

Κατάρτιση Προγράμματος Εξεταστικής.

Γενικά

Η κατάρτιση προγράμματος εξεταστικής είναι μία από τις πιο σημαντικές διοικητικές δραστηριότητες που λαμβάνει χώρα σε όλα τα ακαδημαϊκά ιδρύματα. Στην παρούσα εργασία, παρουσιάζουμε μια έρευνα για την κατάρτιση προγράμματος εξεταστικής που έχει λάβει χώρα την τελευταία δεκαετία. Αυτά τα τελευταία δέκα χρόνια έχει παρατηρηθεί ένα σημαντικά αυξημένο επίπεδο της προσοχής της έρευνας για το αυτόν τον σημαντικό τομέα. Υπήρξε μια σειρά από διορατικές εισφορές στην επιστημονική βιβλιογραφία τόσο από την άποψη των θεωρητικών ζητημάτων όσο και των πρακτικών πτυχών. Ο κύριος στόχος αυτής της έρευνας είναι να αναδείξει τις νέες τάσεις και τα βασικά επιτεύγματα της έρευνας τα οποία έχουν πραγματοποιηθεί την τελευταία δεκαετία. Στόχος είναι επίσης να περιγραφούν μια σειρά από σημαντικά ζητήματα σχετικής έρευνας και προκλήσεις που έχουν δημιουργηθεί από την εργασία.

Ορίζεται πρώτα το πρόβλημα και μελετώνται προηγούμενα έγγραφα της έρευνας. Μετά την παρουσίαση των state-of-the-art μεθοδολογιών, τονίζονται πρόσφατες τάσεις της έρευνας, συμπεριλαμβανομένων υβριδισμοί των μεθοδολογιών έρευνας και της ανάπτυξης τεχνικών που υποκινούνται από την αύξηση του επιπέδου της γενικότητας με την οποία μεθοδολογίες αναζήτησης μπορούν να λειτουργήσουν. Συνοψίζοντας οι πίνακες που παρουσιάζονται, παρέχουν μια συνολική εικόνα των τεχνικών αυτών. Τέλος, επιστούμε την συζήτησή μας στην βιβλιογραφία για να παρουσιάσει ένα (μη εξαντλητικό) φάσμα των πιθανών κατευθύνσεων μελλοντικής έρευνας και ανοικτά ζητήματα της έρευνας για την κατάρτιση προγράμματος εξετάσεων.

Τα προβλήματα χρονοδιαγραμμάτων συναντώνται σε διάφορες μορφές, συμπεριλαμβανομένων των εκπαιδευτικών, τον προγραμματισμό νοσηλείας, αθλητικά χρονοδιαγράμματα και τα χρονοδιαγράμματα μεταφορών. Έχουν αντιπροσωπεύσει μία δύσκολη και σημαντική περιοχή προβλήματος για τους ερευνητές. Τα τελευταία χρόνια παρατηρείται ένα αυξημένο επίπεδο της ερευνητικής δραστηριότητας αυτό τον τομέα. Αυτό

αποδεικνύεται από (μεταξύ άλλων) τη ανάδυση μιας σειράς διεθνών συνεδρίων για την Πρακτική και Θεωρία στον Αυτοματοποιημένο Χρονοπρογραμματισμό (PATAT) και τη δημιουργία της ομάδας EURO (European Association of Operational Research Societies) στα αυτοματοποιημένα χρονοδιαγράμματα.

Μεταξύ της μεγάλης ποικιλίας των προβλημάτων χρονοδιαγράμματος, ο εκπαιδευτικός χρονοπρογραμματισμός είναι ένα από τα πλέον ευρέως μελετημένα. Είναι μία από τις πιο σημαντικές και χρονοβόρες εργασίες που συμβαίνουν περιοδικά (π.χ. κάθε χρόνο, σε τριμηνιαία βάση, κ.λπ.) σε όλα τα ακαδημαϊκά ιδρύματα. Η ποιότητα του χρονοπρογραμματισμού έχει μεγάλη επίδραση σε ένα ευρύ φάσμα των διαφόρων ενδιαφερομένων, συμπεριλαμβανομένων διδασκόντων, φοιτητών και διοικητικών υπαλλήλων. Παραλλαγές του εκπαιδευτικού χρονοπρογραμματισμού περιλαμβάνουν χρονοπρογραμματισμό σχολείων (χρονοπρογραμματισμός classteacher), χρονοπρογραμματισμό μαθημάτων πανεπιστημίου, χρονοπρογραμματισμό εξετάσεων και ανάθεση αιθουσών. Έχει παρατηρηθεί ότι ο χρονοπρογραμματισμός μαθημάτων και εξετάσεων είναι προβλήματα σχετικά, αλλά υπάρχουν και πολύ σημαντικές διαφορές. Η εργασία θα επικεντρωθεί στην κατάρτιση προγράμματος εξεταστικής.

Εντούτοις, ένας σημαντικός αριθμός εγγράφων έρευνας για το θέμα αυτό έχουν δημοσιευθεί από το 1996. Η εργασία αυτή θα επικεντρωθεί από την έρευνα που έχει εμφανιστεί μετά από τη δημοσίευση του (Carter και Laporte 1996). Η τελευταία δεκαετία έχει δημιουργηθεί μια συλλογή από συγκριτικά μέτρησης επιδόσεων σε χρονοδιαγράμματα εξετάσεων που έχουν χρησιμοποιηθεί από πολλούς στις ερευνητικές εργασίες που έχουν εμφανιστεί από το 1996. Ωστόσο, υπήρξε κάποια σύγχυση στη βιβλιογραφία που προκαλείται από την ύπαρξη διαφορετικών συνόλων δεδομένων αναφοράς πρόβλημα με τα ίδια ονόματα. Η εργασία αυτή έχει ως στόχο να εξαλείψει τέτοια σύγχυση, παρουσιάζοντας μια οριστική μετονομασία των συνόλων και αποσαφηνίζοντας την κατάσταση.

Τα προβλήματα κατάρτισης προγράμματος εξεταστικής.

Τα προβλήματα κατάρτισης προγράμματος εξεταστικής μπορούν να οριστούν ως μία ανάθεση ενός συνόλου των εξετάσεων σε περιορισμένο αριθμό χρονοθυρίδων (χρονικά διαστήματα) και αιθουσών (ορισμένης ικανότητας), που υπόκεινται σε ένα σύνολο περιορισμών. Η πολυπλοκότητα και τις προκλήσεις που παρουσιάζονται από τα προβλήματα χρονοδιαγράμματος προκύπτουν από το γεγονός ότι μια μεγάλη ποικιλία από προβλήματα, μερικά από τα οποία έρχονται σε αντίθεση μεταξύ τους, πρέπει να πληρούνται σε διάφορα ιδρύματα. Στη βιβλιογραφία χρονοπρογραμματισμού, οι περιορισμοί είναι συνήθως κατηγοριοποιούνται σε δύο τύπους: αυστηρούς και ελαστικούς περιορισμούς, οι οποίοι εξηγούνται παρακάτω:

- Οι *Αυστηροί Περιορισμοί* δεν μπορεί να παραβιαστούν υπό οποιεσδήποτε συνθήκες (κυρίως λόγω των φυσικών περιορισμών). Για παράδειγμα, αντικρουόμενες εξετάσεις (δηλαδή εκείνες που αφορούν κοινούς πόρους, όπως φοιτητές) δεν μπορούν να προγραμματιστούν ταυτόχρονα. Ένα χρονοδιάγραμμα που πληροί όλους τους αυστηρούς περιορισμούς συνήθως λέγεται ότι είναι εφικτό.
- Οι *Ελαστικοί περιορισμοί* είναι επιθυμητοί, αλλά δεν είναι απολύτως κρίσιμοι. Στην πράξη, είναι συνήθως αδύνατο να βρει εφικτές λύσεις που ικανοποιούν όλους τους ελαστικούς περιορισμούς. Οι ελαστικοί περιορισμοί διαφέρουν (και μερικές φορές έρχονται σε σύγκρουση ο ένας με τον άλλο) από το ένα ίδρυμα στο άλλο όσον αφορά τόσο το είδος όσο και τη σημασία τους. Ο πιο κοινός ελαστικός περιορισμός στην βιβλιογραφία χρονοπρογραμματισμού εξετάσεων είναι να εξαπλωθούν αντικρουόμενες εξετάσεις όσο το δυνατόν περισσότερο σε όλη τη διάρκεια εξέτασης, έτσι ώστε οι μαθητές να έχουν αρκετό χρόνο για επανάληψη μεταξύ των εξετάσεων. Ένα παράδειγμα από έναν άλλο ελαστικό περιορισμό που μπορεί να έρχεται σε

σύγκρουση με αυτό είναι να προγραμματιστούν όλες οι μεγάλες εξετάσεις όσο το δυνατόν νωρίτερα για να δοθεί επαρκής χρόνος για σήμανση. Η ποιότητα των χρονοδιαγραμμάτων συνήθως μετριέται από τον έλεγχο σε ποιο βαθμό οι ελαστικοί περιορισμοί παραβιάζονται στις λύσεις που δημιουργούνται.

Λόγω της μεγάλης ποικιλίας των προβλημάτων που παρουσιάζονται και διερευνώνται, δεν θα ήταν ούτε πρακτικό ούτε ωφέλιμο να παρουσιάσουμε έναν πλήρη κατάλογο όλων των αυστηρών και ελαστικών περιορισμών που προκύπτουν στην έρευνα χρονοδιαγραμμάτων. Παραθέτουμε μερικούς από τους βασικούς αυστηρούς και ελαστικούς περιορισμούς για την κατάρτιση προγράμματος εξετάσεων στους πίνακες 14 και 15, αντίστοιχα. Πιστεύουμε ότι αυτοί καλύπτουν τους περισσότερους από τους περιορισμούς που έχουν εμφανιστεί στη βιβλιογραφία. Μπορεί να παρατηρηθεί ότι μπορούν να ομαδοποιηθούν χονδρικά ως χρονικά εξαρτώμενοι (No. 1 στον Πίνακα 14 και No. 1-7 στον πίνακα 15), ή εξαρτώμενοι από πόρους (No 2 στον Πίνακα 14 και No. 8-11 στον Πίνακα 15) . Τα περισσότερα από τα έγγραφα της έρευνας στην επόμενη υποενότητα παρουσιάζουν λίστες περιορισμών σε εξετάσεις και γενικά χρονοδιαγράμματα. Οι αυστηροί περιορισμοί που αναφέρονται στον Πίνακα 14 και ο πρώτος ελαστικός περιορισμός στον Πίνακα 15 είναι εκείνοι οι οποίοι έχουν ως επί το πλείστον καλυφθεί από την έρευνα στην βιβλιογραφία.

Πίνακας 14. Βασικοί αυστηροί περιορισμοί σε προβλήματα κατάρτισης προγράμματος σπουδών εξεταστικής.

Βασικοί Αυστηροί Περιορισμοί

1. Δεν αναθέτονται ταυτόχρονα εξετάσεις με κοινούς πόρους (π.χ. φοιτητές)
2. Οι πόροι των εξετάσεων πρέπει να είναι επαρκής (π.χ. το μέγεθος των εξετάσεων πρέπει να είναι κάτω από την χωρητικότητα των αιθουσών, αρκετές αίθουσες για όλες τις εξετάσεις)

Πίνακας 15. Βασικοί ελαστικοί περιορισμοί σε προβλήματα κατάρτισης προγράμματος σπουδών εξεταστικής.

Βασικοί Ελαστικοί Περιορισμοί

1. Εξάπλωση των αντικρουόμενων εξετάσεων, όσο το δυνατόν περισσότερο, ή όχι σε x συνεχόμενες χρονοθυρίδες ή ημέρες
2. Οι ομάδες εξετάσεων απαιτούνται να λάβουν χώρα κατά την ίδια ώρα, την ίδια ημέρα ή σε μία θέση
3. Οι εξετάσεις να είναι συνεχόμενες
4. Ο προγραμματισμός όλων των εξετάσεων, ή των μεγαλύτερων εξετάσεων, όσο το δυνατόν νωρίτερα
5. Κατηγοριοποίηση (προτεραιότητα) των εξετάσεων πρέπει να πληρείται
6. Περιορισμένος αριθμός φοιτητών και / ή εξετάσεις σε οποιαδήποτε χρονική στιγμή
7. Χρονικές απαιτήσεις (π.χ. εξετάσεις να είναι (ή όχι) σε ορισμένες χρονοθυρίδες)
8. Αντικρουόμενες εξετάσεις την ίδια ημέρα που βρίσκονται σε κοντινή απόσταση
9. Οι εξετάσεις μπορεί να χωριστούν σε παρόμοιες θέσεις

10. Μόνον εξετάσεις της ίδιας διάρκειας μπορούν να συνδυαστούν στην ίδια αίθουσα

11. Απαιτήσεις σε πόρους (π.χ. εγκατάσταση δωμάτιο)

Προσεγγίσεις / Τεχνικές Κατάρτισης Προγράμματος Εξεταστικής

Υπήρξε αρκετή έρευνα για χρονοπρογραμματισμό εξεταστικής κατά την τελευταία δεκαετία. Σε αυτή την ενότητα κατατάσσεται η μεγάλη ποικιλία των τεχνικών έρευνας που έχουν εμφανιστεί σε αυτό το διάστημα. Σημειώνουμε ότι πολλές από τις επιτυχημένες μεθοδολογίες που έχουν εμφανιστεί στη βιβλιογραφία αντιπροσωπεύουν υβριδισμούς ενός αριθμού τεχνικών. Έτσι, η κατάταξη δεν είναι αυστηρή. Πράγματι, αρκετές από τις μεθοδολογίες μπορούσε να είχαν εμφανιστεί σε δύο ή περισσότερες από τις ταξινομήσεις. Όπου είναι δυνατόν, έχουν ταξινομηθεί από την κύρια τεχνική που χρησιμοποιείται.

1. Διαδοχικές Τεχνικές Βασισμένες σε Γράφους

Τα προβλήματα χρονοπρογραμματισμού, χωρίς ελαστικούς περιορισμούς, μπορούν να μοντελοποιηθούν ως προβλήματα χρωματισμού γραφήματος. Η δημοσίευση των Welsh και Powell το 1967 έχτισε τη γέφυρα μεταξύ του χρωματισμού γραφήματος και χρονοπρογραμματισμού, το οποίο οδήγησε σε αρκετά περισσότερη έρευνα αργότερα σε ευρετικά γραφήματα στον χρονοπρογραμματισμό. Σε προβλήματα χρονοπρογραμματισμού εξετάσεων, οι εξετάσεις μπορούν να εκπροσωπούνται από κορυφές σε ένα γράφημα, και ο αυστηρός περιορισμός μεταξύ εξετάσεων εκπροσωπείται από τις άκρες μεταξύ των κορυφών. Το πρόβλημα χρωματισμού γραφήματος της ανάθεσης χρωμάτων στις κορυφές, έτσι ώστε οι γειτονικές κορυφές να μην έχουν το ίδιο χρώμα, τότε αντιστοιχεί στο πρόβλημα της ανάθεσης χρονοθυρίδων σε εξετάσεις. Διαφορετικοί ελαστικοί περιορισμοί (όπως αυτοί που

παρατίθενται στον Πίνακα 15), πρέπει να εξεταστούν ξεχωριστά και ο βαθμός ο οποίος η λύση χρονοδιαγράμματος ικανοποιείται αντιπροσωπεύει ένα μέτρο της ποιότητας της λύσης.

Οι βασικοί χρωματισμοί γράφων με βάση ευρετικά χρονοδιαγράμματος είναι εποικοδομητικές μέθοδοι που ταξινομούν τις εξετάσεις και τις εκχωρούν μία προς μία με το πόσο δύσκολο είναι να προγραμματιστούν στις χρονοθυρίδες. Υπάρχει μια σειρά από στρατηγικές ταξινόμησης και οι τροποποιημένες παραλλαγές τους που εμφανίζονται στην βιβλιογραφία χρονοπρογραμματισμού. Στον Πίνακα 16, απεικονίζονται μερικές από τις στρατηγικές ταξινόμησης που χρησιμοποιούνται ευρέως. Μία τυχαία μέθοδος ταξινόμησης έχει επίσης χρησιμοποιηθεί στην βιβλιογραφία για να εισαγάγει τυχαιότητα σε υβριδικές προσεγγίσεις και για να παρέχει συγκρίσεις.

Πίνακας 16: Ευρέως Μελετημένες Στρατηγικές Ταξινόμησης σε Ευρετικά Γράφων για την κατάρτιση προγράμματος εξεταστικής.

Ευρετικά (Heuristics)	Στρατηγική Ταξινόμησης(Ordering Strategy)
Saturation Degree	Αύξουσα από τον αριθμό των χρονοθυρίδων που είναι διαθέσιμες για τις εξετάσεις στο χρονοδιάγραμμα για την ώρα
Largest Degree	Φθίνουσα από τον αριθμό των συγκρούσεων που έχουν κάποιες εξετάσεις με άλλες
Largest Weighted Degree	Όπως στον Largest Degree αλλά σταθμισμένο από τον αριθμό των εμπλεκόμενων φοιτητών
Largest Enrolment	Φθίνουσα με τον αριθμό των εγγραφών για τις εξετάσεις
Random Ordering	Τυχαία ταξινομεί τις εξετάσεις
Color Degree	Φθίνουσα από τον αριθμό των συγκρούσεων που έχει μία εξέταση με αυτές που έχουν προγραμματιστεί

Τα ευρετικά με βάση τα γραφήματα ως απλές εποικοδομητικές μέθοδοι έπαιξαν ένα πολύ σημαντικό ρόλο κατά τις πρόσφατες ημέρες στην έρευνας χρονοδιαγραμμάτων. Αν και αρχικά παρουσιάζονται ως τεχνικές (αν και απλές) από μόνες τους, εξακολουθούν να εφαρμόζονται και να προσαρμόζονται στην τρέχουσα ερευνητική βιβλιογραφία. Μεγάλη δύναμη τους είναι ότι μπορούν να προσφέρουν αρκετά καλά αποτελέσματα μέσα σε ένα

μικρό υπολογιστικό χρόνο και είναι πολύ εύκολο να εφαρμοστούν. Συχνά χρησιμοποιούνται για την κατασκευή αρχικών λύσεων, ή να οικοδομήσουν καλές μερικές λύσεις πριν από την εφαρμογή τεχνικών βελτίωσης. Σε ένα πρόσφατο άρθρο από τον **Burke** επισκοπούνται τεχνικές χρωματισμού γραφημάτων.

Οι **Carter, Laporte** και **Lee** το 1996, μελέτησαν τις πρώτες πέντε στρατηγικές ταξινόμησης στον Πίνακα 16 για τα πραγματικά προβλήματα και τα προβλήματα χρονοδιαγράμματος εξετάσεων που δημιουργούνται τυχαία. Οι μεγαλύτερες ομάδες, οι οποίες είναι τα μεγαλύτερα υπό-γραφήματα, όπου κάθε μία από τις κορυφές είναι δίπλα σε όλες τις άλλες, χρησιμοποιήθηκαν για την κατασκευή αρχικών λύσεων, με βάση με ποια ευρετικά γραφημάτων και τεχνικές οπισθοδρόμησης χρησιμοποιήθηκαν για την κατασκευή των λύσεων. Η ιδέα είναι ότι η πληθικότητα της μεγαλύτερης ομάδας καθορίζει τον μικρότερο αριθμό των χρονοθυρίδων που απαιτούνται για το πρόβλημα. Τα αποτελέσματα έδειξαν ότι κανένα από τα ευρετικά δεν ξεπέρασε οποιοδήποτε από τα υπόλοιπα σε όλα τα προβλήματα που δοκιμάζονται. Μια άλλη σημαντική συμβολή αυτής της εργασίας είναι η εισαγωγή ενός συνόλου από 13 προβλήματα χρονοδιαγράμματος εξετάσεων, τα οποία έγιναν πρότυπα αναφοράς στο πεδίο. Έχουν μελετηθεί ευρέως και χρησιμοποιήθηκαν από διαφορετικές προσεγγίσεις κατά τη διάρκεια των χρόνων (βλ. Πίνακα 19). Το 2001, ο **Carter** και **Johnson** διερεύνησαν τους υπό-γράφους που είναι αρκετά πυκνοί (σχεδόν κλειστές ομάδες), στις 11 από τις περιπτώσεις στα παραπάνω δεδομένα. Παρατήρησαν ότι στα πραγματικά προβλήματα χρονοπρογραμματισμού εξετάσεων υπάρχουν συνήθως πολλές από τις μεγαλύτερες ομάδες και έδειξαν ότι χρησιμοποιώντας τις σχεδόν ομάδες μπορεί ενδεχομένως να επεκταθεί και να βελτιωθεί η παραπάνω προσέγγιση.

Οι **Burke, Newall** και **Weare** το 1998, μελέτησαν την επίδραση της εισαγωγής ενός τυχαίου στοιχείου στην εφαρμογή των ευρετικών γραφημάτων (Βαθμός κορεσμού, Βαθμός Χρώματος και Largest Degree στον Πίνακα 16), με την ανάπτυξη δύο παραλλαγών των

στρατηγικών επιλογής: (1) επιλογή τουρνουά (tournament selection) που επιλέγει τυχαία ένα από ένα υποσύνολο από τις πρώτες εξετάσεις στη ταξινομημένη λίστα και (2) επιλογή προκατάληψης (bias selection) που λαμβάνει την πρώτη εξέταση από την ταξινομημένη λίστα από ένα υποσύνολο του συνόλου των εξετάσεων. Αυτές οι απλές τεχνικές, όταν δοκιμάστηκαν σε τρία από τα σύνολα δεδομένων του Τορόντο, βελτίωσαν τα καθαρά ευρετικά γραφημάτων με οπισθοδρόμηση όσον αφορά τόσο την ποιότητα όσο και την ποικιλία των λύσεων.

Οι **Burke** και **Newall** το 2004 διερεύνησαν μια δυναμική στρατηγική ταξινόμησης που κατάταξε τις εξετάσεις προσαρμοστικά κατά τη διάρκεια της επίλυσης προβλημάτων σε μια επαναληπτική διαδικασία. Ένας ευρετικός τροποποιητής σχεδιάστηκε για να ενημερώνει την ταξινόμηση των εξετάσεων σύμφωνα με την εμπειρία που αποκτήθηκε σε σχέση με τη δυσκολία της ανάθεσης αυτών στην προηγούμενη διεργασία. Παρατηρήθηκε ότι ένα σταθερό προκαθορισμένο ευρετικό (χρησιμοποιείται ως ένα μέτρο δυσκολίας) σε μία παραδοσιακή διαδοχική στρατηγική (όπως φαίνεται στον Πίνακα 16) δεν αποδίδει πάντα καλά σε όλο το φάσμα των προβλημάτων. Η μέθοδος που περιγράφεται σε αυτό το έγγραφο προσαρμόζει την ευρετική ταξινόμηση κατά τη διάρκεια της επίλυσης προβλημάτων για το συγκεκριμένο πρόβλημα που πρέπει να λυθεί. Εκτενή πειράματα διεξήχθησαν σε 11 από τα συνόλων δεδομένων του Τορόντο, και ένα άλλο αναφοράς (αυτό που λέμε Nottingham στοιχεία, βλ. Υποενότητα 3.2). Αυτή η προσέγγιση δείχθηκε να είναι απλή και αποτελεσματική (συγκρίσιμη ή περιστασιακά καλύτερη από state-of-the-art προσεγγίσεις) και δεν εξαρτάται από την αρχική διάταξη των εξετάσεων.

Η ασαφής λογική είχε χρησιμοποιηθεί από τον Asmuni το 2004 για να ταξινομήσει τις εξετάσεις να προγραμματιστούν με βάση ευρετικά χρωματισμού γραφημάτων για τα σύνολα δεδομένων του Τορόντο. Η ιδέα είναι ότι κατά την ταξινόμηση των εξετάσεων από το πόσο δύσκολο είναι, ασαφείς συναρτήσεις μπορούν να χρησιμοποιηθούν για να δώσουν μια

κατάλληλη αξιολόγηση. Θεωρήθηκε ότι διαφορετικές ασαφείς λειτουργίες πρέπει να χρησιμοποιούνται σε διαφορετικά προβλήματα για την επίτευξη των βέλτιστων αποτελεσμάτων.

Ο **Corr** ανέπτυξε ένα νευρωνικό δίκτυο από το οποίο ένα μέτρο της δυσκολίας ανάθεσης εξετάσεων κατά τη διάρκεια της κατασκευής του χρονοδιαγράμματος μπορεί να ληφθεί αναδρομικά από την εισαγωγή των ενημερωμένων καταστάσεων κατασκευής της λύσης. Ο στόχος είναι να εκχωρηθούν προσαρμοστικά οι πιο δύσκολες εξετάσεις στο πρώιμο στάδιο της κατασκευής της λύσης. Το νευρωνικό δίκτυο εκπαιδεύτηκε με την αποθήκευση των καταστάσεων της κατασκευής του χρονοδιαγράμματος (διανύσματα χαρακτηριστικών γνωρισμάτων) χρησιμοποιώντας τρία ευρετικά γράφημα. Η εργασία απέδειξε την σκοπιμότητα της χρησιμοποίησης μεθόδων με βάση νευρωνικά δίκτυα ως προσαρμοστικές και γενικά εφαρμόσιμες τεχνικές σε προβλήματα χρονοδιαγραμμάτων.

Λόγω των περιορισμών των εποικοδομητικών μεθόδων, όπου οι πρόωρες αναθέσεις δύνανται να οδηγήσουν σε καταστάσεις που δεν είναι διαθέσιμες εφικτές χρονοθυρίδες για τις εξετάσεις που έμειναν αργότερα στο πλαίσιο της διαδικασίας κατασκευής, υπαναχωρήσεις συνήθως χρησιμοποιούνται που αποδεσμεύουν τις πρώτες αντικρουόμενες εξετάσεις για να διαθέσουν τις υπάρχουσες.

Όπως αναφέρθηκε προηγουμένως, οι τεχνικές που «υβριδίζουν» ευρετικά γραφημάτων με άλλες μεθόδους εξακολουθούν να εμφανίζονται στην πιο σύγχρονη ερευνητική βιβλιογραφία χρονοπρογραμματισμού εξετάσεων.

2. Τεχνικές Βασισμένες σε Παραμέτρους

Ο προγραμματισμός με λογική περιορισμών και τις τεχνικές ικανοποίησης περιορισμών έχουν τις ρίζες τους στην έρευνα Τεχνητής Νοημοσύνης. Τέτοιες μέθοδοι έχουν προσελκύσει

την προσοχή των ερευνητών σε χρονοδιαγράμματα που αυτό οφείλεται στην ευκολία και την ευελιξία με τις οποίες μπορούν να χρησιμοποιηθούν για προβλήματα χρονοπρογραμματισμού. Οι εξετάσεις μοντελοποιούνται ως μεταβλητές με πεπερασμένους τομείς. Οι τιμές εντός των τομέων που εκπροσωπούν τις χρονοθυρίδες και τις αίθουσες για τις μεταβλητές, έχουν ανατεθεί διαδοχικά για την κατασκευή λύσεων για τα προβλήματα. Πρώιμες έρευνες επικεντρώθηκαν στην εξεύρεση εφικτών λύσεων (δηλαδή ικανοποιώντας όλους τα αυστηρούς περιορισμούς). Οι Brailsford, Potts και Smith το 1999 εισήγαγαν διάφορες μεθόδους αναζήτησης για προβλήματα ικανοποίησης περιορισμών και απέδειξαν ότι αυτή η τεχνική μπορεί να εφαρμοστεί σε προβλήματα βελτιστοποίησης.

Οι τεχνικές που βασίζονται σε περιορισμούς είναι συνήθως ακριβές υπολογιστικά, για το γεγονός ότι ο αριθμός των πιθανών αναθέσεων αυξάνεται εκθετικά με τον αριθμό των μεταβλητών. Αυτές, από μόνες τους, δεν μπορούν να παρέχουν συνήθως υψηλής ποιότητας λύσεις σε σύγκριση με τις state-of-the-art προσεγγίσεις για πολύπλοκα προβλήματα βελτιστοποίησης. Η υπαναχώρηση χρησιμοποιείται όταν δεν υπάρχουν τιμές που μπορούν να εκχωρηθούν στις μεταβλητές στη συνέχεια της διαδικασίας. Διαφορετικές ευρετικά και τεχνικές έχουν συνήθως ενσωματωθεί με τέτοιες μεθόδους για τη μείωση της πολυπλοκότητας του χρόνου για την επίλυση πρακτικών προβλημάτων. Για παράδειγμα, η στρατηγική σήμανσης υποδεικνύει τη σειρά με την οποία οι μεταβλητές πρέπει να κινηθούν και είναι συνήθως όπου τα ευρετικά έχουν εισαχθεί.

Ο **David** εφάρμοσε τεχνικές ικανοποίησης περιορισμών για να μοντελοποιήσει ένα πρόβλημα χρονοπρογραμματισμού εξετάσεων σε ένα γαλλικό σχολείο, το Ecole des Mines de Nantes. Η πολυπλοκότητα του χρόνου ήταν ζωτικής σημασίας έτσι μερικές λύσεις αποκτήθηκαν αρχικά, με βάση σε ποιες ιδιαίτερες τοπικές στρατηγικές επισκευής χρησιμοποιήθηκαν διαδοχικά να λάβουν ολοκληρωμένες λύσεις και να κάνουν βελτιώσεις. Η προσέγγιση αυτή έτρεξε αρκετές φορές με διαφορετικές αρχικές εργασίες για να μειωθεί η πιθανότητα της έλλειψης καλών

λύσεων. Είχε χρησιμοποιηθεί με επιτυχία στο σχολείο και συνήθως μπορεί να παράγει λύσεις μέσα σε ένα δευτερόλεπτο.

Οι **Reis** και **Oliveira** ανέπτυξαν ένα σύστημα κατάρτισης προγράμματος εξεταστικής βασισμένο σε ECLiPSe, το οποίο είναι ένα σύστημα που βασίζεται σε Prolog που χρησιμεύει ως πλατφόρμα για την ανάπτυξη διάφορων επεκτάσεων στον προγραμματισμό με λογική περιορισμών. Μια σειρά από αυστηρούς και ελαστικούς περιορισμούς στο πρόβλημα κατασκευάστηκαν σε ένα μοντέλο ικανοποίησης περιορισμών, όπου σύνολα μεταβλητών χρησιμοποιήθηκαν και χειρίστηκαν από τις βιβλιοθήκες στο ECLiPSe. Η εφαρμογή του σε τυχαία και πραγματικά μεγάλα δεδομένα του προβλήματος χρονοπρογραμματισμού εξετάσεων στο Πανεπιστήμιο του Φερνάντο Πεσσόα στο Πόρτο απέδειξε την αποτελεσματικότητα του μοντέλου.

Ο **Merlot** χρησιμοποιείσαι προγραμματισμό με περιορισμούς με ένα παρόμοιο τρόπο με εκείνον χρησιμοποιώντας την OPL, μια γλώσσα προγραμματισμού βελτιστοποίησης, για να παράγει αρχικές λύσεις. Στη συνέχεια, μια προσομοιωμένη απόπτωση (Simulated Annealing) και μια μέθοδο αναρρίχησης λόφου (hill climbing) (βλ. Ενότητα 3 παρακάτω) χρησιμοποιήθηκαν για να βελτιώσουν τις λύσεις. Οι μεταβλητές (εξετάσεις) διατάχθηκαν από τα μεγέθη των περιοχών τους (διαθέσιμες χρονοθυρίδες) και έχουν προγραμματιστεί στις πρώτες χρονοθυρίδες μία προς μία. Ο καθαρός προγραμματισμός περιορισμών λαμβάνει το καλύτερο αποτέλεσμα για ένα από τα σύνολα δεδομένων του Τορόντο. Η συνολική υβριδική προσέγγιση δοκιμάστηκε σε προβλήματα στο Πανεπιστήμιο της Μελβούρνης, δύο παραλλαγές σε περιπτώσεις του Τορόντο και των δεδομένων του Νότινγχαμ. Η προσέγγιση αυτή έλαβε τα καλύτερα αποτελέσματα που έχουν αναφερθεί στην βιβλιογραφία σε αρκετές περιπτώσεις του Τορόντο και στα σύνολα δεδομένων του Νότινγχαμ κατά το χρόνο.

Οι **Duong** και **Lam** επίσης εφάρμοσαν προγραμματισμό με περιορισμούς για να δημιουργήσουν αρχικές λύσεις για μία μεθοδολογία προσομοιωμένης ανόπτωσης (Simulated Annealing) για τα προβλήματα χρονοπρογραμματισμού εξετάσεων στο HoChiMinh City University of Technology. Υπαναχωρήσεις και επιπλέον έλεγχοι χρησιμοποιήθηκαν για να μειώσουν τις προσπάθειες αναζήτησης. Η στρατηγική επισήμανσης διέταξε δυναμικά τις μεταβλητές (εξετάσεις) από έναν αριθμό παραγόντων όπως το μέγεθος του τομέα και του αριθμού των σπουδαστών.

Πρόσφατες έρευνες σε τεχνικές με βάση περιορισμούς αντιπροσωπεύουν σύνολα μεθοδολογιών τα οποία υβριδίσθηκαν με άλλες τεχνικές. Η στρατηγική επισήμανσης συνήθως ενσωματώνεται με διαφορετικά ευρετικά προβλημάτων για τη διάταξη των μεταβλητών και είναι κρίσιμη για την επιτυχία της μεθόδου. Η ανάπτυξη κάποιου ισχυρού συστήματος/γλώσσας προγραμματισμού με περιορισμούς (π.χ. Eclipse, CHIP, OPL, Prolog) υποστήριξε σημαντικά την κατασκευή ολοκληρωμένων συστημάτων χρονοπρογραμματισμού εξετάσεων σε εφαρμογές πραγματικού κόσμου. Ωστόσο, μόνο συγκεκριμένα προβλήματα σε διάφορα ιδρύματα έχουν αντιμετωπιστεί με την προσέγγιση αυτή στην βιβλιογραφία. Δεν έχουν γίνει συγκρίσεις μεταξύ τεχνικών με βάση περιορισμούς και άλλων state-of-the-art προσεγγίσεων (εκτός από με μια αυτόματη μέθοδο) στα ίδια προβλήματα, εκτός από του Merlot για τα σύνολα δεδομένων του Τορόντο και του Νότινγχαμ. Αξίζει να σημειωθεί, ωστόσο, ότι αυτή η μέθοδος μπορεί να παράγει τα καλύτερα αποτελέσματα κατά τη βιβλιογραφία σχετικά με ορισμένα προβλήματα αναφοράς.

3. Τεχνικές βασισμένες στην Τοπική Αναζήτηση

Οι τεχνικές με βάση την τοπική αναζήτηση (π.χ. Tabu Search, προσομοιωμένη ανόπτωση και παραλλαγές τους) και οι Εξελικτικοί Αλγόριθμοι θεωρούνται συνήθως ότι ανήκουν στα μετα-ευρετικά. Οι τοπικές μέθοδοι αναζήτησης είναι μια οικογένεια γενικών τεχνικών που

επιλύουν προβλήματα με την αναζήτηση από μια κατεστημένη λύση στη γειτονιά της. Διαφορετικές γειτονικές δομές και κινούμενοι τελεστές εντός του χώρου αναζήτησης διακρίνουν διαφορετικές τοπικές τεχνικές αναζήτησης. Η αναζήτηση καθοδηγείται από μια καθορισμένη λειτουργία στόχου, η οποία χρησιμοποιείται για την αξιολόγηση της ποιότητας των παραγόμενων χρονοδιαγραμμάτων.

Οι τεχνικές αυτές αποτελούν ένα μεγάλο σώμα δουλειάς κατά την τελευταία δεκαετία και έχουν εφαρμοστεί σε μια ποικιλία από προβλήματα χρονοπρογραμματισμού, κυρίως επειδή διαφορετικοί περιορισμοί μπορούν να αντιμετωπιστούν σχετικά εύκολα. Η απόδοση και η αποτελεσματικότητα αυτών των τεχνικών είναι πάρα πολύ εξαρτώμενες από τις παραμέτρους και τις ιδιότητες του χώρου αναζήτησης (π.χ. συνδεδεσιμότητα, ruggedness), έτσι αρκετή γνώση είναι συνήθως εφαρμόσιμη για την αντιμετώπιση συγκεκριμένων προβλημάτων. Ένα μεγάλο ποσό των παραλλαγών και συνδυασμών έχουν ερευνηθεί. Εμείς θα ασχοληθούμε πρώτα να με την Tabu αναζήτηση.

3.1 Tabu Search

Η Tabu Αναζήτηση διερευνά το χώρο αναζήτησης με το να μην επισκέπτεται ξανά μια λίστα των πρόσφατων κινήσεων (διατηρείται σε ταμπού λίστα). Μπορούν, ωστόσο, να επιλεγούν εάν δημιουργούν την καλύτερη λύση που λαμβάνεται μέχρι τώρα με τη χρήση μιας στρατηγικής αναρροφήσεως. Διαφορετικά, η έρευνα κινείται προς άλλες γειτονιές, ακόμη και αν οι προκύπτουσες λύσεις είναι χειρότερες από ό, τι τα κατεστημένων λύσεις, οι οποίες είναι σε θέση να ξεφύγουν από τοπικά βέλτιστα. Οι παράμετροι πρέπει να τελειοποιηθούν στο σχεδιασμό της προσέγγισης και αυτό εξαρτάται σε μεγάλο βαθμό στο πρόβλημα. Τέτοιες παράμετροι περιλαμβάνουν την λίστα ταμπού και τα κριτήρια διακοπής μεταξύ άλλων.

Οι **Di Gaspero και Schearf** (2001) πραγματοποίησαν μια πολύτιμη έρευνα για μια οικογένεια τεχνικών με βάση τις Tabu αναζητήσεις των οποίων οι γειτονιές αφορούν εκείνους που συνέβαλαν στις παραβιάσεις του αυστηρών ή ελαστικών περιορισμών. Εξαντλητικές και μεροληπτικές στρατηγικές επιλογής μελετήθηκαν επίσης. Το μήκος της tabu λίστας είναι δυναμικό και η συνάρτηση κόστους ορίζεται προσαρμοστικά κατά την αναζήτηση. Η προσέγγιση αυτή δοκιμάστηκε σε δύο σύνολα προβλημάτων συγκριτικής μέτρησης επιδόσεων (του Τορόντο και του Νότινγχαμ) και φάνηκε να λειτουργεί παρόμοια με τα ευρετικά γραφήματα, την clique initialisation και Μιμητικούς Αλγόριθμους, αλλά χειρότερα από ό, τι στην πολλαπλών σταδίων μιμητική προσέγγιση. Οι συγγραφείς πειραματικά απέδειξαν ότι η λειτουργία προσαρμογής κόστους και η αποτελεσματική επιλογή των γειτονιών σχετικά με τις παραβιάσεις ήταν βασικά χαρακτηριστικά της προσέγγισης. Το 2002 ο **Di Gaspero** βελτίωσε την προσέγγιση με τη χρήση πολλαπλών γειτονιών που βασίζονται σε μια αναζήτηση token-ring που απασχολεί κυκλικά τον επαναχρωματισμό (αλλάζει μόνο μία εξέταση) και την ανακίνηση (εναλλαγή ομάδων των εξετάσεων), που ακολουθείται από *kickers* (αλλαγή ακολουθίας μονών εξετάσεων) για περαιτέρω βελτίωση των λύσεων που λαμβάνονται. Η τεχνική επέκτεινε την ιδέα της διαφοροποίησης της αναζήτησης από τοπικά βέλτιστα.

Οι **White και Xie** (2001) ανέπτυξαν μία αναζήτηση Tabu τεσσάρων σταδίων που ονομάζεται OTTABU, όπου οι λύσεις βελτιώθηκαν σταδιακά, εξετάζοντας περισσότερους περιορισμούς σε κάθε στάδιο, για το πρόβλημα χρονοπρογραμματισμού εξετάσεων στο Πανεπιστήμιο της Οτάβα. Σε περίπτωση που το 1ο στάδιο αποτυγχάνει, εφικτές λύσεις μπορούν να δημιουργηθούν με την προσθήκη ορισμένων επιπλέον χρονοθυρίδων στο 2ο στάδιο. Στη συνέχεια, οι λύσεις σταδιακά βελτιώνονται με την εξέταση περιορισμών 2ης και 3ης σειράς στο πρόβλημα στο 3ο και 4ο στάδιο, αντίστοιχα. Εκτός από την πρόσφατη βραχυπρόθεσμη μνήμη, μία συχνότητα μακροπρόθεσμης μνήμη χρησιμοποιήθηκε επίσης για την καταγραφή

της συχνότητας των πιο δραστικών κινήσεις στην ιστορία αναζήτησης. Το μέγεθος της μακροπρόθεσμης μνήμης ορίστηκε από την ανάλυση του αριθμού των λιγότερο σημαντικών εξετάσεων για το πρόβλημα. Στην προσέγγιση αυτή επεκτάθηκε, όπου και οι δύο από τις ταμπού λίστες θα μπορούσαν να χαλαρώσουν δυναμικά (αδειάσουν) μετά από ένα ορισμένο χρονικό διάστημα αναζήτησης με καμία βελτίωση. Οι συγγραφείς πειραματικά έδειξαν ότι η απασχόληση μακροπρόθεσμης μνήμης μπορεί να βελτιώσει σημαντικά την Αναζήτηση Tabu για προβλήματα του πραγματικού κόσμου.

Οι **Paquete και Stutzle** (2002) ανέπτυξαν μια μεθοδολογία Tabu Αναζήτησης για χρονοπρογραμματισμό εξετάσεων που διέταξε τις προτεραιότητες που δόθηκαν για τους περιορισμούς. Οι περιορισμοί θεωρήθηκαν με δύο τρόπους: (1) ένας περιορισμός τη φορά από την υψηλότερη προτεραιότητα, όπου δεσμοί έσπασαν με την εξέταση των περιορισμών χαμηλής προτεραιότητας. (2) όλοι οι περιορισμοί τη φορά, ξεκινώντας από τον υψηλότερης προτεραιότητας. Η 2η στρατηγική, έλαβε καλύτερα αποτελέσματα, ενώ η 1η στρατηγική ήταν πιο συνεπής. Το μήκος της ταμπού λίστας ορίστηκε προσαρμοστικά με βάση τον αριθμό των παραβιάσεων των λύσεων. Παρατηρήθηκε ότι το μήκος της tabu λίστας έπρεπε να αυξηθεί με το μέγεθος των προβλημάτων.

3.2 Προσομοιωμένη Ανόπτηση (Simulated Annealing)

Η Προσομοιωμένη Ανόπτηση υποκινείται από την φυσική διαδικασία ανόπτησης [2]. Η ιδέα είναι να ψάξει μια ευρύτερη περιοχή του χώρου αναζήτησης κατά την έναρξη της διαδικασίας με την αποδοχή χειρότερων κινήσεων με μεγαλύτερη πιθανότητα, η οποία βαθμιαία μειώνεται καθώς η αναζήτηση συνεχίζεται. Μία *θερμοκρασία* χρησιμοποιείται μέσα σε ένα *ψυκτικό πρόγραμμα* για να ελέγχει την πιθανότητα της αποδοχής χειρότερων κινήσεων στην αναζήτηση. Πολλές παράμετροι πρέπει να συντονιστούν στην προσομοιωμένη ανόπτηση περιλαμβανομένων των αρχικών και τελικών θερμοκρασιών, και το ψυκτικό παράγοντα στο

ψυκτικό χρονοδιάγραμμα. Αυτές οι παράμετροι επηρεάζουν την απόδοση και την επιτυχία της προσέγγισης αυτής.

Οι **Thompson και Dowsland** (1998) εκτέλεσαν πολύτιμο έργο να αναπτύξει μια προσέγγιση δύο σταδίων, όπου εφικτές λύσεις από το 1ο στάδιο βελτιώθηκαν στο 2ο στάδιο με προσομοιωμένη ανόπτηση λαμβάνοντας υπόψη σχετικά ελαστικούς περιορισμούς. Όπως διαφορετικοί στόχοι εξετάστηκαν σε διάφορα στάδια με τη σειρά, οι λύσεις από το πρώιμο στάδιο μπορεί να είναι κακές και έτσι μία τεχνική οπισθοδρόμηση προτάθηκε. Ο **Dowsland** παρατήρησε επίσης ότι ο τρόπος ορισμού της γειτονιάς, η σημασία και πόσο δύσκολο να επιτευχθούν οι στόχοι επηρεάζεται σε μεγάλο βαθμό ο τρόπος με τον οποίο αντιμετωπίζεται σε κάθε στάδιο. Με βάση το έργο τους, οι συγγραφείς διερεύνησαν περαιτέρω την γειτονιά της αλυσίδας Kempe, όπου οι αλυσίδες των εξετάσεων και όχι μεμονωμένες εξετάσεις μεταφέρθηκαν. Αυτό έδωσε περισσότερη ευελιξία για να καταστεί δυνατή η κίνηση των μεγάλων δύσκολων εξετάσεων εντός του χρονοδιαγράμματος. Κατέληξαν στο συμπέρασμα ότι οι πιο σημαντικοί παράγοντες για την προσομοιωμένη ανόπτηση ήταν το *ψυκτικό πρόγραμμα* και ο τρόπος με τον οποίο οι γειτονιές ορίστηκαν και υπέστησαν δειγματοληψία. Οι συγγραφείς ανέφεραν ότι το σύστημα χρονοπρογραμματισμού εξετάσεων που αναπτύχθηκε έχει χρησιμοποιηθεί στο πανεπιστήμιο του Swansea με επιτυχία από το 1993.

Ο **Bullheimer** (1998) συζήτησε για το πώς ένα μοντέλο για Τετραγωνικά Προβλήματα Αναθέσεων προσαρμόστηκε για να διαμορφώσει μια μικρής κλίμακας πρακτικό πρόβλημα χρονοπρογραμματισμού εξετάσεων στο Πανεπιστήμιο του Magdeburg. Τα μοντέλα έδωσαν τη δυνατότητα στους διαχειριστές του πανεπιστημίου να ελέγχουν πόσο πρέπει να απέχουν οι αντικρουόμενες εξετάσεις. Η Προσομοιωμένη Ανόπτηση χρησιμοποιήθηκε όπου δύο σύνολα γειτονικών δομών (μετακίνηση των χρονοθυρίδων των εξετάσεων και μετακίνηση μονών εξετάσεων) μελετήθηκαν. Ωστόσο, οι λεπτομέρειες των παραμέτρων του αλγορίθμου δεν δόθηκαν.

Ο **Merlot** (2003) χρησιμοποίησε μια προσέγγιση προσομοιωμένης ανόπτησης αρχικοποιημένη με τεχνικές προγραμματισμού με περιορισμούς και ακολουθείται από αναρρίχηση λόφου για να βελτιώσει περαιτέρω τη λύση. Μια τροποποιημένη γειτονιά αλυσίδας Kempe χρησιμοποιήθηκε. Τα καλύτερα αποτελέσματα μέχρι τώρα για αρκετές από τις περιπτώσεις του Τορόντο επιτεύχθηκαν από αυτή την υβριδική προσέγγιση. Πράγματι, η μέθοδος εξακολουθεί να έχει μερικά από τα καλύτερα γνωστά αποτελέσματα. Οι συντάκτες πρότειναν ότι οι μέθοδοι που συνδυάζουν κατασκευή λύσης με τοπική αναζήτηση θα κυριαρχήσουν το μέλλον του χρονοπρογραμματισμού εξετάσεων.

Οι **Duong και Lam** (2004) που χρησιμοποίησαν προσομοιωμένη ανόπτηση στις αρχικές λύσεις που παράγονται από τον προγραμματισμό περιορισμών του προβλήματος χρονοπρογραμματισμού εξετάσεων στο HCMC University of Technology. Μια γειτονιά αλυσίδας Kempe ήταν εμπλεκόμενη στην προσομοιωμένη ανόπτηση, όπου το πρόγραμμα ψύξης ορίστηκε πειραματικά χρησιμοποιώντας μηχανισμούς και αλγορίθμους. Οι συγγραφείς παρατήρησαν ότι όταν δίνεται περιορισμένος χρόνος, είναι σημαντικό να συντονιστούν τα στοιχεία στην προσομοιωμένη ανόπτηση με τα ειδικά προβλήματα που πρέπει να επιλυθούν.

Ο **Burke** (2004) μελέτησε μια παραλλαγή της προσομοιωμένης ανόπτησης, που ονομάζεται αλγόριθμος Great Deluge. Η αναζήτηση δέχεται χειρότερες κινήσεις για όσο διάστημα η μείωση στην ποιότητα είναι κάτω από ένα ορισμένο *επίπεδο*, το οποίο αρχικά οριστεί ως η ποιότητα της αρχικής λύσης και σταδιακά χαμηλώνει κατά έναν παράγοντα φθοράς. Ο παράγοντας φθοράς και μια εκτίμηση της επιθυμητής ποιότητας αποτελούν τις παραμέτρους σε αυτήν την προσέγγιση. Οι συγγραφείς σημειώνουν ότι οι παράμετροι αυτές μπορούν να προ-ορίζονται από τους χρήστες, οι οποίοι συνήθως δεν είναι ειδικοί στην προσομοιωμένη ανόπτηση. Οι αρχικές λύσεις, εν τούτοις, πρέπει να είναι εφικτές για τον υπολογισμό του συντελεστή φθοράς έτσι ένας Δείκτης Κορεσμού εκτελέστηκε πολλές φορές, από τον οποίο οι καλύτερες λύσεις χρησιμοποιήθηκαν ως σημεία εκκίνησης. Η προσέγγιση αυτή ήταν

ανώτερη σε μια προσομοιωμένη ανόπτηση που αναπτύχθηκε από τους συγγραφείς. Δείχθηκε να είναι αποτελεσματική και να δημιουργούνται μερικά από τα καλύτερα αποτελέσματα για τα σύνολα δεδομένων του Τορόντο και του Νότινχαμ, σε σύγκριση με άλλες προσεγγίσεις. Περιεκτικά πειράματα διεξήχθησαν επίσης για να αναλύσουν το trade-off μεταξύ του χρόνου και της ποιότητας λύση για τα προβλήματα διαφορετικού μεγέθους.

3.3 Άλλες τεχνικές με βάση την Τοπική αναζήτηση

Πρόσφατα, μαζί με τη μελέτη των διαφορετικών τρόπων διαφυγής από τοπικά βέλτιστα στις τεχνικές με βάση την τοπική αναζήτηση, ορισμένοι ερευνητές στράφηκαν προς την διερεύνηση της επίδρασης του σχεδιασμού διαφορετικών γειτονιών και έχουν αποκτήσει κάποια επιτυχία στα προβλήματα χρονοπρογραμματισμού. Αυτό έδειξε ότι όχι μόνο ο τρόπος αναζήτησης, αλλά επίσης, η δομή της γειτονιάς είχε σημαντική επίπτωση στην αναζήτηση αλγορίθμων. Για παράδειγμα, οι γειτονικές δομές αλυσίδας Kempe όπως αναφέρθηκε πάνω ερευνήθηκαν από έναν αριθμό ερευνητών στο χρονοπρογραμματισμό εξετάσεων. Η ιδέα είναι ότι οι αλυσίδες των αντικρουόμενων εξετάσεων ανταλλάσσονται μεταξύ των χρονοθυρίδων. Σε σύγκριση με τις πρότυπες κινήσεις σε μεμονωμένες εξετάσεις, αυτό έφερε μεγαλύτερη ευελιξία στην πλοήγηση των χώρων αναζήτησης για διαφορετικά προβλήματα.

Ο **Abdullah** το 2006, ανέπτυξε μια μεγάλη έρευνα γειτονιάς με βάση τη μεθοδολογία της κατασκευής του γραφήματος βελτίωσης που αναπτύχθηκε αρχικά από τους **Ahuja και Orlin** για διαφορετικά προβλήματα βελτιστοποίησης. Για να παράγουν μεγάλες γειτονιές, αντί να εξετάζουν μόνο τα παραδοσιακά ζεύγη τελεστών με βάση την ανταλλαγή, μία γειτονική δομή βασισμένη σε δέντρο σχεδιάστηκε για να πραγματοποιήσει κυκλική ανταλλαγών μεταξύ όλων των χρονοθυρίδων. Η προσέγγιση έδωσε τα καλύτερα αποτελέσματα σε μια σειρά από προβλήματα συνόλου δεδομένων του Τορόντο κατά τη στιγμή της δημοσίευσης. Ωστόσο, μία

μεγάλη ποσότητα υπολογιστικού χρόνου χρειαζόταν. Η προσέγγιση περισσότερο αναπτύχθηκε, όπου οι κινήσεις βελτίωσης διατηρήθηκαν σε *tabu* λίστα.

Μια άλλη τεχνική που αφορά διαφορετικές γειτονιές είναι η Μεταβλητή Αναζήτηση Γειτονιάς. Η προσέγγιση αυτή διαφέρει συστηματικά έναν αριθμό γειτονικών δομών. Ο στόχος είναι να ξεφύγουν από τοπικά βέλτιστα με την αλλαγή από το χώρο αναζήτησης που ορίζεται από μια γειτονιά στην άλλη. Ωστόσο, δεν έχει γίνει πολύ δουλειά σε χρονοπρογραμματισμό εξετάσεων που χρησιμοποιούν αυτή την προσέγγιση. Ο **Burke** διερεύνησε παραλλαγές της μεταβλητής αναζήτησης γειτονιάς και έλαβε τα καλύτερα αποτελέσματα στην βιβλιογραφία σε μερικά από τα προβλήματα συνόλου δεδομένων στο Τορόντο. Τα αποτελέσματα βελτιώθηκαν περαιτέρω με τη χρήση ενός προτύπου Γενετικών Αλγόριθμων για να επιδεχθούν έξυπνα υποσύνολα των γειτονιών. Η τελευταία προσέγγιση έχει ισχυρούς δεσμούς με την εργασία σε υπέρ-ευρετικά και έδειξε ελπιδοφόρες κατευθύνσεις για την ανάπτυξη γενικών προσεγγίσεων στις γειτονιές και όχι απευθείας σε λύσεις. Στα υπερ-ευρετικά, η Μεταβλητή Αναζήτηση Γειτονιάς χρησιμοποιήθηκε επίσης όπου ευρετικά γραφημάτων ερευνήθηκαν παρά οι γειτονιές.

Εκτός από το σχεδιασμό διαφορετικών γειτονικών δομών στην τοπική αναζήτηση με βάση τις τεχνικές, μερικοί ερευνητές έχουν εξετάσει επίσης τον τρόπο με τον οποίο επαναληπτικές τεχνικές μπορούν να βοηθήσουν στην επίλυση σύνθετων προβλημάτων. Στην Επαναληπτική Τοπική αναζήτηση, η αναζήτηση επανεκκινείται μετά από την πλήρωση ορισμένων κριτηρίων. Το κίνητρο είναι να εξερευνηθούν περισσότερες περιοχές του χώρου αναζήτησης σε σύντομο χρονικό διάστημα. Εφαρμόστηκε για πρώτη φορά σε ένα πρόβλημα χρωματισμού γραφήματος το 2002.

Οι **Caramia, DellOlmo και Italiano** (2001) ανέπτυξαν μία καλά ρυθμισμένη μέθοδο τοπικής αναζήτησης όπου ένας άπληστος *προγραμματιστής* αναθέτει εξετάσεις στο ελάχιστο δυνατό

αριθμό χρονοθυρίδων και ένας *μειωτής ποινής* βελτιώνει το χρονοδιάγραμμα χωρίς να αυξάνει τον αριθμό των χρονοθυρίδων. Όταν καμία βελτίωση δεν μπορεί να γίνει ο αριθμός των χρονοθυρίδων αυξάνεται σταδιακά από έναν *εναλλάκτη ποινής*. Η διαδικασία επαναλαμβάνεται χρησιμοποιώντας μια τεχνική μετάθεση για τον επαναπροσδιορισμό των προτεραιοτήτων των εξετάσεων. Ένα σωστά συντονισμένο σχήμα *ελέγχου* χρησιμοποιήθηκε επίσης για να απελευθερώσει τη μνήμη της αναζήτησης. Η προσέγγιση αυτή εξακολουθεί να κατέχει τα καλύτερα αποτελέσματα που αναφέρονται στην βιβλιογραφία σε αρκετές περιπτώσεις για τα σύνολα δεδομένων του Τορόντο.

Οι **Casey και Thompson** (2003) διερεύνησαν μία Άπληστα Τυχαιοποιημένη Προσαρμοστική Διαδικασίες Αναζήτησης (Greedy Randomised Adaptive Search Procedures - GRASP) προσέγγιση, η οποία είναι μια σχετικά νέα τεχνική. Στην GRASP, ένας αλγόριθμος τοπικής αναζήτησης ξεκινά επαναληπτικά μετά την εύρεση του τοπικού βέλτιστου με βάση τις αρχικές λύσεις που παράγονται από μία άπληστη προσέγγιση. Η αρχική λύση σε κάθε επανάληψη παρήχθη με έναν τροποποιημένο Βαθμό Κορεσμού, όπου μία εξέταση από τις πρώτες n (πειραματικά ορισμένο ως 2-6) εξετάσεις που ταξινομήθηκαν ανατέθηκε στο χρονοδιάγραμμα. Η οπισθοδρόμηση χρησιμοποιήθηκε σε συνδυασμό με μία *tabu* λίστα για να απαγορευθούν οι αόριστοι κύκλοι. Μία περιορισμένη μορφή της προσομοιωμένης ανόπτησης με την υψηλή θερμοκρασία εκκίνησης και γρήγορη ψύξη χρησιμοποιήθηκε στο στάδιο βελτίωσης. Η κινήσεις της αλυσίδας *Kempe* απασχολούνταν στις εξετάσεις όπου αυτό συνέβαλε στο κόστος. Οι προσεγγίσεις που εφαρμόζονται για τα σύνολα δεδομένων του Τορόντο ανέφεραν ανταγωνιστικά αποτελέσματα για μερικές από τις περιπτώσεις εκείνη τη χρονική στιγμή.

3.4 Περίληψη για την διερεύνηση των τεχνικών με βάση την Τοπική Αναζήτηση.

Κατά την τελευταία δεκαετία, οι τεχνικές με βάση την τοπική αναζήτηση έχουν μελετηθεί σε πολύ μεγάλο βαθμό και έχουν αποκτήσει ένα σημαντικό επίπεδο επιτυχίας στον χρονοπρογραμματισμό. Όλη η εργασία που συζητήθηκε παραπάνω είτε έχει δοκιμαστεί σε συγκριτική αξιολόγηση δεδομένων είτε έχει υλοποιηθεί σε πραγματικές εφαρμογές. Διαφορετικοί τρόποι αποδοχής των κινήσεων (δηλαδή στρατηγικές κίνησης, στρατηγική αποδοχής και στρατηγικές επιλογής) μελετήθηκαν έτσι ώστε να ξεφύγουν από τα τοπικά βέλτιστα ορίζοντας μια ποικιλία προσεγγίσεων μετά-ευρετικών για χρονοπρογραμματισμό εξετάσεων. Ωστόσο, ένα σημαντικό μειονέκτημα αυτών των προσεγγίσεων είναι η προσπάθεια που απαιτείται για να συντονιστούν οι παράμετροι για συγκεκριμένα προβλήματα για να πάρουμε υψηλής ποιότητας λύσεις.

4. Αλγόριθμοι Βασισμένοι στον Πληθυσμό.

4.1 Εξελικτικοί Αλγόριθμοι.(Evolutionary Algorithms)

Οι Εξελικτικοί Αλγόριθμοι αποτελούν μια οικογένεια αλγορίθμων με βάση τον πληθυσμό συμπεριλαμβανομένων Γενετικών Αλγορίθμων, Μιμητικών Αλγορίθμων και Αλγορίθμων Μυρμηγκιών. Οι Γενετικοί Αλγόριθμοι έχουν μελετηθεί πιο πολύ από την άποψη της έρευνας του χρονοπρογραμματισμού εξετάσεων. Ειδικότερα, υβριδισμοί των γενετικών αλγορίθμων με τις τοπικές μεθόδους αναζήτησης (μερικές φορές ονομάζονται μιμητικοί αλγόριθμοι) έχουν οδηγήσει σε κάποια επιτυχία στο πεδίο.

Οι Γενετικοί Αλγόριθμοι προσομοιώνουν την εξελικτική πορεία στη φύση με την χειραγώγηση και εξέλιξη πληθυσμών των λύσεων στο χώρο αναζήτησης. Οι λύσεις κωδικοποιούνται ως *χρωμοσώματα* και εξελίσσονται από μια διαδικασία αναπαραγωγής χρησιμοποιώντας φορείς διασταύρωσης και μετάλλαξης, με στόχο την όλο και καλύτερες λύσεις μέσω μιας σειράς των γενεών. Ένα σύνολο παραμέτρων και φορέων σε Γενετικούς Αλγόριθμους πρέπει να καθοριστεί και ρυθμιστεί σωστά, κάνοντας την προσέγγιση

(συνήθως) πιο περίπλοκη από ό, τι εκείνη των μεθόδων με βάση την τοπική αναζήτηση. Η στρατηγική αναζήτησης στους Γενετικούς Αλγόριθμους είναι θεμελιωδώς διαφορετική από τις προσεγγίσεις που βασίζονται στην τοπική αναζήτηση που συζητήθηκε παραπάνω με την έννοια ότι οι διάφορες λύσεις αντιμετωπίζονται ταυτόχρονα (ένας πληθυσμός λύσεων) παρά μόνο μία λύση που βελτιώνεται μέσα από μια σειρά από επαναλήψεις.

Οι **Corne, Ross και Fang** το 1994 παρείχαν μια σύντομη έρευνα σχετικά με τη χρήση Γενετικών Αλγόριθμων σε γενικό εκπαιδευτικό χρονοπρογραμματισμό και απεύθυναν ορισμένα θέματα και μελλοντικές προοπτικές. Μία σημαντική συμβολή του έργου αφορά τη χρήση της άμεση εκπροσώπησης στους γενετικούς αλγόριθμους, η οποία φαίνεται να είναι ανίκανη για την αντιμετώπιση ορισμένων δομών προβλημάτων σε ορισμένα ειδικά προβλήματα χρωματισμού γραφημάτων. Το 2003 οι **Ross, Hart και Corne** ενημέρωσαν την ανωτέρω επισκόπηση το 1994 στις παραστάσεις και τους αλγόριθμους που χρησιμοποιούνται σε Εξελικτικούς Αλγόριθμους για τα διάφορα είδη των προβλημάτων χρονοπρογραμματισμού. Κατέληξαν στο συμπέρασμα ότι οι ερευνητές αντιμετώπιζαν τα δικά τους προβλήματα σε διάφορα ιδρύματα και δεν υπήρχε ακόμα η ανάγκη για περισσότερες συγκρίσεις των προσεγγίσεων σε ένα ευρύ φάσμα των προβλημάτων.

Οι **Ross, Corne και Terashima-Marin** (1996) έδειξαν ότι η μετάβαση περιοχών υπάρχει σε επιλύσιμα προβλήματα χρονοπρογραμματισμού από τον πειραματισμό με την δημιουργούνται ειδικών προβλημάτων χρωματισμού γραφημάτων, διαφορετικής συνδεσιμότητας και ομοιογένειας. Οι συγγραφείς ανέφεραν ότι η μελέτη μπορεί να βοηθήσει στην κατανόηση του πώς διαφορετικοί αλγόριθμοι αποδίδουν σε πολύπλοκα προβλήματα χρονοπρογραμματισμού. Το 1998, οι **Ross, Hart και Corne** έδωσαν περαιτέρω ενδείξεις για την αδυναμία της χρήσης της άμεσης κωδικοποίησης σε Γενετικούς Αλγόριθμους. Παρατήρησαν την αποτυχία ενός αριθμού (εξελικτικών και μη εξελικτικών) προσεγγίσεων για την επίλυση ειδικών κατηγοριών προβλημάτων χρωματισμού γραφημάτων. Ισχυρίστηκαν

ότι οι γενετικοί αλγόριθμοι θα πρέπει να αναζητούν αλγόριθμους και όχι πραγματικά λύσεις. Πράγματι τα υπερ-ευρετικά (όπου ένα σύνολο ευρετικών χαμηλού επιπέδου αναζητείται από έναν αλγόριθμο υψηλού επιπέδου) κάνουν ακριβώς αυτό.

Οι **Terashima-Marin, Ross και Valenzuela-Rendon** το 1999 σχεδίασαν έναν clique-based τελεστή διασταύρωσης σε προβλήματα χρονοπρογραμματισμού που μεταφέρθηκε σε προβλήματα χρωματισμού γραφημάτων. Διαφορετικές στρατηγικές ανασυνδυασμού δοκιμάστηκαν στις διεργασίες αναπαραγωγής για να διατηρήσουν τις κλίκες σε γονείς στους απογόνους. Επεσήμαναν το ίδιο πρόβλημα με την άμεση αναπαράσταση στους Γενετικούς Αλγορίθμους όπως συζητήθηκε παραπάνω με. Πρότειναν εναλλακτικές λύσεις για μελλοντική εργασία. Οι ίδιοι συγγραφείς μελέτησαν επίσης τη λειτουργία ποινής σε τόσο τυχαία όσο και πραγματικά προβλήματα χρονοπρογραμματισμού που απασχολούν την Θεωρία Σκληρότητας (Hardness Theory), η οποία προβλέπει, που είναι οι δυσκολότερες περιπτώσεις εντός των προβλημάτων χρονοπρογραμματισμού. Ωστόσο, παρατηρήθηκε ότι προσθέτοντας το μέτρο αυτό, δεν είναι χρήσιμο για την καθοδήγηση των Γενετικών Αλγορίθμων σε υποσχόμενες περιοχές του χώρου αναζήτησης. Με βάση την ανωτέρω εργασία, διερευνήθηκε οι μη άμεση κωδικοποίηση σε Γενετικούς Αλγόριθμους, όπου στρατηγικές κατασκευής λύσεων και ευρετικά κωδικοποιήθηκαν (π.χ. διαμορφώσεις μεθόδων ικανοποίησης περιορισμών, την ταξινόμηση των κόμβων που έχουν ανατεθεί και ευρετικά που ασχολούνται με περιορισμούς, κλπ). Ενθαρρυντικά αποτελέσματα που λήφθηκαν με αυτή την προσέγγιση στα σύνολα δεδομένων του Τορόντο έδειξαν το ενδεχόμενο της μη άμεσης αναπαράστασης σε Γενετικούς Αλγόριθμους.

Ο **Erben** (2001) ανέπτυξε ένα Γενετικό Αλγόριθμο ομαδοποίησης όπου κατάλληλες λειτουργίες κωδικοποίησης και καταλληλότητας μελετήθηκαν. Τα γονίδια αυτά ομαδοποιούνται για κάθε χρώμα σε προβλήματα χρωματισμού γραφημάτων (όπου μοντελοποιούν που τα προβλήματα χρονοπρογραμματισμού εξεταστικής με μόνο αυστηρούς

περιορισμούς). Ειδικά σχεδιασμένοι τελεστές διασταύρωσης και μετάλλαξης για την κωδικοποίηση της ομάδας χρησιμοποιήθηκαν. Αν και τα αποτελέσματα από την άποψη της ποιότητας λύσης δεν ανταγωνίζονται με τα καλύτερα, η προσέγγιση απαιτεί λιγότερο υπολογιστικό χρόνο από μερικές από τις μεθόδους στη βιβλιογραφία.

Ο **Sheibani**, το 2002, κατασκεύασε ένα ειδικό μαθηματικό μοντέλο και ανέπτυξε έναν πρότυπο Γενετικό Αλγόριθμο για την επίλυση των προβλημάτων χρονοπρογραμματισμού εξεταστικής σε κέντρα κατάρτισης με στόχο τη μεγιστοποίηση των διαστημάτων μεταξύ των εξετάσεων. Ένα δίκτυο δραστηριότητας χρησιμοποιήθηκε για να εκτιμηθεί η εγγύτητα μεταξύ των εξετάσεων, η οποία χρησιμοποιήθηκε στην συνάρτηση καταλληλότητας στο Γενετικό Αλγόριθμο.

Οι **Wong, Cote και Gely** (2002) συζήτησαν ορισμένα θέματα που αφορούν την εφαρμογή του δικού τους γενετικού αλγορίθμου για την επίλυση ενός προβλήματος χρονοπρογραμματισμού εξετάσεων στο Cole de Technologie Suprieure, το οποίο διαμορφώθηκε ως Πρόβλημα Ικανοποίησης Περιορισμών. Η επιλογή των τουρνουά χρησιμοποιήθηκε για την επιλογή των γονέων και στρατηγικών επισκευής ενσωματωμένων με μετάλλαξη για να παράγουν καλύτερες υποψηφιότητες. Το 2005, οι **Cote, Wong, και Sabourin** διερεύνησαν έναν διπλού στόχου εξελικτικό αλγόριθμο με τους στόχους της ελαχιστοποίησης του μήκους του χρονοδιαγράμματος και της απόστασης μεταξύ αντικρουόμενων εξετάσεων. Δύο τελεστές τοπικής αναζήτησης (αντί των τελεστών ανασυνδυασμού) χρησιμοποιήθηκαν για να ασχοληθούν με τους αυστηρούς και ελαστικούς περιορισμούς. Ήταν μία κλασική Tabu Αναζήτηση και μια απλουστευμένη Variable Neighbourhood Descent με αλυσίδα Kempe και γειτονιές μόνης κίνησης. Μια διαδικασία κατάταξης βασίζεται στη δύναμη Pareto για τη διεξαγωγή της αξιολόγησης των ατόμων στον πληθυσμό. Η προσέγγιση έλαβε ανταγωνιστικά αποτελέσματα σε μια σειρά από προβλήματα αναφοράς σε αντίθεση με μερικές από τις μεθόδους στη βιβλιογραφία. Το έγγραφο παρείχε

επίσης μία επανεξέταση σε όλες τις state-of-the-art προσεγγίσεις για τα σύνολα δεδομένων του Τορόντο κατά τη χρονική στιγμή.

4.2 Μιμητικοί Αλγόριθμοι (Memetic Algorithms)

Οι **Μιμητικοί αλγόριθμοι** είναι μια επέκταση των Γενετικών Αλγορίθμων των οποίων η βασική ιδέα είναι ότι τα άτομα σε έναν πληθυσμό βελτιώνονται κατά τη διάρκεια της ζωής τους μέσα μια γενιά. Αυτό συχνά εφαρμόζεται με τη χρήση των τοπικών μεθόδων αναζήτησης (σε μορφή της αναρρίχησης λόφου ή στρατηγικών επισκευής, κλπ) σε μεμονωμένα μέλη ενός πληθυσμού μεταξύ των γενεών. Οι **Burke και Landa Silva** συζήτησαν ορισμένα θέματα που αφορούν το σχεδιασμό των Μιμητικών Αλγορίθμων για τον προγραμματισμό των προβλημάτων χρονοπρογραμματισμού. Πρόσφατες ερευνητικές ιδέες και μελλοντικές κατευθύνσεις παρουσιάστηκαν για αυτό το θέμα.

Η δυνατότητα να εξερευνήσουν με τη χρησιμοποίηση μίας μεθόδου με βάση τον πληθυσμό και να αξιοποιήσει για την τοπική αναζήτηση επιτρέπει τέτοιες μεθόδους να αντιμετωπίσουν αποτελεσματικά τα μεγάλα πολύπλοκα προβλήματα. Ωστόσο, υπάρχει συνήθως ένα τίμημα όσον αφορά τον απαιτούμενο υπολογιστικό χρόνο. Επίσης, η σωστή ισορροπία μεταξύ της *εξερεύνησης* και *εκμετάλλευσης* πρέπει να δημιουργηθεί. Υπάρχει ένας αριθμός των εις βάθος μελετών για Μιμητικούς Αλγόριθμους που αφορούν δομές του χώρου αναζήτησης και διαφορετικούς τρόπους υβριδισμών σε μία περιοχή των συνδυαστικών προβλημάτων βελτιστοποίησης.

Οι **Burke, Newall και Weare** (1996) ανέπτυξαν έναν Μιμητικό αλγόριθμο ο οποίος απασχολεί ελαστικούς και αυστηρούς τελεστές μετάλλαξης για να ξανά-αναθέσουν μόνες εξετάσεις και σύνολα των εξετάσεων, αντίστοιχα, με σκοπό να ξεφύγουν από τοπικά βέλτιστα. Καμία από αυτές τις μεταλλάξεις από μόνες τους δεν παρέχει ουσιαστική βελτίωση σχετικά με την ποιότητας λύσης. Η αναρρίχηση λόφου χρησιμοποιήθηκε για να βελτιώσει τα

άτομα και έδειξε ότι αυτό βελτίωσε την ποιότητα των χρονοδιαγραμμάτων, αν και περισσότερος υπολογιστικός χρόνος απαιτήθηκε. Μια άλλη συμβολή αυτής της εργασίας ήταν η εισαγωγή περισσότερων προβλημάτων αναφοράς χρονοπρογραμματισμού εξετάσεων (όπως ονομάζονται δεδομένα Nottingham). Αυτά έχουν χρησιμοποιηθεί ευρέως από έναν αριθμό ερευνητών. Οι ίδιοι συγγραφείς επίσης ερεύνησαν τις επιπτώσεις της πολυμορφίας σε αρχικούς πληθυσμούς σε Μιμητικούς Αλγόριθμους (1998). Για να δημιουργηθεί μία καλή ποικιλομορφία στον αρχικό πληθυσμό, η τυχειότητα εισήχθη με τη χρήση διαφορετικών στρατηγικών επιλογής σε ευρετικά γραφημάτων, και τρία μέτρα πολυμορφίας, αναπτύχθηκαν επίσης, για τη μελέτη της ανταλλαγής ανάμεσα στην ποιότητα και την ποικιλομορφία. Έχει αποδειχθεί ότι η μελέτη της πολυμορφίας στην αρχικοποίηση προσφέρει μεγάλα οφέλη για τους Μιμητικούς Αλγόριθμους. Το 1999, οι **Burke και Newall** μελέτησαν την αποσύνθεση των προβλημάτων χρονοπρογραμματισμού εξετάσεων με αναθέτοντας κάθε φορά ένα υποσύνολο n εξετάσεων οι οποίες είναι οι πιο δύσκολες που μετρήθηκαν από ευρετικά γραφημάτων (δηλ. Βαθμός Χρώματος, Μεγαλύτερος Βαθμός, Βαθμός). Οπισθοδρόμηση και τεχνικές πρόβλεψης χρησιμοποιούνται για να αποφευχθεί το πρόβλημα των πρόωρων αναθέσεων που δημιουργούν ασυμβατότητα αργότερα. Οι εξετάσεις που ανατέθηκαν σε προηγούμενα στάδια είναι σταθερές και το υπό-πρόβλημα στο παρόν στάδιο λύνεται με τη χρήση του Μιμητικού Αλγόριθμου. Ο αλγόριθμος μείωσε δραματικά τον χρόνο που απαιτείται για την πολυπλοκότητα και παρήγαγε υψηλής ποιότητας λύσεις όταν ασχολήθηκε με μεγαλύτερα προβλήματα χρονοπρογραμματισμού. Η τεχνική αποσύνθεσης ήταν πραγματικά ανεξάρτητη από τον μιμητικό αλγόριθμο χρονοπρογραμματισμού ο οποίος χρησιμοποιήθηκε σε καθένα από τα αποσυντεθημένα υποσύνολα.

4.3 Αλγόριθμοι Μυρμηγκιών (Ant Algorithms)

Οι Αλγόριθμοι Μυρμηγκιών ανήκουν στην οικογένεια των τεχνικών με βάση τον πληθυσμό. Αυτοί προσομοιώνουν τον τρόπο που τα μυρμήγκια αναζητούν την συντομότερη διαδρομή

για τα τρόφιμα με το να αφήνουν φερομόνη στο δρόμο. Τα συντομότερα μονοπάτια δημιουργούν ισχυρότερα επίπεδα φερομόνης πάνω από ένα χρονικό διάστημα. Στον αλγόριθμο, κάθε μυρμήγκι χρησιμοποιείται για την κατασκευή μίας λύσης και οι πληροφορίες κατά τη διάρκεια της αναζήτησης διατηρούνται ως φερομόνη, η οποία χρησιμοποιείται για να βοηθήσει στη δημιουργία λύσεων στο επόμενο στάδιο. Στον χρονοπρογραμματισμό εξεταστικής, οι Αλγόριθμοι Μυρμηγκιών αντιπροσωπεύουν σχετικά πρόσφατα διερευνημένες τεχνικές. Σε αυτό το πλαίσιο, δεν έχουν ιδιαίτερα μελετηθεί ευρέως. Σχετικές εργασίες, ωστόσο, υπάρχουν στα προβλήματα χρωματισμού γραφημάτων, όπου η συχνότητα των χρωμάτων που εκχωρήθηκαν για τις κορυφές στην κατασκευή της λύσης, χρησιμοποιήθηκαν σαν φερομόνη.

Ο **Naji Azimi**, το 2004 υλοποίησε ένα Σύστημα Αποικία Μυρμηγκιών και το σύγκρινε με την Προσομοιωμένη Ανόπτηση, την Αναζήτηση Tabu και έναν Γενετικό Αλγόριθμο κάτω ένα ενιαίο πλαίσιο για την επίλυση συστηματικά σχεδιασμένων προβλημάτων χρονοπρογραμματισμού εξετάσεων. Οι αρχικές λύσεις για το Σύστημα Αποικίας Μυρμηγκιών δημιουργήθηκαν ευρετικά και βελτιώθηκαν από την τοπική αναζήτηση αργότερα. Τα αποτελέσματα που αναλύθηκαν κατά τη διάρκεια της λειτουργίας ανέφεραν ότι η προσέγγιση αποικίας μυρμηγκιών απέδωσε τα καλύτερα (αν και όχι για όλα τα προβλήματα) και η Tabu Αναζήτηση είχε το υψηλότερο επίπεδο βελτίωσης κατά την τυχαία δημιουργημένη αρχική λύση. Παρατηρήθηκε ότι οι υβριδικές προσεγγίσεις λειτουργούν καλύτερα από κάθε ενιαίο αλγόριθμο, και το διαδοχικό Σύστημα Αποικίας Μυρμηγκιών που ακολουθείται από Tabu Αναζήτηση απέκτησε το καλύτερο αποτελέσματα. Ωστόσο, μόνο τυχαία δεδομένα ελέγχθηκαν.

Οι **Dowland** και **Thompson** το 2005 ανέπτυξαν Αλγόριθμους Μυρμηγκιών με βάση το μοντέλο χρωματισμού γραφημάτων, για την επίλυση των προβλημάτων χρονοπρογραμματισμού εξετάσεων χωρίς ελαστικούς περιορισμούς (δηλαδή να βρεθεί ο

μικρότερος αριθμός χρονοθυρίδων). Εκτενής πειράματα διεξήχθησαν για τη μέτρηση της απόδοσης του αλγορίθμου με διαφορετικές διαμορφώσεις. Αυτές περιλαμβάνουν τις μεθόδους αρχικοποίησης (δηλαδή αναδρομικό Μεγαλύτερο βαθμό και Βαθμό Κορεσμού), υπολογισμοί μονοπατιού, τρεις παραλλαγές των λειτουργιών καταλληλότητας και διαφορετικές ρυθμίσεις παραμέτρων. Τα αποτελέσματα που ελήφθησαν ήταν ανταγωνιστικά με τα άλλα για το ίδιο σύνολο δεδομένων. Παρατηρήθηκε επίσης ότι οι μέθοδοι αρχικοποίησης είχαν σημαντική επίδραση στην ποιότητα της λύσης. Επεκτάσεις του αλγορίθμου να ενσωματώσει άλλους περιορισμούς επίσης συζητήθηκαν.

4.4 Περίληψη της συζήτησης των τεχνικών με βάση τον πληθυσμό

Οι εξελικτικές μέθοδοι (ιδιαίτερα εξελικτικά υβρίδια) ήταν πολύ αποτελεσματικές στην παροχή υψηλής ποιότητας λύσεων για το πρόβλημα του χρονοπρογραμματισμού εξετάσεων. Πρόσφατες έρευνες για τους Εξελικτικούς Αλγόριθμους συζήτησαν έχει τα ζητήματα της κωδικοποίησης για την αντιμετώπιση με των δομών προβλημάτων που η άμεση κωδικοποίηση δεν είναι σε θέση να αντιμετωπίσει. Αυτό άνοιξε μια νέα ερευνητική κατεύθυνση στους Εξελικτικούς Αλγορίθμους και έχει οδηγήσει σε μερικές από τις αρχικές εργασίες στα Hyper-heuristics (βλ. Ενότητα 4.6). Πολυκριτηριακές τεχνικές αποτελούν επίσης μία σημαντική κατεύθυνση της έρευνας στον τομέα των Εξελικτικών Αλγόριθμων για προβλήματα χρονοπρογραμματισμού εξετάσεων.

Οι έχουν Αλγόριθμοι Μυρμηγκιών έχουν εφαρμοστεί σε χρονοπρογραμματισμό εξετάσεων με κάποιες αρχικές παρατηρήσεις. Όπως οι σχετικά νέες τεχνικές, που αντιπροσωπεύουν κάποιες δυνατότητες και θα πρέπει να προσελκύσουν περισσότερο την προσοχή στον τομέα του χρονοπρογραμματισμού εξετάσεων.

5. Τεχνικές Πολλαπλών Κριτηρίων

Στην πλειοψηφία των αλγορίθμων / προσεγγίσεις για χρονοπρογραμματισμό, τα σταθμισμένα κόστη των παραβιάσεων διαφόρων περιορισμών αθροίζονται και χρησιμοποιούνται για να δείξουν την ποιότητα των λύσεων. Ωστόσο, σε πραγματικές συνθήκες οι περιορισμοί συχνά λαμβάνονται υπόψη από διαφορετικές οπτικές γωνίες από διάφορα μέρη που εμπλέκονται στην διαδικασία του χρονοπρογραμματισμού. Το απλό άθροισμα του κόστους για διαφορετικούς περιορισμούς δεν μπορεί να πάντα να φροντίζει για την κατάσταση σε τέτοιες περιπτώσεις. Τεχνικές πολλαπλών κριτηρίων έχουν μελετηθεί πρόσφατα στον χρονοπρογραμματισμό, με στόχο την διαχείριση διαφορετικών περιορισμών εύκολα θεωρώντας ένα διάνυσμα των περιορισμών, αντί ενός μοναδικού σταθμισμένου αθροίσματος. Στις τεχνικές πολλαπλών κριτηρίων, κάθε κριτήριο μπορεί να θεωρηθεί ότι αντιστοιχεί σε ένα περιορισμό, ο οποίος έχει ένα ορισμένο επίπεδο σημασίας και εξετάζεται χωριστά. Σε κάποιες μεθόδους, έχουν χρησιμοποιηθεί πολλαπλά στάδια για την αντιμετώπιση διαφορετικών στόχων. Οι **Landa, Burke** και **Petrovic** παρέχουν μια ανασκόπηση ενός μεγάλου αριθμού εφαρμογών χρονοπρογραμματισμού που χρησιμοποιούν τεχνικές πολλαπλών κριτηρίων.

Οι **Burke, Bykov** και **Petrovic** (2001) ανέπτυξαν μία προσέγγιση πολλαπλών κριτηρίων δύο σταδίων που είχε να κάνει με εννέα κριτήρια στα προβλήματα χρονοπρογραμματισμού εξετάσεων (π.χ. χωρητικότητα του δωματίου, εγγύτητα των εξετάσεων, ο χρόνος και η σειρά των εξετάσεων, κλπ). Στο 1ο στάδιο, ο Δείκτης Κορεσμού χρησιμοποιήθηκε για να παράγει ένα σύνολο από εφικτές λύσεις, όπου κάθε κριτήριο εξετάζεται μεμονωμένα. Το 2ο στάδιο, στη συνέχεια βελτίωσε ευρετικά αυτές οι λύσεις ταυτόχρονα. Μια μέθοδος πολλαπλών κριτηρίων που ονομάζεται Συμβιβαστικός Προγραμματισμός χρησιμοποιήθηκε, όπου η ποιότητα των λύσεων αξιολογήθηκε από την απόσταση μεταξύ τους σε ένα ιδανικό σημείο

που αντιπροσωπεύει βέλτιστες λύσεις σχετικά με όλα τα κριτήρια. Αυτή η τεχνική μελετήθηκε περαιτέρω από τους **Petrovic** και **Bykov**. Αυτοί βάσισαν την προσέγγιση πολλαπλών κριτηρίων τους στον αλγόριθμο Great Deluge. Ένα σημείο αναφοράς που παρέχεται από τους χρήστες χρησιμοποιήθηκε για να συντάξει μία τροχιά στο χώρο κριτηρίων. Τα βάρη κριτηρίων μπορούν να αλλάξουν δυναμικά για να κατευθύνουν την αναζήτηση, ξεκινώντας από τυχαία σημεία, προς το σημείο αναφοράς. Στοχεύει στο ιδανικό σημείο στο χώρο κριτηρίων. Ωστόσο, τα αρχικά βάρη που χρειάζονται να καθοριστούν ήταν εξαρτημένα από τα προβλήματα. Επίσης, η αναζήτηση δεν είναι εγγυημένη να συγκλίνει. Δημοσιευμένα αποτελέσματα χρησιμοποιήθηκαν ως σημεία αναφοράς για την προσέγγιση και τα τελικά αποτελέσματα ήταν καλύτερα για μερικά από τα προβλήματα αναφοράς που δοκιμάστηκαν. Αυτές οι προσεγγίσεις παρείχαν την ευελιξία για τους χρονοπρογραμματιστές να λάβουν επιθυμητές λύσεις με τη διαχείριση των βαρών των διαφόρων περιορισμών.

6. Υπέρ - Ευρετικά

Η εξάρτηση από την ρύθμιση παραμέτρων ή ο τρόπος της ενσωματωμένης γνώσης τομέα (δηλαδή η υψηλή κωδικοποίηση των αυστηρών και ελαστικών περιορισμών) αντικρούεται από την μετά-ευρετική ανάπτυξη για τον χρονοπρογραμματισμό εξετάσεων. Μερικές από τις πιο αποτελεσματικές τεχνικές αναφοράς για τα δεδομένα στη βιβλιογραφία είναι τα μετά-ευρετικά. Ωστόσο, οι περισσότερες από αυτές τις μεθόδους αντιπροσωπεύουν μια εξατομικευμένη προσέγγιση για ένα συγκεκριμένο πρόβλημα (σε αυτή την περίπτωση, χρονοπρογραμματισμό εξετάσεων). Οι μέθοδοι αυτές συνήθως δεν λειτουργούν καλά ή δεν είναι ικανές να ασχοληθούν με άλλα προβλήματα. Πράγματι, μπορεί να είναι η περίπτωση ότι οι μέθοδοι αυτές δεν λειτουργούν συνεπώς σε όλες τις περιπτώσεις προβλημάτων χρονοπρογραμματισμού εξετάσεων. Συχνά, ρύθμιση παραμέτρων μπορεί να διαδραματίσει

σημαντικό ρόλο. Η προσπάθεια του συντονισμού παραμέτρων για να ταιριάζει με νέα προβλήματα μπορεί να θεωρηθεί σαν να είναι τόσο δύσκολο όσο η ανάπτυξη νέων προσεγγίσεων. Αυτό το γνωστό θέμα έχει οδηγήσει μια σειρά από ερευνητές να αναπτύξουν νέες τεχνολογίες που στοχεύουν στη λειτουργία σε υψηλότερο επίπεδο γενικότητας.

Τα Υπέρ-ευρετικά υποκινούνται από αυτές τις παρατηρήσεις και προσελκύουν ένα αυξημένο επίπεδο της προσοχής της έρευνας. Ο όρος μπορεί να θεωρηθεί ότι αντιπροσωπεύει *ευρετικά που επιλέγουν ευρετικά*, δηλαδή ένας χώρος αναζήτησης των ευρετικών είναι το επίκεντρο της προσοχής μάλλον παρά ένας χώρος αναζήτησης των λύσεων (όπως είναι η περίπτωση με τις περισσότερες υλοποιήσεις των μετά-ευρετικών). Ο στόχος είναι να αναπτυχθούν πιο γενικές προσεγγίσεις αντί να νικηθούν οι τέλεια ρυθμισμένες και συγκεκριμένες στο πρόβλημα προσεγγίσεις που συχνά απαιτούν πολλή προσπάθεια για τη ρύθμιση των παραμέτρων και είναι συνήθως κατάλληλες μόνο για συγκεκριμένα προβλήματα.

Όπως προαναφέρθηκε στην ενότητα 4.1, οι **Ross, Hart** και **Corne** πρότειναν ότι ένας Γενετικός Αλγόριθμος θα μπορούσε να χρησιμοποιηθεί επιτυχώς στην αναζήτηση για καλούς αλγόριθμους παρά συγκεκριμένες λύσεις. Οι **Terashima-Marin, Ross** και **Valenzuela-Rendon** διερεύνησαν με χρήση εξελικτικών αλγορίθμων για την αναζήτηση στρατηγικών κατασκευής λύσεων.

Ο **Ahmadi** το 2003 ανέπτυξε μια Μεταβλητή Αναζήτηση Γειτονιάς για να βρει καλούς συνδυασμούς παραμετροποιημένων ευρετικών για διαφορετικά προβλήματα χρονοπρογραμματισμού εξετάσεων. Μεταθέσεις των ευρετικών χαμηλού επιπέδου (δηλαδή επιλογή επτά εξετάσεων, επιλογή δύο χρονοθυρίδων και επιλογή τριών αιθουσών ευρετικών) και των συναφών τους παραμέτρους (βάρη) χρησιμοποιήθηκαν για την κατασκευή λύσεων.

Οι **Ross, Marin-Blazquez** και **Hart** το 2004 ανέπτυξαν έναν Γενετικό Αλγόριθμο γενικής σταθερής κατάστασης για να αναζητήσει στο πλαίσιο ενός απλουστευμένου χώρου

αναζήτησης των περιγραφών του προβλήματος, οι οποίες αντιστοιχούν στα συμβάντα και χρονοθυρίδες που συλλέγουν ευρετικά για την κατασκευή λύσεων. Δεδομένου ότι η αναζήτηση του Γενετικού Αλγόριθμου ήταν σε ευρετικά παρά πραγματικές λύσεις, τρεις διαφορετικές λειτουργίες καταλληλότητας δοκιμάστηκαν. Οι περιγραφές της κατάστασης του προβλήματος (που αντιστοιχεί σε ευρετικά) ήταν πειραματική μελέτη σε σχέση με αυτές τις λειτουργίες καταλληλότητας. Ενθαρρυντικά αποτελέσματα τόσο για την πορεία αναφοράς όσο και των προβλημάτων χρονοπρογραμματισμού εξετάσεων έδειξαν πολύτιμες κατευθύνσεις δυναμικής έρευνας αυτής της προσέγγισης για μια σειρά από προβλήματα.

Οι **Kendall** και **Hussin** το 2004 ερεύνησαν μια Tabu Αναζήτηση βασισμένη σε υπέρ-ευρετικά, όπου και οι δύο κινούμενες στρατηγικές και εποικοδομητικά ευρετικά γραφημάτων απασχολούνταν ως χαμηλού επιπέδου ευρετικά. Οι αλγόριθμοι εξετάστηκαν για προβλήματα χρονοπρογραμματισμού εξετάσεων στο MARA University of Technology, και δείχθηκε ότι παράγαγε καλύτερα αποτελέσματα σε σύγκριση με τις λύσεις που παράχθηκαν χειροκίνητα.

Ο **Burke** διερεύνησε την απασχόληση της Case-Based Reasoning, μια τεχνική που βασίζεται στη γνώση, ως έναν ευρετικό επιλογέα για την επίλυση των προβλημάτων χρονοπρογραμματισμού μαθημάτων και εξετάσεων. Το 2006, τεχνικές ανακάλυψης γνώσης χρησιμοποιήθηκαν για να ανακαλύψουν τις πιο σχετικές λειτουργίες που χρησιμοποιούνται για την αξιολόγηση της ομοιότητας μεταξύ καταστάσεων επίλυσης προβλημάτων. Ο στόχος ήταν να επιλέξουν τα καλύτερα ευρετικά από την πιο παρόμοια προηγούμενη κατάσταση επίλυσης προβλημάτων για να κατασκευάσει καλές λύσεις για το πρόβλημα. Το θέμα του καθορισμού της ομοιότητας μεταξύ των προβλημάτων χρονοπρογραμματισμού εξετάσεων έχει επίσης μελετηθεί όσον αφορά την επιλογή της καλύτερης μεθόδου επίλυσης προβλημάτων. Το (2005), διαφορετικοί τρόποι υβριδισμού τα χαμηλού επιπέδου ευρετικά γραφημάτων (με και χωρίς CBR) συγκρίθηκαν για την επίλυση των συνόλων δεδομένων του Τορόντο. Αποδείχθηκε ότι χρησιμοποιώντας τεχνικές που βασίζονται στη γνώση και όχι

τυχαία / συστηματικά υβριδισμού ευρετικά σε ένα υπέρ-ευρετικό πλαίσιο παρουσίασε καλά αποτελέσματα. Οι **Yang** και **Petrovic** χρησιμοποίησαν τον Case- Based Reasoning για να επιλέξουν ευρετικά γραφημάτων για την κατασκευή αρχικών λύσεων για τον αλγόριθμο Great Deluge και έλαβαν τα καλύτερα αποτελέσματα που αναφέρθηκαν στην βιβλιογραφία για αρκετές περιπτώσεις του Τορόντο κατά τη χρονική στιγμή. Τα χαρακτηριστικά γραφήματα χρησιμοποιήθηκαν για τη μοντελοποίηση των περιορισμών στα προβλήματα, έτσι ώστε τα προηγούμενα προβλήματα με παρόμοιους περιορισμούς ανακτήθηκαν για την επίλυση των προβλημάτων στο χέρι με την επαναχρησιμοποίηση των πιο κατάλληλων ευρετικών γραφημάτων.

Ο **Burke** (2006) διερεύνησε με τη χρήση της Tabu Search για να αναζητήσετε μεταθέσεις των ευρετικών γραφημάτων για την κατασκευή λύσεων για τα προβλήματα χρονοπρογραμματισμού. Ένα διαφορετικό αριθμό χαμηλού επιπέδου ευρετικών γραφημάτων μελετήθηκαν με βάση αυτό το υπέρ-ευρετικό για να εκχωρήσει προσαρμοστικά τις πιο δύσκολες εξετάσεις σε διαφορετικά στάδια της κατασκευής της λύσης. Παρατηρήθηκε ότι όσο μεγαλύτερος είναι ο αριθμός των ευφών χαμηλού επιπέδου ευρετικά, τόσο καλύτερη η απόδοση μπορεί να είναι. Ωστόσο, το μέγεθος της χώρας αναζήτησης θα αυξηθεί σημαντικά, με αποτέλεσμα ο υπολογιστικός χρόνος μπορεί να είναι ένα ζήτημα. Τα αποτελέσματα τόσο για τα προβλήματα χρονοπρογραμματισμού εξετάσεων και μαθημάτων ήταν ανταγωνιστικά με τις καλύτερες state-of-the-art προσεγγίσεις που έχουν αναφερθεί στην βιβλιογραφία και απέδειξαν την απλότητα και την αποτελεσματικότητα αυτής της γενικής προσέγγισης. Οι **Qu** και **Burke** διερεύνησαν περαιτέρω την επίδραση της χρησιμοποίησης διαφορετικών υψηλού επιπέδου αλγόριθμους αναζήτησης (δηλαδή Απότομης Καθόδου, Tabu Search, Επαναλαμβανόμενης Τοπικής Αναζήτησης και Μεταβλητής Αναζήτησης Γειτονίας) στο ενοποιημένο γράφημα με βάση ένα υπέρ-ευρετικό πλαίσιο. Πειραματικά αποτελέσματα έδειξαν ότι η μέθοδος αναζήτησης με διαφορετικά ευρετικά υψηλού επιπέδου εντός του

χώρου αναζήτησης των ευρετικών γραφική παράσταση δεν ήταν αποφασιστικής σημασίας. Τα χαρακτηριστικά των δομών γειτονιάς και των χώρων αναζήτησης αναλύθηκαν. Αποδείχθηκε ότι η εξερεύνηση πάνω από το μεγάλο χώρο της λύσης επέτρεψε στην προσέγγιση να έχει καλά αποτελέσματα τόσο για τα προβλήματα χρονοπρογραμματισμού εξετάσεων και μαθημάτων.

Διάφορες στρατηγικές και μεθοδολογίες έχουν χρησιμοποιηθεί ως οι υψηλού επιπέδου μέθοδοι επιλογής σε ένα υπέρ-ευρετικό πλαίσιο για να επιλέξει τα κατάλληλα χαμηλού επιπέδου ευρετικά. Αυτά τα χαμηλού επιπέδου ευρετικά μπορεί να είναι είτε ευρετικά κατασκευής ή ευρετικά βελτίωσης. Τέτοιες μέθοδοι είναι τα θεμέλια των μεθοδολογιών που αυτόματα σχεδιάζουν και προσαρμόζουν ευρετικά χρονοπρογραμματισμού. Αυτό έχει οδηγήσει σε κάποια εργασία στην ανάλυση του χώρου αναζήτησης των ευρετικών (αντί για λύσεις) με το στόχο της ουσιαστικής κατανόησης των διαδικασιών αναζήτησης που υποστηρίζουν αυτή τη νέα προοπτική με την έρευνα χρονοπρογραμματισμού.

Κεφάλαιο 5

Πειράματα και Συγκρίσεις

1. Κατάρτιση Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών (Curriculum Based Course Timetabling)

Περιπτώσεις Προβλήματος και Πειραματικό Πρωτόκολλο

Για να αξιολογηθεί η αποτελεσματικότητα των προτεινόμενων (Adaptive Tabu Search) ATS αλγόριθμων, έχουμε διεξάγει πειράματα σε δύο διαφορετικά σύνολα δεδομένων. Το πρώτο σετ (4 περιπτώσεις με το όνομα test1 ~ test4) είχε χρησιμοποιηθεί στο παρελθόν στη βιβλιογραφία για την παλιά έκδοση του προβλήματος χρονοπρογραμματισμού μαθημάτων.

Το δεύτερο σετ (7 περιπτώσεις με το όνομα comp01 ~ comp07) είναι από τον Δεύτερο Διεθνή Διαγωνισμό Χρονοπρογραμματισμού. Όλες αυτές οι 11 περιπτώσεις μπορούν να μεταφορτωθούν από την ιστοσελίδα <http://tabu.diegm.uniud.it/ctt/index.php>. Ο αλγόριθμος μας έχει προγραμματιστεί σε C και τρέχει σε έναν υπολογιστή με Windows XP με επεξεργαστή 3.44GHz και 2G RAM. Για να αποκτηθούν τα υπολογιστικά αποτελέσματα, ο ATS αλγόριθμος λειτουργεί 100 φορές σε κάθε περίπτωση με διαφορετικές τυχαίες εισόδους. Η κατάσταση παύσης είναι ακριβώς το χρονικό όριο που απαιτείται από τους κανόνες του διαγωνισμού ITC-2007. Στον υπολογιστή μας, αυτό αντιστοιχεί σε 390 δευτερόλεπτα.

Συγκριτικά Αποτελέσματα και Συζήτηση

Ο Πίνακας 17 δείχνει τα αποτελέσματα του αλγορίθμου ATS στις 11 περιπτώσεις των δύο συνόλων στοιχείων και για τις δύο διαμορφώσεις καθώς και τα προηγούμενα γνωστά καλύτερα αποτελέσματα που είναι διαθέσιμα στη βιβλιογραφία. Για τον αλγόριθμο ATS, έχουμε αναφέρει τις ακόλουθες πληροφορίες: το καλύτερο σκορ f_{\min} , ο μέσος όρος βαθμολογίας f_{ave} και η τυπική απόκλιση σ πάνω από 100 ανεξάρτητες επαναλήψεις. Για τους αλγορίθμους αναφοράς, μόνο τα καλύτερα αποτελέσματα f_{\min} είναι διαθέσιμα. Θα πρέπει να αναφερθεί ότι οι όροι παύσης των τριών αλγορίθμων αναφοράς είναι επίσης ο όρος του χρονικού ορίου της ITC-2007. Οι αλγόριθμοι των Di Gaspero, και L., Schaerf χρησιμοποιούν μια δυναμική Tabu αναζήτηση που επιτρέπει ανέφικτες αναθέσεις κατά τη διάρκεια της επίλυσης του προβλήματος και η γειτονιά είναι αυτή που απλά μεταφέρει μία διάλεξη σε μια διαφορετική χρονική περίοδο και / ή μία διαφορετική αίθουσα. Παρατηρεί κανείς ότι αυτή η γειτονιά είναι ακριβώς το υποσύνολο της βασικής γειτονιά N_1 . Ο αλγόριθμος του Muller αποτελείται από μια εποικοδομητική φάση, έναν αλγόριθμο Αναρρίχησης λόφου και την τεχνική Great Deluge και χρησιμοποιεί έξι εξειδικευμένες κινήσεις (γειτονίες).

Από τον πίνακα 17, παρατηρεί κανείς ότι ο ATS αλγόριθμος εξουσιάζει τους αλγόριθμους των Di Gaspero, και L., Schaerf για 8 από 11 περιπτώσεις (με έντονους χαρακτήρες). Για τις 3 υπόλοιπες περιπτώσεις των οποίων το βέλτιστο είναι γνωστό, αυτό επιτυγχάνεται από τον αλγόριθμο ATS εντός μερικών δευτερόλεπτων. Επιπλέον, οι τυπικές αποκλίσεις του ATS για όλες τις δοκιμασμένες περιπτώσεις είναι μικρές, και δείχνει την ευρωστία του.

Αν συγκρίνουμε τον ATS και τον νικητή αλγόριθμο του ITC-2007 για τις 7 περιπτώσεις του διαγωνισμού, παρατηρεί κανείς ότι τα αποτελέσματα των δύο αλγορίθμων είναι αρκετά συγκρίσιμα. Για αυτές τις 7 περιπτώσεις, ο ATS φτάνει καλύτερα (χειρότερα αντίστοιχα) αποτελέσματα από ότι τα αποτελέσματα του αλγορίθμου του Muller για 2 (Αντίστοιχα 3) περιπτώσεις, με ίσια αποτελέσματα για τις υπόλοιπες 2 περιπτώσεις. Σημειώστε ότι ο ATS κατατάσσεται στη δεύτερη θέση για το 3 κομμάτι του ITC-2007.

Ας αναφερθεί ότι μελετήθηκε επίσης η συμπεριφορά του προτεινόμενου αλγορίθμου ATS σχετικά με την επίδραση του ποινής-καθοδηγούμενου μηχανισμού διαταραχής και τους διάφορους μηχανισμούς προσαρμογής. Επιπλέον, μία λεπτομερής μελέτη διεξήχθη για να αναλύσει διάφορες γειτονιές (συμπεριλαμβανομένων εκείνων χρησιμοποιούνται στο παρόν έγγραφο) και τους διαφορετικούς τρόπους του συνδυασμού τους. Αυτή η ανάλυση έδειξε ότι το νέο προτεινόμενο διπλής αλυσίδας Kempe γειτονιάς N_4 και του token-ring συνδυασμού της με την απλή γειτονιά N_1 συμβάλει σημαντικά στην αποτελεσματικότητα του αλγορίθμου ATS.

Πίνακας 17. Υπολογιστικά αποτελέσματα και σύγκριση με τις 11 περιπτώσεις

Instance	Formulation I				Formulation II				
	ATS Heuristic			Best in [7, 6]	ATS Heuristic			Best in [6]	Best in [14]
	f_{min}	f_{ave}	σ	f_{min}	f_{min}	f_{ave}	σ	f_{min}	
test1	212	212	0	213	224	229.5	1.8	234	—
test2	8	8	0	8	16	17.1	1.0	17	—
test3	35	35.3	0.3	36	73	82.9	4.1	86	—
test4	28	32.8	2.1	43	76	89.4	5.8	132	—
comp01	4	4	0	4	5	5	0	5	5
comp02	22	24.5	2.3	35	34	60.6	7.5	75	43
comp03	41	44.8	3.5	52	70	86.6	6.3	93	72
comp04	19	21.8	2.8	21	38	47.9	4.0	45	35
comp05	224	229.4	5.1	244	298	328.5	11.7	326	298
comp06	25	27.6	3.1	27	47	69.9	7.4	62	41
comp07	4	6.3	2.4	13	19	28.2	5.6	38	14

Συμπεράσματα

Έχει παραχθεί μια μαθηματική διατύπωση του προβλήματος Κατάρτισης Ωρολογίου Προγράμματος βάση του Προγράμματος Σπουδών και παρουσίασε ένα εξαιρετικά αποτελεσματικό υβριδικό αλγόριθμο Προσαρμοζόμενης Tabu Αναζήτησης για την επίλυση αυτού του δύσκολου προβλήματος. Η αποτελεσματικότητα του ATS αλγόριθμου προέρχεται από έναν αριθμό από τα πρωτότυπα χαρακτηριστικά. Πρώτον, έχει εισαχθεί η δομή γειτονιάς διπλής αλυσίδας Kempe για το πρόβλημα χρονοπρογραμματισμού. Δεύτερον, για τη διαδικασία TS, έχει επινοηθεί μια συνδυασμένη στρατηγική εκμετάλλευσης της γειτονιάς αλυσίδας Kempe και του βασικής γειτονιάς ανταλλαγής. Τρίτον, έχει προταθεί ένας μηχανισμός για τον προσαρμοστικό συνδυασμό του TS και της διατάραξης. Τα υπολογιστικά αποτελέσματα στις 11 περιπτώσεις των δύο σχηματισμών δείχνουν ότι ο υβριδικός ATS κυριαρχεί τους αλγόριθμους αναφοράς των Di Gaspero, και L., Schaerf και ανταγωνίζεται πολύ καλά με τον νικητή του ITC-2007 του Muller. Ας πούμε ότι τα περισσότερα από τα συστατικά του προτείνονται στο παρόν έγγραφο παραμένουν γενικά και άμεσα θα εφαρμόζονται ή θα προσαρμόζονται σε άλλα συνδυαστικά προβλήματα.

2. Κατάρτιση Ωρολογίου Προγράμματος βάση των δηλώσεων των φοιτητών (Post Enrollment based Course Timetabling)

Σε αυτή την ενότητα, θα ερευνηθεί πειραματικά η απόδοση από την προτεινόμενη υβριδική προσέγγιση για το πρόβλημα κατάρτισης ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών σε σύγκριση με πολλούς άλλους αλγόριθμους. Όλοι οι αλγόριθμοι κωδικοποιήθηκαν σε GNU C++ με την έκδοση 4.1 και έτρεξαν σε έναν υπολογιστή με ισχύ 3,20 GHz. Χρησιμοποιούμε 24 περιπτώσεις αναφοράς στο πρόβλημα κατάρτισης ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών για την δοκιμή των αλγόριθμων, που προτάθηκαν για το ITC-2007. Ο Πίνακας 18 παρουσιάζει τα χαρακτηριστικά αυτών των περιπτώσεων.

Δύο σειρές πειραμάτων διεξήχθησαν σε αυτή τη μελέτη. Η πρώτη ομάδα πειραμάτων είναι αφιερωμένη στην ανάλυση της ευαισθησίας των βασικών παραμέτρων για την εκτέλεση των HGATS για το πρόβλημα κατάρτισης ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών. Η δεύτερη ομάδα πειραμάτων συγκρίνουν την απόδοση του HGATS με δύο σχετικών αλγόριθμων για τη δοκιμή των περιπτώσεων του προβλήματος κατάρτισης ωρολογίου προγράμματος βάση των δηλώσεων των φοιτητών. Στο τέλος, εμείς συγκρίνουμε πειραματικά μας αποτελέσματα με την τρέχουσα state-of-the-art μέθοδο από τη βιβλιογραφία σχετικά με τις δοκιμασμένες περιπτώσεις.

Ανάλυση ευαισθησίας των βασικών παραμέτρων του HGATS

Η απόδοση της προτεινόμενης υβριδικής προσέγγισης εξαρτάται σχετικά με τις παραμέτρους και τελεστές που χρησιμοποιούνται, ειδικά σε Γενετικούς Αλγόριθμους Καθοδηγούμενης Αναζήτησης (GSGA). Μέσα από προηγούμενη εργασία (των Jat και Yang 2009), βρέθηκε ότι τα α , β , γ , και τ είναι βασικές παράμετροι που μπορούν να επηρεάσουν σε μεγάλο βαθμό την απόδοση του GSGA για το Πρόβλημα Χρονοπρογραμματισμού Μαθημάτων, όπου α είναι το ποσοστό των καλύτερων ατόμων που επιλέγονται από τον τρέχων πληθυσμό για τη δημιουργία της δομής δεδομένων MEM, β είναι το ποσοστό του συνολικού αριθμού των

συμβάντων που χρησιμοποιούνται για τη δημιουργία ενός παιδιού μέσω της δομής δεδομένων MEM, γ είναι η πιθανότητα που δείχνει εάν ένα παιδί έχει δημιουργηθεί μέσω της MEM ή με διασταύρωση, και το τ αποφασίζει τη συχνότητα της ενημέρωσης της MEM (δηλαδή, η MEM ενημερώνεται κάθε τ γενιές). Ως εκ τούτου, θα δοκιμάσουμε τον αλγόριθμο HGATS με διαφορετικές ρυθμίσεις αυτών των παραμέτρων. Ο Πίνακας 19 δείχνει διαφορετικές παραμέτρους και τις ρυθμίσεις τους που δοκιμάστηκαν στα πειράματα. Κάποιες άλλες παράμετροι για τον HGATS ορίστηκαν ως εξής: το μέγεθος του πληθυσμού N είχε οριστεί σε 50 και η πιθανότητα μετάλλαξης P_m ορίστηκε στο 0,5.

Πίνακας 18: Περιπτώσεις του προβλήματος ITC-2007

ITC-2007 Instance	1	2	3	4	5	6	7	8	9	10	11	12
Number of events	400	400	200	200	400	400	200	200	400	400	200	200
Number of rooms	10	10	20	20	20	20	20	20	10	10	10	10
Number of features	10	10	10	10	20	20	20	20	20	20	10	10
Number of students	500	500	1000	1000	300	300	500	500	500	500	1000	1000
Max students per event	33	32	98	82	19	20	43	39	34	32	88	81
Max events per student	25	24	15	15	23	24	15	15	24	23	15	15
Mean features per room	3	4	3	3	2	3	5	4	3	3	3	4
Mean features per event	1	2	2	2	1	2	3	3	1	2	1	23
ITC-2007 Instance	13	14	15	16	17	18	19	20	21	22	23	24
Number of events	400	400	200	200	100	200	300	400	500	600	400	400
Number of rooms	20	20	10	10	10	10	10	10	20	20	20	20
Number of features	10	10	20	20	10	10	10	10	20	20	30	30
Number of students	300	300	500	500	500	500	1000	1000	300	500	1000	1000
Max students per event	20	20	41	40	195	65	55	40	16	22	69	41
Max events per student	24	24	15	15	23	23	14	15	23	25	24	15
Mean features per room	2	3	2	5	4	4	3	3	3	3	5	5
Mean features per event	1	1	3	3	2	2	1	1	1	2	3	3

Πίνακας 19: Ρυθμίσεις παραμέτρων στον HGATS

Parameter	Settings			
α	0.2	0.4	0.6	0.8
β	0.1	0.3	0.5	0.7
γ	0.2	0.4	0.6	0.8
τ	20	40	60	80

Για να μάθουμε ποιες ρυθμίσεις παραμέτρων έχουν μια μεγάλη επίδραση στην απόδοση του HGATS, τρέξαμε τον HGATS 50 φορές για όλους τους συνδυασμούς των παραμέτρων στον Πίνακα 19. Εδώ, θα αναφέρουν κάποια τυπικά αποτελέσματα στην Εικόνα 14, όπου η δυναμική απόδοση του HGATS του σχετικά με τη μέση τιμή της αντικειμενικής τιμής σε σχέση με τον αριθμό των αξιολογήσεων για πάνω από 50 δοκιμές με μία παράμετρο να

αλλάζει ενώ οι άλλες παράμετροι παραμένουν σταθερές σε διαφορετικές περιπτώσεις του PECTP. Η Εικόνα 14a δείχνει το αποτέλεσμα της αλλαγής του α στην περίπτωση 2007-16 του προβλήματος με $\beta = 0,3$, $\gamma = 0,8$, και $\tau = 20$. Η Εικόνα 14b δείχνει το αποτέλεσμα της αλλαγής του β στην περίπτωση 2007-17 με $\alpha = 0,2$, $\gamma = 0,8$, και $\tau = 20$. Η Εικόνα 14c δείχνει το αποτέλεσμα της αλλαγής του γ στην περίπτωση 2007-11 με $\alpha = 0,2$, $\beta = 0,3$, και $\tau = 20$. Η Εικόνα 14d δείχνει το αποτέλεσμα της αλλαγής του τ στην περίπτωση 2007-3 με $\alpha = 0,2$, $\beta = 0,3$ και $\gamma = 0,8$.

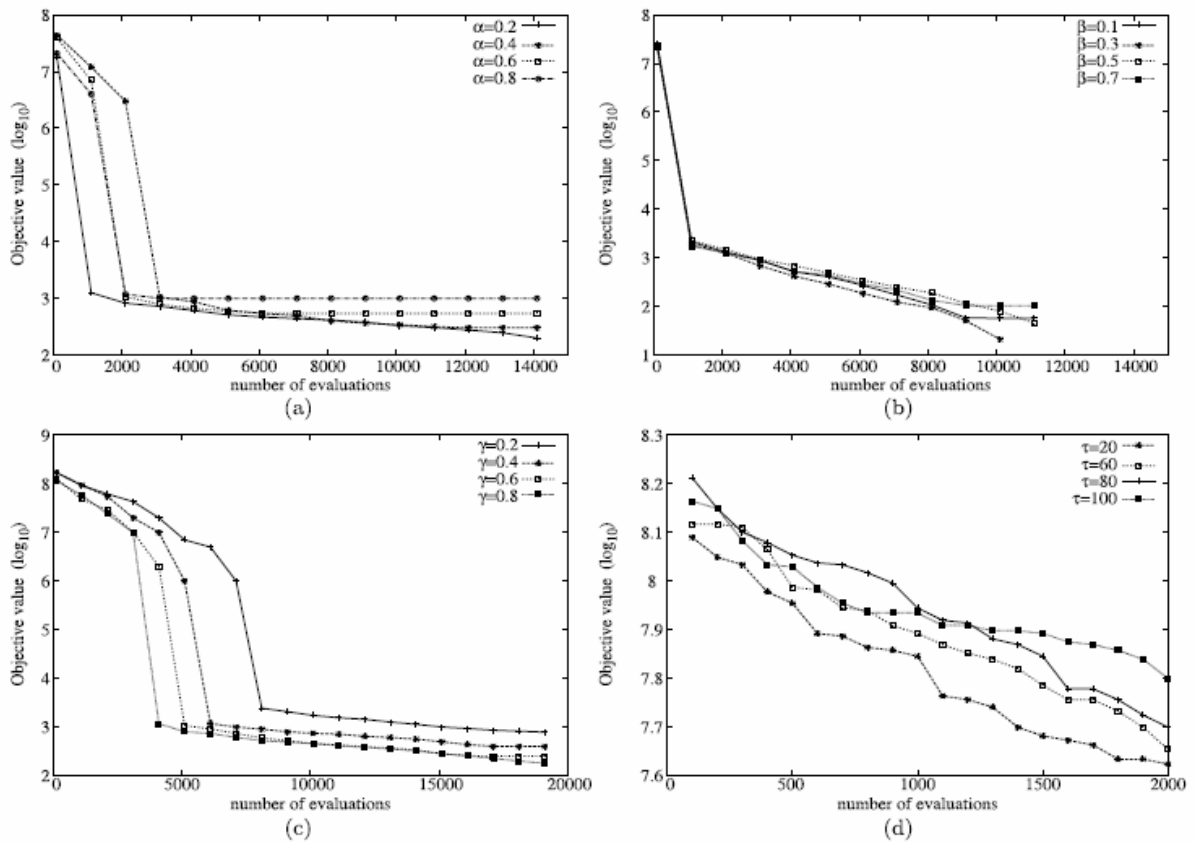
Από την Εικόνα 14, διάφορα αποτελέσματα μπορούν να παρατηρηθούν και αναλύονται παρακάτω. Πρώτον, η παράμετρος α έχει μία σημαντική επίδραση σχετικά με την απόδοση του HGATS για τον PECTP. Η απόδοση του HGATS πέφτει όταν η τιμή του α αυξάνεται από το 0,2 στο 0,8, βλέπε την Εικόνα 14a. Αυτό συμβαίνει επειδή επιλέγουμε ένα μικρό μέρος του πληθυσμού για τη δημιουργία της δομής δεδομένων MEM, η MEM μπορεί να προσφέρει μια ισχυρή καθοδήγηση κατά τη διάρκεια των γενετικών εργασιών και να βοηθήσει τον HGATS να εκμεταλλευτεί την περιοχή του χώρου αναζήτησης που αντιστοιχεί στα καλύτερα άτομα του πληθυσμού επαρκώς. Αυτή η επαρκής εκμετάλλευση μπορεί να εξασφαλίσει ότι ο HGATS επιτυγχάνει γρήγορα καλύτερες λύσεις. Σε αντίθεση, όταν ένα μεγάλο μέρος του πληθυσμού έχει ληφθεί για τη δημιουργία ή την ενημέρωση της MEM, τότε η MEM θα χάσει την επίδραση της στην καθοδήγηση του HGATS να εκμεταλλευτεί πολλά υποσχόμενους τομείς στον χώρο αναζήτησης. Με άλλα λόγια, όταν η α είναι ρυθμισμένη σε μεγάλες τιμές, ο HGATS τείνει να είναι GALS και, ως εκ τούτου, την απόδοση πέφτει ή είναι αδύναμη. Αυτό μπορεί να παρατηρηθεί από την Εικόνα 14a: όταν η τιμή του α αυξάνεται, η καλύτερη λύση του HGATS δεν μπορεί να βελτιωθεί μετά από ένα ορισμένο αριθμό των αξιολογήσεων, π.χ., μετά από περίπου 4000 αξιολογήσεις όταν $\alpha = 0,6$ και μετά από περίπου 2000 αξιολογήσεις όταν $\alpha = 0,8$.

Δεύτερον, σχετικά με την επίδραση της β , μια ενδιαφέρουσα συμπεριφορά του HGATS μπορεί να παρατηρηθεί στην περίπτωση προβλήματος 2007-17 με $\alpha = 0,2$, $\gamma = 0,8$, $\tau = 20$, και διαφορετικές τιμές β στην Εικόνα 14b. Από την Εικόνα 14b, μπορεί να φανεί ότι όταν η τιμή του β αυξάνεται από 0,1 στο 0,3, η απόδοση του HGATS βελτιώνεται λόγω της ενισχυμένης επίδρασης της δομής δεδομένων MEM. Ωστόσο, όταν η τιμή του β αυξάνεται περαιτέρω, η απόδοση του HGATS μειώνεται. Αυτό συμβαίνει επειδή, αν ένα μεγάλο μέρος των ατόμων που δημιουργείται μέσω της MEM, π.χ., όταν $\beta = 0,7$, η πιθανότητα της δημιουργίας ενός όμοιου παιδιού μπορεί να αυξάνεται κάθε γενιά και μετά από μερικές γενιές, ο HGATS μπορεί να παγιδευτεί σε μια υπό-βέλτιστη κατάσταση και ως εκ τούτου, δεν μπορεί να αποκτήσει την βέλτιστη λύση. Από την Εικόνα 14b, μπορεί να φανεί ότι η ρύθμιση της τιμής του β σε 0.5 ή 0.7 οδηγεί σε μια προηγούμενη στασιμότητα στην εκτέλεση του HGATS κατά τη διάρκεια της διαδικασίας επίλυσης.

Τρίτον, όσον αφορά την επίδραση της γ , από την Εικόνα 14c, μπορεί εύκολα να φανεί ότι η αύξηση της τιμής των αποτελεσμάτων γ οδηγεί σε καλύτερες λύσεις. Ο λόγος έγκειται στο γεγονός ότι η μικρή τιμή της γ οδηγεί στον προτεινόμενο GSGA να ενεργεί ως συμβατικός GA. Η επίδραση της γ δείχνει επίσης τη σημασία της θέσπισης της δομής δεδομένων MEM.

Τέταρτον, όσον αφορά την επίδραση της τ , μπορεί να φανεί από την Εικόνα 14d ότι ενημερώνοντας την MEM κάθε 20 γενιές δίνει μια καλύτερη απόδοση για τον HGATS από ότι η ενημέρωση της MEM κάθε 80 γενιές. Αυτό οφείλεται στο γεγονός ότι στην πρώτη περίπτωση ο χώρος αναζήτησης έχει διερευνηθεί περισσότερο από ότι στην τελευταία περίπτωση, η οποία αυξάνει την ποικιλία και δίνει μια μεγαλύτερη πιθανότητα να δημιουργηθούν καλύτερα άτομα. Η διαφορά είναι σημαντική όταν η τ έχει οριστεί από 20 μέχρι το 100. Αυτό συμβαίνει επειδή αυξάνοντας την τιμή του τ επιβραδύνει την ενημέρωση της δομής δεδομένων MEM επομένως υποβαθμίζει την αποτελεσματικότητα της MEM.

Με βάση τις ανωτέρω αναλύσεις παραμέτρων, για τα ακόλουθα πειράματα, θέτουμε τις παραμέτρους για τον HGATS ως εξής: $\alpha = 0.2$, $\beta = 0.3$, $\gamma = 0.8$, και $\tau = 20$.



Εικόνα 14: Σύγκριση της επίδρασης των παραμέτρων στην απόδοση του HGATS σε διαφορετικές περιπτώσεις του προβλήματος: a 2007-21 με $\beta = 0.3$, $\gamma = 0.8$ και $\tau = 20$, b 2007-17 με $\alpha = 0.2$, $\gamma = 0.8$ και $\tau = 20$, c 2007-11 με $\alpha = 0.2$, $\beta = 0.3$ και $\tau = 20$, και d 2007-03 με $\alpha = 0.2$, $\beta = 0.3$ και $\gamma = 0.8$

Σύγκριση με σχετικούς αλγορίθμους

Αυτό το σύνολο των πειραμάτων συγκρίνει την απόδοση του HGATS με δύο σχετικούς αλγορίθμους: ο ένας είναι ίδιος όπως ο προτεινόμενος GSGA εκτός από το ότι η τεχνική καθοδηγούμενης αναζήτησης είναι απενεργοποιημένη. Δηλαδή, είναι η τυπική σταθερή κατάσταση GA με LS1 και LS2, που υποδηλώνεται ως GALS σε αυτή τη μελέτη. Για τον GALS, ο τελεστής διασταύρωσης εφαρμόζεται με πιθανότητα διασταύρωσης $P_c = 0.8$. Ο

δεύτερος αλγόριθμος είναι ο αλγόριθμος TS. Το βασικό πλαίσιο του δοκιμαζόμενου αλγόριθμος TS είναι εμπνευσμένο από τους Rossi-Doria (2002) με τους ίδιους νέους φορείς γειτονιάς, όπως το tabu ευρετικό που χρησιμοποιείται στον HGATS. Οι ρυθμίσεις παραμέτρων που προσδιορίζονται για τον HGATS από τα προηγούμενα πειράματα χρησιμοποιήθηκαν σε HGATS και GALS (κατά περίπτωση) σε αυτό το τμήμα. Η *InitializeIndividual* () χρησιμοποιείται για τις αρχικές λύσεις για όλους τους αλγόριθμους, προκειμένου να έχουν μια δίκαιη σύγκριση των επιδόσεών τους. Υπήρχαν 50 εκτελέσεις του κάθε αλγόριθμου για κάθε στιγμιότυπο του προβλήματος. Ο χρόνος εκτέλεσης για κάθε εκτέλεση του αλγορίθμου σε κάθε περίπτωση του προβλήματος τέθηκε σε $t_{max} = 600$ δευτερόλεπτα με βάση την κατανομή του χρόνου που χρησιμοποιείται από τον ITC-2007. Άλλες ρυθμίσεις των παραμέτρων είναι οι εξής: ο πληθυσμός N μέγεθος ορίστηκε σε 50 και η πιθανότητα μετάλλαξης P_m ορίστηκε σε 0,5.

Οι αλγόριθμοι αξιολογούνται επί τη βάση των δύο τιμών, Df και SCP. Ο Πίνακας 20 παρουσιάζει τα αποτελέσματα των αλγορίθμων σε όρους του καλύτερου, του χειρότερου, κατά μέσο όρο, και της τυπικής απόκλισης των τιμών των Df και SCP για πάνω από τα 50 εκτελέσεις για τις 24 περιπτώσεις του προβλήματος. Από τον Πίνακα 20, μπορεί να φανεί ότι ο HGATS παράγει ένα χαμηλότερο μέσο όρο και την τυπική απόκλιση της αντικειμενικής αξίας για τα περισσότερες από τις περιπτώσεις του PECTP. Ο HGATS παράγει καλές λύσεις εξαιτίας της χρήσης της δομής δεδομένων MEM και τα συστήματα LS. Όπως αναφέρθηκε προηγουμένως, αυτό οφείλεται στο γεγονός ότι έχουμε αντιστοιχίσει σε ένα συμβάν ένα ζεύγος αίθουσας και χρονοθυρίδας που εξήλθε από ένα από τα καλύτερα άτομα των προηγούμενων πληθυσμών. Αυτό σημαίνει ότι το ζεύγος ικανοποιεί διαφορετικούς περιορισμούς που είναι κατάλληλοι για αυτή την περίπτωση. Οι τοπικές και tabu τεχνικές αναζήτησης βοηθούν περαιτέρω να βρεθεί το τοπικό βέλτιστο ενός ατόμου. Με αυτόν τον

τρόπο, θα αυξηθεί η πιθανότητα να έχουμε όλο και καλύτερες λύσεις κατά τη διάρκεια της διαδικασίας επίλυσης.

Η εικόνα 15 δείχνει την απόδοση των διαφορετικών αλγορίθμων όσον αφορά την αντικειμενική τιμή στην λογαριθμική κλίμακα έναντι του τον αριθμό των αξιολογήσεων. Από την εικόνα 15, μπορεί να φανεί στις περιπτώσεις προβλήματος 2007-14 και 2007-17, οι HGATS και TS επιτυγχάνουν μία λύση καθώς ο αριθμός των αξιολογήσεων αυξάνεται. Ο HGATS μειώνει σημαντικά στην αντικειμενική τιμή και δίνει μια βέλτιστη λύση μετά από 9000 και 4000 αξιολογήσεις, αντιστοίχως.

Τα *t*-test αποτελέσματα της έρευνας συγκρίνοντας στατιστικά αλγορίθμους με 98 βαθμούς ελευθερίας στο επίπεδο σημαντικότητας 0,05 φαίνονται στον Πίνακα 20. Στον Πίνακα 20, το *t*-test αποτέλεσμα δεικνύεται ως "s +", "s-", "+", "-", ή "~" όταν ο πρώτος αλγόριθμος είναι σημαντικά καλύτερος από, σημαντικά χειρότερος από, ασήμαντα καλύτερος από, ασήμαντα χειρότερος από, ή στατιστικά ισοδύναμος με το δεύτερο αλγόριθμο, αντίστοιχα.

Από τον Πίνακα 20, μπορεί να φανεί ότι η απόδοση του HGATS είναι σημαντικά καλύτερη από την απόδοση των άλλων δύο αλγορίθμων στις περισσότερες περιπτώσεις του προβλήματος. Μπορεί, επίσης να έχει παρατηρηθεί ότι η απόδοση του GALS είναι σημαντικά καλύτερη από την απόδοση του TS πρόβλημα στις περισσότερες περιπτώσεις του προβλήματος. Αυτό το αποτέλεσμα δείχνει ότι ένα μόνο ευρετικό δεν είναι αρκετό για την επίλυση ενός PECTP. Αυτά τα αποτελέσματα δείχνουν ότι η ενσωμάτωση των κατάλληλων LS τεχνικών με ξεναγό τεχνικές αναζήτησης μπορεί να βελτιώσει σημαντικά την απόδοση των γενετικών αλγορίθμων για το PECTP.

Για να δούμε το όφελος της εισαγωγής της δεύτερης φάσης (δηλαδή, το ευρετικό TS) στον HGATS, έχουμε επίσης καταγράψει το ποσοστό των εφικτών λύσεων που λαμβάνονται από τον HGATS μετά την Φάση I και Φάση II μετά από 50 εκτελέσεις σε κάθε περίπτωση του

PECTP προβλήματος του ITC-2007. Τα αποτελέσματα φαίνονται στον Πίνακα 21. Από τον Πίνακα 21, μπορεί να φανεί ότι η το TS ευρετικό είναι επωφελές για η απόδοση του HGATS στις περισσότερες περιπτώσεις του PECTP της δοκιμής.

Σύγκριση με αλγόριθμους από τη βιβλιογραφία

Σε αυτήν την ενότητα, προκειμένου να δικαιολογηθεί η απόδοση του προτεινόμενου αλγόριθμου, συγκρίνουμε τα πειραματικά αποτελέσματα του HGATS με βάση τα διαθέσιμα αποτελέσματα άλλων αλγορίθμων για τις περιπτώσεις προβλήματος του PECTP του ITC-2007. Ένας άλλος λόγος για τη σύγκριση των αποτελεσμάτων με βάση τα διαθέσιμα αποτελέσματα είναι ότι είναι ενδιαφέρον να φανεί η συμπεριφορά των GAs για προβλήματα PECTPs με μεγάλους περιορισμούς μεταξύ των διαφορετικών ευρετικών μεθόδων και μεθόδων βελτιστοποίησης, οι οποίες δεν έχουν ερευνηθεί ακόμα στην βιβλιογραφία. Οι αλγόριθμοι που συγκρίνονται περιγράφονται ως ακολούθως:

- HGATS: Η υβριδική προσέγγιση που προτείνεται στην παρούσα εργασία.
- Μικτή μετά-ευρετική προσέγγιση (MMA): Ο Cambazard (2008), πρότεινε την MMA, η οποία περιλαμβάνει TS και προσομοιωμένη ανόπτηση που χρησιμοποιείται σε συνδυασμό με διάφορους γειτονικούς φορείς.
- CTI: Ο Atsuta (2008) πρότεινε μια τεχνική που είναι ο συνδυασμός ενός λύτη ικανοποίησης περιορισμών γενικού σκοπού, TS, και επαναλαμβανόμενων LS τεχνικών.
- Υβριδικός αλγόριθμος (HA): Ο Chiarandini (2008) πρότεινε έναν HA που συνδυάζει μια εποικοδομητική διαδικασία για την επίτευξη της σκοπιμότητας, ακολουθούμενη από LS και προσομοιωμένη ανόπτηση για την ικανοποίηση των ελαστικών περιορισμών.

- ACO: Ο Nothegger (2008) πρότεινε έναν αλγόριθμο ACO σε συνδυασμό με μία τοπική ρουτίνα αναζήτησης βελτίωση.

- Αλγόριθμος βασισμένος σε LS (LSA): Ο Müller (2008) χρησιμοποίησε έναν LSA με ρουτίνες που λαμβάνονται από τη Βιβλιοθήκη Επίλυσης Περιορισμών. Διάφοροι αλγόριθμοι αναζήτησης γειτονιάς, επίσης, χρησιμοποιούνται να εξαλειφθούν παραβιάσεις των αυστηρών και ελαστικών περιορισμών.

Ο Πίνακας 22 δίνει τα αποτελέσματα της σύγκρισης, όπου ο όρος «Df» αντιπροσωπεύει την απόσταση προς το εφικτό και ο "BSCP" σημαίνει η καλύτερη τιμή SCP πάνω από 10 εκτελέσεις. Ένα πράγμα που πρέπει να λάβετε υπόψη είναι ότι τα αποτελέσματα του διαγωνισμού ITC-2007 των άλλων αλγορίθμων βασίστηκαν στις 10 εκτελέσεις ανά περίπτωση. Για ορθή σύγκριση, παρουσιάζονται επίσης, τα αποτελέσματά μας με βάση 10 εκτελέσεις ανά παράδειγμα εδώ.

Από τον Πίνακα 22, μπορεί να φανεί ότι ο προτεινόμενος HGATS επιτυγχάνει την σκοπιμότητα για όλες τις περιπτώσεις του προβλήματος για πάνω από 10 εκτελέσεις. Μπορεί επίσης να φανεί ότι η πιθανότητα του HGATS να πάρει βέλτιστες λύσεις είναι υψηλότερη από ό, τι άλλων αλγορίθμων. Ο HGATS πέτυχε την βέλτιστη λύση στις 10 από τις 24 περιπτώσεις του προβλήματος. Δίνει το καλύτερο αποτέλεσμα στις περιπτώσεις του προβλήματος 2007-4, 2007-5, 2007-16, 2007-20 πάνω από όλους τους συγκρινόμενους αλγορίθμους. Από τα αποτελέσματα, μπορούμε να δούμε ότι η οργανωμένη στρατηγική αναζήτησης και ο κατάλληλος συνδυασμός των τοπικών και tabu προσεγγίσεις αναζήτησης μπορεί να βοηθήσει στην ελαχιστοποίηση των αντικειμενικών τιμών και να δώσει καλύτερα αποτελέσματα για το PECTP σε σύγκριση με άλλες πληθυσμιακά και ευρετικά βασισμένους αλγορίθμους στην βιβλιογραφία.

Πίνακας 20: Σύγκριση των αλγόριθμων για διαφορετικές περιπτώσεις του προβλήματος.

PECTP	Alg	Best		Worse		Average		Std	
		Df	SCP	Df	SCP	Df	SCP	Df	SCP
2007-1	TS	0	1069	51	1732	11.33	1202.6	17.55	257.04
	GALS	0	641	12	976	3.44	704.89	5.13	152.33
	HGATS	0	501	12	842	0	587	1.84	108.61
2007-2	TS	0	989	72	2213	25.87	1191.22	17.44	386.77
	GALS	0	747	50	2311	8.78	1005.11	16.87	504.79
	HGATS	0	342	0	695	0	476.2	0	96.94
2007-3	TS	0	756	18	821	5.89	794.33	5.84	21.9
	GALS	0	509	0	801	0	697.44	0	129.53
	HGATS	0	3770	432	0	0	407.78	0	19.73
2007-4	TS	0	794	76	1130	18.33	910.5	29.44	141.45
	GALS	0	521	11	791	2	669	4.09	46
	HGATS	0	234	4	524	0	369	0.33	26.72
2007-5	TS	0	496	65	678	22	544.2	9.62	884
	GALS	0	98	20	310	8.62	154	9.62	78
	HGATS	0	0	0	325	0	118	0	88.05
2007-6	TS	0	218	0	788	0	428	0	272.93
	GALS	0	10	0	430	0	207	0	134.18
	HGATS	0	0	0	342	0	201	0	139.5
2007-7	TS	0	84	198	508	82	258	66.59	183.74
	GALS	0	275	70	489	25.5	381.75	35.20	89.531
	HGATS	0	0	2	543	0.53	418	1.62	98.404
2007-8	TS	0	0	0	751	0	481	0	315.7
	GALS	0	0	0	424	0	322	0	193.1
	HGATS	0	0	0	309	0	257.12	0	120.78
2007-9	TS	0	1711	152	2361	40.12	1797	53.04	294.85
	GALS	0	1547	115	2141	28.87	1237	39	412.37
	HGATS	0	989	42	1183	4.5	1002	20.09	81.12
2007-10	TS	0	763	0	1978	0	999	0	406
	GALS	4	548	26	1040	5	850	9	154
	HGATS	0	499	0	810	0	614	0	117
2007-11	TS	0	680	0	1980	0	968	0	414
	GALS	0	701	0	984	0	897	0	84
	HGATS	0	246	0	691	0	452	0	121
2007-12	TS	0	373	56	1563	17	702	23	393
	GALS	0	444	0	984	0	576	0	178
	HGATS	0	172	13	546	1.625	226	0.59	129
2007-13	TS	0	624	20	1873	6.37	1230	9.1	380
	GALS	0	201	0	1639	0	852	0	392
	HGATS	0	0	0	717	0	616	0	249
2007-14	TS	0	241	17	416	4.75	287	7.62	76
	GALS	0	61	0	104	0	78.2	0	17
	HGATS	0	0	0	19	0	4.125	0	7.29

Πίνακας 20: (Συνέχεια)

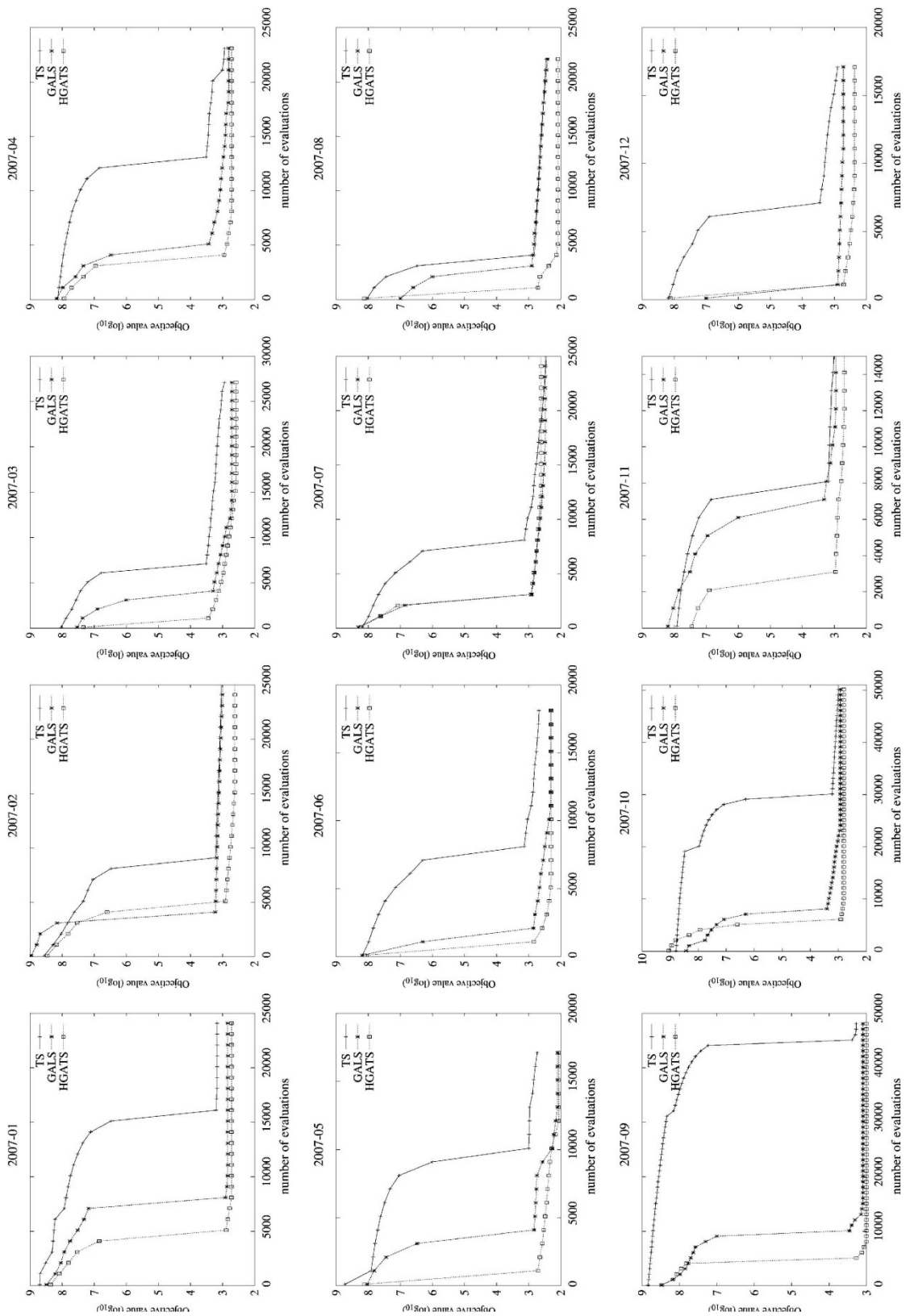
PECTP	Alg	Best		Worse		Average		Std	
		Df	SCP	Df	SCP	Df	SCP	Df	SCP
2007-15	TS	0	101	0	164	0	135	0	33
	GALS	0	14	0	97	0	69	0	21
	HGATS	0	0	0	37	0	26	0	6.54
2007-16	TS	0	109	0	1158	0	563	0	161
	GALS	0	168	0	771	0	377	0	195
	HGATS	0	0	0	270	0	168	0	115.27
2007-17	TS	0	0	0	42	0	32	0	10
	GALS	0	0	0	21	0	5	0	7.4
	HGATS	0	0	0	11	0	2.5	0	4.65
2007-18	TS	0	0	0	1241	0	924	0	420.52
	GALS	0	0	0	842	0	631	0	270
	HGATS	0	0	0	572	0	446	0	108
2007-19	TS	147	1078	346	1867	138	1372	110	334
	GALS	0	1015	430	2693	174	1612	154	673.65
	HGATS	0	84	319	1900	133	810	115	513.7
2007-20	TS	40	348	113	1192	71	1100	29	133
	GALS	0	318	138	1942	67	1199	86	439
	HGATS	0	297	234	2305	75	1274	95	622
2007-21	TS	0	137	261	1162	69.5	805	96	267
	GALS	0	0	10	621	22.5	305	4.6	241
	HGATS	0	0	15	1359	2.5	780	2	422
2007-22	TS	91	1742	102	2439	97.37	2051	4.47	260.6
	GALS	42	1579	188	2466	94	1715	42	396
	HGATS	0	1142	73	1315	33.125	1196	38	243
2007-23	TS	0	2062	34	5556	362	1604	16	8.75
	GALS	11	1001	43	1291	81	1193	13	20
	HGATS	0	963	16	1896	1.2	1152	3.6	2
2007-24	TS	0	629	0	2309	0	1407	0	541
	GALS	0	368	9	2007	2.25	1112	4.16	463.6
	HGATS	0	274	0	2142	0	1002	0	519

Συμπέρασμα και μελλοντική εργασία

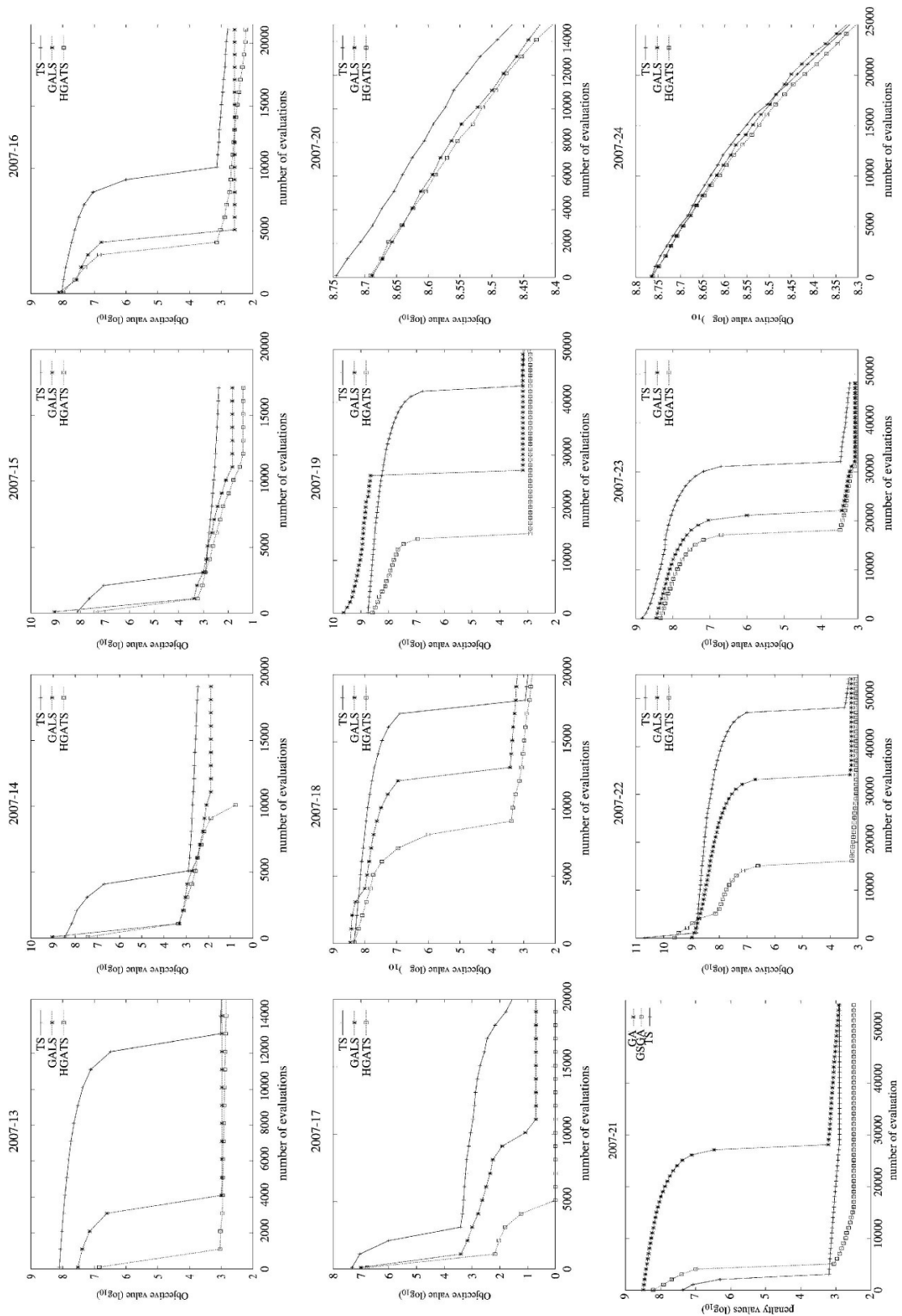
Στον GSGA, μια οργανωμένη στρατηγική αναζήτησης χρησιμοποιεί μια δομή δεδομένων για την αποθήκευση χρήσιμων πληροφοριών, δηλ., ένας κατάλογος από αίθουσες και ζεύγη χρονοθυρίδων για κάθε συμβάν που εξάγεται από τα καλύτερα άτομα που επιλέγονται από τον πληθυσμό και έχει μηδενική τιμή ποινής. Αυτή η δομή δεδομένων χρησιμοποιείται για να

κατευθύνει η παραγωγή απογόνων στον επόμενο πληθυσμό. Το κύριο πλεονέκτημα αυτής της δομής δεδομένων έγκειται στο ότι παρέχει τμήματα πρώην καλών λύσεων, οι οποίες διαφορετικά θα είχαν χαθεί κατά τη διαδικασία επιλογής, και επαναχρησιμοποιεί τις αποθηκευμένες πληροφορίες στις επόμενες γενιές. Αυτό μπορεί να επιτρέψει στον αλγόριθμο να ανακτήσει γρήγορα τις καλύτερες λύσεις που αντιστοιχούν με τις προηγούμενους και νέους πληθυσμούς. Στον προτεινόμενο HGATS αλγόριθμο, δύο LS τεχνικές που χρησιμοποιούνται για τη βελτίωση της ποιότητας των ατόμων μέσω της αναζήτησης έξι γειτονικών δομών. Δεδομένου ότι η δεύτερη φάση του HGATS, ένα TS σύστημα χρησιμοποιείται για να βελτιώσει περαιτέρω την καλύτερη λύση από τον GSGA στην πρώτη φάση.

Εικόνα 15: Δυναμική απόδοση των αλγορίθμων για διαφορετικές περιπτώσεις του προβλήματος.



Εικόνα 15: (Συνέχεια)



Πίνακας 21: Οι τιμές t -test συγκρινόμενων αλγορίθμων για διαφορετικές περιπτώσεις του προβλήματος.

PECTP	Df			SCP		
	HGATS-GALS	HGATS-TS	GALS-TS	HGATS-GALS	HGATS-TS	GALS-TS
2007-1	—	s+	s+	+	s+	s+
2007-2	s+	s+	s+	s+	s+	+
2007-3	~	+	+	s+	s+	s+
2007-4	+	s+	s+	s+	s+	+
2007-5	~	~	~	+	+	+
2007-6	s+	s+	s+	s+	s+	+
2007-7	~	~	~	+	s+	s+
2007-8	s+	s+	+	+	s+	+
2007-9	+	~	—	+	s+	s+
2007-10	~	~	~	s+	s+	+
2007-11	s—	s+	s+	s+	s+	+
2007-12	~	s+	s+	s+	s+	+
2007-13	~	s+	s+	+	s+	s+
2007-14	~	s+	s+	s+	s+	s+
2007-15	~	~	~	s+	s+	s+
2007-16	~	~	~	s+	s+	s+
2007-17	~	~	~	+	s+	s+
2007-18	~	~	~	+	s+	+
2007-19	+	+	+	s+	s—	s—
2007-20	+	+	+	s—	s—	+
2007-21	s+	s+	s+	s—	+	s+
2007-22	s+	s—	s—	s+	s+	s+
2007-23	s+	~	s+	s+	s+	s+
2007-24	s+	~	s—	+	s+	+

Πίνακας 22: Ποσοστό εφικτών λύσεων που λαμβάνονται από τον HGATS μετά τη φάση I και φάση II πάνω από 50 εκτελέσεις σε κάθε περίπτωση του προβλήματος του ITC-2007.

HGATS Phase	Problem Instances											
	1	2	3	4	5	6	7	8	9	10	11	12
Phase I (GSGA)	52%	64%	92%	96%	82%	78%	78%	100%	48%	52%	100%	80%
Phase II (TS)	92%	100%	100%	98%	100%	100%	94%	100%	82%	100%	100%	96%
HGATS Phase	Problem Instances											
	13	14	15	16	17	18	19	20	21	22	23	24
Phase I (GSGA)	86%	86%	100%	90%	100%	100%	20%	26%	30%	48%	48%	50%
Phase II (TS)	100%	100%	100%	100%	100%	100%	54%	68%	94%	70%	96%	100%

3. Κατάρτιση Προγράμματος εξεταστικής (Examination Timetabling)

Δεδομένα αναφοράς κατάρτισης προβλήματος εξεταστικής

Το υψηλό επίπεδο του ερευνητικού ενδιαφέροντος στην Κατάρτιση Προγράμματος εξεταστικής οδήγησε στην δημιουργία από μία ποικιλία διαφορετικών προβλημάτων αναφοράς που έχουν μελετηθεί ευρέως. Τα θεσπισμένα κριτήρια αξιολόγησης, με τις παραλλαγές των καθορισμένων προτύπων, έδωσε έναν τρόπο για ουσιαστικές επιστημονικές συγκρίσεις και την ανταλλαγή των επιτευγμάτων της έρευνας. Ωστόσο, υπήρξε κάποια σύγχυση στη βιβλιογραφία που οφείλεται στην κυκλοφορία δύο διαφορετικών εκδόσεων των οκτώ από αυτά τα προβλήματα αναφοράς (από τα σύνολα δεδομένων του Πανεπιστήμιου του Τορόντο). Αυτό, φυσικά, σημαίνει ότι έχουμε πραγματικά 21 προβλήματα που έχουν μελετηθεί στην βιβλιογραφία (αντί 13). Ένας στόχος του παρόντος τμήματος του εγγράφου είναι να συνοψίσει ποιες από τις μεθόδους που έχουν εμφανιστεί στην βιβλιογραφία είναι οι καλύτερες σε αυτά τα σημεία αναφοράς. Αυτό είναι ιδιαίτερα σημαντικό δεδομένου της σύγχυσης που αναφέρεται παραπάνω.

3.1. Δεδομένα Αναφοράς του Πανεπιστήμιου του Τορόντο

Οι **Carter, Laptore** και **Lee** το 1996, εισήγαγαν μια σειρά από 13 πραγματικού κόσμου προβλήματα χρονοπρογραμματισμού εξετάσεων από τρία λύκεια, πέντε καναδικά πανεπιστήμια, ένα αμερικανικό πανεπιστήμιο, ένα βρετανικό πανεπιστήμιο και ένα πανεπιστήμιο στη Σαουδική Αραβία. Κατά τη διάρκεια των ετών είχαν δοκιμαστεί ευρέως στην έρευνα χρονοπρογραμματισμού εξετάσεων από διαφορετικές state-of-the-art προσεγγίσεις και έχουν θεωρηθεί ως ένα σημείο αναφοράς στο πεδίο. Όπως αναφέρθηκε παραπάνω, έχει υπάρξει ένα ζήτημα σχετικά με την κυκλοφορία των διαφορετικών ομάδων με το ίδιο όνομα.

Στο πρόβλημα, για να αναδειχθεί η πυκνότητα των συγκρουόμενων εξετάσεων σε κάθε μία από τις περιπτώσεις, μία *Conflict Matrix* C ορίστηκε όπου κάθε στοιχείο $c_{ij} = 1$ αν η εξέταση i

συγκρουστεί με την εξέταση j (έχουν κοινούς φοιτητές), ή $c_{ij} = 0$ διαφορετικά. Η *Conflict Density* αντιπροσωπεύει την αναλογία μεταξύ του αριθμού των στοιχείων με αξία "1" στο συνολικό αριθμό των στοιχείων στο πλέγμα των συγκρούσεων.

Δύο παραλλαγές των στόχων ορίστηκαν:

- Να ελαχιστοποιηθεί ο αριθμός των χρονικών θυρίδων που απαιτούνται για το πρόβλημα (χρωματισμός γραφήματος) (ονομάζεται ως Toronto a στον Πίνακα 24) και
- Να ελαχιστοποιηθεί ο μέσος όρος του κόστους ανά φοιτητή (που ονομάζεται Toronto b στον Πίνακα 23).

Για τον 1ο στόχο, ο σκοπός είναι να βρεθούν τα εφικτά χρονοδιαγράμματα με το συντομότερο μήκος. Για τον 2ο στόχο, μια λειτουργία αξιολόγησης ορίστηκε για τον υπολογισμό του κόστους των χρονοδιαγραμμάτων που δημιουργούνται. Για τους φοιτητές που δίνουν δύο εξετάσεις χρονοθυρίδες s , το κόστος ανατέθηκε χρησιμοποιώντας τιμές εγγύτητας w_s , δηλαδή $w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$ και $w_5 = 1$. Ο στόχος είναι να διαχωριστούν οι συγκρουόμενες εξετάσεις μέσα σε ένα περιορισμένο αριθμό χρονοθυρίδων. Οι συγγραφείς εισήγαγαν επίσης επτά πραγματικές εφαρμογές με πλευρικούς περιορισμούς (δηλαδή μέγιστη χωρητικότητα αίθουσας ανά χρονοθυρίδα, προ-εκχωρημένες εξετάσεις, μέγιστος αριθμός εξετάσεων ανά χρονοθυρίδα, δεν υπάρχουν εξετάσεις x στις y χρονοθυρίδες, κλπ). Ο στόχος αυτός αλλάχτηκε αργότερα και δοκιμάστηκε από έναν αριθμό προσεγγίσεων.

Κατά τη διάρκεια των χρόνων, ωστόσο, δύο παραλλαγές των δεδομένων κυκλοφόρησαν και δοκιμάστηκαν από διαφορετικές προσεγγίσεις. Για να διακριθούν τα στοιχεία που ελέγχονται και να δημιουργηθεί ένα πρότυπο αναφοράς για μελλοντική χρήση στο χρονοδιάγραμμα, έχουμε εξετάσει προσεκτικά τα δεδομένα που έχουν εμφανιστεί σε δύο διαφορετικές μορφές υπό το ίδιο όνομα για οκτώ από αυτά τα προβλήματα συγκριτικής μέτρησης επιδόσεων.

Παραθέτουμε τα χαρακτηριστικά των δύο αυτών εκδόσεων δεδομένων στον Πίνακα 23. Έχουμε καθορισμένο εκ των υστέρων "I" και "II" αντίστοιχα στα σύνολα δεδομένων που κυκλοφορούν για να γίνει διάκριση μεταξύ τους. Η αναφορά "I" έχει χρησιμοποιηθεί για την περίπτωση του προβλήματος το οποίο πιστεύουμε ότι έχει εμφανιστεί πιο συχνά στην βιβλιογραφία. Για τις περιπτώσεις του προβλήματος της αναφοράς "II", κάποια σύγκριση συνέβη καθώς τρεις από τις περιπτώσεις (car91 I, και car92 II pur93 II) έχουν συγκρούσεις με τον αριθμό των εγγραφών (δηλαδή ένας διαφορετικός αριθμός εισακτέων ορίζεται σε δύο αρχεία δεδομένων για κάθε περίπτωση). Αργότερα σε αυτό το τμήμα θα προσπαθήσουμε να ρίξουμε φως σχετικά με την ερώτηση ποιός τεχνικός έχει εφαρμοστεί σε ποια έκδοση από αυτές τις περιπτώσεις στην λογοτεχνία.

Πίνακας 23: Χαρακτηριστικά δύο εκδόσεων των συνόλων δεδομένων αναφοράς του Τορόντο

Problem Instance	Exams	Students	Enrolments	Conflict Density	Timeslots
car91 I	682	16925	56877	0.13	35
car91 II	682	16925	56242/56877	0.13	35
car92 I	543	18419	55522	0.14	32
car92 II	543	18419	55189/55522	0.14	32
ear83 I	190	1125	8109	0.27	24
ear83 II	189	1108	8014	0.27	24
hec92 I	81	2823	10632	0.42	18
hec92 II	80	2823	10625	0.42	18
kfu93	461	5349	25113	0.06	20
lse91	381	2726	10918	0.06	18
pur93 I	2419	30032	120681	0.03	42
pur93 II	2419	30032	120686/120681	0.03	42
rye92	486	11483	45051	0.07	23
sta83 I	139	611	5751	0.14	13
sta83 II	138	549	5689	0.14	13
tre92	261	4360	14901	0.06	23
uta92 I	622	21266	58979	0.13	35
uta92 II	638	21329	59144	0.13	35
ute92	184	2749	11793	0.08	10
yor83 I	181	941	6034	0.29	21
yor83 II	180	919	6012	0.29	21

Πίνακας 24: Παραλλαγές των συνόλων δεδομένων αναφοράς του Τορόντο

	Variants	Objectives
Toronto a	graph coloring	to minimise the number of timeslots needed
Toronto b	un-capacitated with cost	to space out conflicting exams within limited (fixed number of) timeslots
Toronto c	capacitated with cost	to minimise students sitting two exams in a row on the same day
Toronto d	capacitated with modified cost	same as above, and to minimise students sitting two exams overnight
Toronto e	estimated capacity and timeslots	to minimise students sitting two adjacent exams the same day

Οι **Burke, Newall** και **Weare** το 1996, τροποποίησαν τον στόχο από έξι προβλήματα του πραγματικού κόσμου λαμβάνοντας υπόψη τη μέγιστη χωρητικότητα της αίθουσας ανά χρονοθυρίδα, και τις παρακείμενες εξετάσεις την ίδια ημέρα. Το 1998, οι χρονοθυρίδες στα προβλήματα διακρίθηκαν, ορίζοντας τρεις χρονοθυρίδες την ημέρα από Δευτέρα έως Παρασκευή και μία χρονοθυρίδα το Σάββατο. Ο στόχος είναι η ελαχιστοποίηση των φοιτητών που έχουν δύο διαδοχικές εξετάσεις την ίδια ημέρα και τη νύχτα. Αυτές οι δύο παραλλαγές ονομάζονται Toronto c και Toronto d στον Πίνακα 24. Ο **Terashima-Marin** το 1999 τροποποίησε το σύνολο δεδομένων με την ανάθεση σε κάθε στιγμιότυπο του προβλήματος έναν κατ'εκτίμηση αριθμό χρονοθυρίδων και σε κάθε χρονοθυρίδα υπολογίζεται ένα μέγιστο καθισμάτων / χωρητικότητας. Αυτή η παραλλαγή ονομάζεται Toronto e στον Πίνακα 24.

Οι προσεγγίσεις που αναπτύσσονται και δοκιμάζονται σε διαφορετικές παραλλαγές των συνόλων δεδομένων του Τορόντο κατά τη διάρκεια των ετών, αναφέρονται στον Πίνακα 25 (ταξινομημένα με το έτος κατά το οποίο το έργο δόθηκε στη δημοσιότητα). Οι τιμές σε "()" ακολουθώντας τις παραλλαγές των δεδομένων στον Πίνακα 25 δίνουν τον αριθμό των περιπτώσεων του προβλήματος που δοκιμάστηκαν με τις αντίστοιχες προσεγγίσεις. Το μεγαλύτερο μέρος της εργασίας δεν προσδιορίζει τα ακριβή χαρακτηριστικά των δεδομένων που έχουν δοκιμαστεί, και σε πολλά από τα έγγραφα είναι αδύνατον να καθοριστεί ποια έκδοση (I ή II) των δεδομένων ελέγχθηκαν (για τις οκτώ προβληματικές περιπτώσεις).

3.2. Δεδομένα Αναφοράς του Πανεπιστήμιου του Νόττινχαμ

Οι **Burke, Newall** και **Weare** το 1996 εισήγαγαν επίσης τα δεδομένα του 1994 χρονοδιαγράμματος εξετάσεων στο Πανεπιστήμιο του Nottingham ως σημείο αναφοράς. Χρησιμοποιήθηκε αργότερα από έναν αριθμό ερευνητών στην δοκιμή διαφορετικών προσεγγίσεων. Ο Πίνακας 26 παρουσιάζει τα χαρακτηριστικά του συνόλου δεδομένων. Γνωρίζουμε ότι το 23 είναι ο λιγότερο δυνατός αριθμός των χρονοθυρίδων λόγω των περιορισμών στην χωρητικότητα της αίθουσας. Ο στόχος είναι να ελαχιστοποιηθούν οι μαθητές που έχουν δύο διαδοχικές εξετάσεις την ίδια ημέρα.

Πίνακας 25. Προσεγγίσεις για τα σύνολα δεδομένων αναφοράς του Τορόντο

Reference	Approach/Technique	Problem
Carter et al [51] 1996	Graph heuristics with clique initialisation and backtracking	a(13), b(13)
Carter&Johnson [48] 1996	Almost cliques with sufficient density as the initialisation for graph heuristics	a(13)
Burke et al [38] 1996	Memetic Algorithm with hill climbing and light and heavy mutation	c(5)
Burke et al [39] 1998	Different initialisation strategies in Memetic Algorithms measured by diversity	d(3)
Burke et al [40] 1998	Non-determinisms introduced by selection strategies in graph heuristics	d(3)
Burke&Newall [35] 1999	Multi-stage Evolutionary Algorithm based on Memetic Algorithm	d(3)
Terashima -Marin et al [138] 1999	Genetic Algorithm with in-direct coding of the constructive strategies and heuristics	e(12)
Caramia et al [45] 2001	Iterated algorithm with novel improving factors	a(13), b(13), c(5)
Di Gaspero [71] &Schaerf 2001	Adaptive tabu list and cost function in Tabu Search	b(11), c(5), d(3)
Di Gaspero [70] 2002	Multiple neighbourhood Tabu Search	b(7),d(3)
White&Xie [144] [145] 2001&2004	Tabu Search with long term memory Relaxation on long and short term tabu lists	b(2) b(7)
Paquete& [113] Stutzle 2002	Tabu Search with Lex-tie and Lex-seq strategies in the objective function	b(8)
Merlot et al [104] 2003	Constraint programming as initialisation for Simulated Annealing and hill climbing	a(12), b(12), c(5), d(2)
Casey& [52] Thompson 2003	GRASP with modified Saturation Degree initialisation and Simulated Annealing improvement	b(10)
Burke&Newall [36] 2003	Great Deluge with adaptive ordering as the initialisation	b(11)
Burke&Newall [37] 2004	Graph heuristics with adaptive heuristic modifier to dynamically order the exams	b(11), d(3)
Burke&Bykov et al [16] 2004	Time-predefined Great Deluge and Simulated Annealing	b(13), d(2)
Asmuni et al [8] 2004	Fuzzy rules with Largest Degree, Saturation Degree and Largest Enrolment	b(12)
Ross et al [128] 2004	Genetic Algorithm evolving constructive strategies and heuristics	e(12)
Burke et al [20] 2005	Hybridising graph heuristics in hyper-heuristic by CBR and systematic strategies	b(4)
Cote et al [61] 2005	Bio-objective Evolutionary Algorithm with local search operators in the recombination process	b(12)
Kendall&Hussin [90] 2005	Tabu Search based hyper-heuristic	b(8)
Yang&Petrovic [149] 2005	Similarity measure using fuzzy set on selecting hybridisations of Great Deluge and graph heuristics	b(12)

Πίνακας 25 (συνέχεια). Προσεγγίσεις για τα σύνολα δεδομένων αναφοράς του Τορόντο

Reference	Approach/Technique	Problem
Abdullah et al [3] 2006	Large neighbourhood search with tree-based neighbourhood structure	b(12), c(5)
[4] 2006	Tabu Search based large neighborhood search	c(5)
Burke et al [34] 2006	Graph based hyper-heuristic using Tabu Search	b(11)
Qu&Burke [118] 2006	Graph based hyper-heuristic framework with different high level search algorithms	b(11)
Burke et al [22] 2006	Genetic Algorithms on selecting subset of neighborhoods in Variable Neighborhood Search	b(11)
Burke et al [42] 2006	Case based heuristic selection for the solution construction	b(11)

Το (1998), τα παραπάνω προβλήματα περιορίζονται περαιτέρω τροποποιώντας την αντικειμενική συνάρτηση να εξετάσει επίσης τις συνεχόμενες εξετάσεις κατά τη διάρκεια της νύχτας. Στον Πίνακα 26 μπορούμε να επισημάνουμε αυτές τις παραλλαγές, ως Nottingham *a* και *b*. Ο Πίνακας 27 παρουσιάζει τις προσεγγίσεις που εφαρμόζονται σε αυτές τις βάσεις δεδομένων και στα σύνολα δεδομένων του Πανεπιστημίου της Μελβούρνης (βλ. παράγραφο 3.3 κατωτέρω) στην βιβλιογραφία.

Πίνακας 26. Χαρακτηριστικά των συνόλων δεδομένων αναφοράς του Nottingham

	Nottingham <i>a</i>	Nottingham <i>b</i>
Exams	800	800
Students	7896	7896
Timeslots	23, 26	23
Enrolments	34265	34265
Conflicts	10034	10034
Capacity	1550	1550
Density	0.03	0.03
Objective	minimise adjacent exams on the same day	minimise adjacent exams on the same day and overnight

3.3. Δεδομένα Αναφοράς του Πανεπιστημίου της Μελβούρνης

Ο **Merlot** εισήγαγε σύνολα δεδομένων χρονοδιαγράμματος εξετάσεων από το Πανεπιστήμιο της Μελβούρνης στο συνέδριο PATAT το 2002. Δύο σύνολα δεδομένων εισήχθησαν. Για αυτά τα σύνολα δεδομένων, υπήρχαν δύο χρονοθυρίδες για κάθε ημέρα για κάθε μία από τις πέντε εργάσιμες ημέρες, και η χωρητικότητα της κάθε συνεδρίας να ποικίλει. Η διαθεσιμότητα των συνεδριών για ορισμένες από τις εξετάσεις ήταν περιορισμένη. Σε μία περίπτωση του προβλήματος αυτό εμπόδισε όλες τις εφικτές λύσεις έτσι ένα εναλλακτικό σύνολο δεδομένων δημιουργήθηκε που επιτρέπει εφικτές λύσεις.

Πίνακας 27. Προσεγγίσεις για τα σύνολα δεδομένων αναφοράς του Νότινχαμ και της Μελβούρνης.

Reference	Approach/Technique	Problem
Burke&Newall [37] 2004	Adaptive ordering based on graph heuristics using heuristic modifier	Nottingham <i>b</i>
Merlot et al [104] 2003	Constraint logic programming as initialisation for Simulated Annealing and hill climbing	Nottingham <i>a,b</i> Melbourne <i>I II</i>
Di Gaspero& Schearf [71] 2001	Tabu Search using exhaust and biased selection by the costs of exams	Nottingham <i>a,b</i>
Burke et al [16] 2004	Great Deluge with a number of runs of Saturation Degree	Nottingham <i>b</i>
Caramia et al [45] 2001	Iterative approach where the number of timeslots was gradually increased after greedy improvement	Nottingham <i>a</i>
Ahmadi et al [5] 2003	VNS to search permutations of heuristics and their weights	Nottingham <i>a</i>
Cote et al [61] 2005	Evolutionary algorithms with bio-objective constraint satisfaction	Nottingham <i>a</i> Melbourne <i>I</i>
Burke et al [38] 1996	Memetic algorithms with light & heavy mutations graph heuristic initialisation	Nottingham <i>a</i>
Burke&Newall [35] 1999	Multi-stage evolutionary algorithm initialised by graph heuristics with backtracking	Nottingham <i>b</i>
Burke et al [15] 2001	Multi-criteria approach dealing with 9 criteria based on initial solutions by Saturation Degree	Nottingham <i>b</i>

Πίνακας 28. Χαρακτηριστικά των συνόλων δεδομένων αναφοράς της Μελβούρνης.

	Exams	Timeslots	Students	Enrolls	Objective
<i>I</i>	521	28	20656	62248	minimise adjacent exams on the same day or overnight
<i>II</i>	562	31	19816	60637	same as above

3.4. Αποτελέσματα των Προβλημάτων Αναφοράς

Όπως αναφέρθηκε ανωτέρω, έχει υπάρξει ένας μεγάλος αριθμός δημοσιευμένων εγγράφων που έχουν εργαστεί με τα σύνολα δεδομένων που συζητήθηκαν παραπάνω. Επιπλέον, οι δυσκολίες γύρω από την δημοσίευση των συνόλων δεδομένων του Τορόντο έχει οδηγήσει σε κάποια σύγχυση στο ποιες μέθοδοι αντιστοιχούν σε κάθε πρόβλημα. Οι Πίνακες 29-31 προσπαθούν να το διευκρινίσουν αυτό. Περιλαμβάνουν όλες τις μεθόδους που έχουν ασχοληθεί με τα προβλήματα του Τορόντο και προσπαθούν να απεικονίσουν ποιες μέθοδοι χρησιμοποιούνται, σε ποια προβλήματα.

Πίνακας 29. Αποτελέσματα στην βιβλιογραφία για τις δύο εκδόσεις του συνόλου δεδομένων *b* του Τορόντο (βλέπε Πίνακα 24). Οι τιμές με πλάγια γράμματα αντιπροσωπεύουν ότι είμαστε σίγουροι για την ακρίβεια από την άποψη των εκδόσεων των χρησιμοποιούμενων συνόλων δεδομένων. Οι τιμές με έντονους χαρακτήρες αντιπροσωπεύουν τα καλύτερα αποτελέσματα που αναφέρθηκαν. Τα "-" αντιπροσωπεύουν το αντίστοιχο πρόβλημα δεν ελέγχεται ή μια εφικτή λύση δεν μπορεί να αποκτηθεί.

data set version I/II	Carter et al. (1996) [51] I	Caramia et al. (2001) [45] I	Di & Schaerf (2001) [71] I	Di Gaspero (2002) [70] I	Paquete & Stutzle (2002) [113] I	Burke & Newall (2003) [36] I	Casey & Thompson (2003) [52] II	Merlot et al. (2003) [104] I
car91	7.1	6.6	6.2	5.7	-	4.65	5.4	5.1
car92	6.2	6.0	5.2	-	-	4.1	4.4	4.3
ear83	36.4	29.3	45.7	39.4	38.9	37.05	34.8	35.1
hec92	10.8	9.2	12.4	10.9	11.2	11.54	10.8	10.6
kfu93	14.0	13.8	18.0	-	16.5	13.9	14.1	13.5
lse91	10.5	9.6	15.5	12.6	13.2	10.82	14.7	10.5
rye92	7.3	6.8	-	-	-	-	-	-
sta83	161.5	158.2	160.8	157.4	168.3	168.73	134.9	157.3
tre92	9.6	9.4	10.0	-	9.3	8.35	8.7	8.4
uta92	3.5	3.5	4.2	4.1	-	3.2	-	3.5
ute92	25.8	24.4	27.8	-	29.0	25.83	25.4	25.1
yor83	41.7	36.2	41.0	39.7	38.9	37.28	37.5	37.4

Πίνακας 29. (συνέχεια) Αποτελέσματα στη βιβλιογραφία σχετικά με τις παραλλαγές του συνόλου δεδομένων *b* του Τορόντο.

data set version I/II	Burke & Newall (2004) [37] I	Burke et al. (2004) [16] I	Asmuni et al (2005) [8] I	Cote et al (2005) [61] I	Kendall & Hissan (2005) [90] I	Yang & Petrovic (2005) [149] I	Abdullah et al (2006) [3] I	Burke et al (2006) [34] I	Burke et al (2006) [22] I
car91	5.0	4.8	5.29	5.4	5.37	4.5	5.2	5.36	4.6
car92	4.3	4.2	4.56	4.2	4.67	3.93	4.4	4.53	4.0
ear83	36.2	35.4	37.02	34.2	40.18	33.7	34.9	37.92	32.8
hec92	11.6	10.8	11.78	10.4	11.86	10.83	10.3	12.25	10.0
kfu93	15.0	13.7	15.81	14.3	15.84	13.82	13.5	15.2	13.0
lse91	11.0	10.4	12.09	11.3	-	10.35	10.2	11.33	10.0
rye92	-	8.9	10.35	8.8	-	8.53	8.7	-	-
sta83	161.9	159.1	160.42	157.0	157.38	158.35*	159.2	158.19	159.9
tre92	8.4	8.3	8.67	8.6	8.39	7.92	8.4	8.92	7.9
uta92	3.4	3.4	3.57	3.5	-	3.14	3.6	3.88	3.2
ute92	27.4	25.7	27.78	25.3	27.6	25.39	26.0	28.01	24.8
yor83	40.8	36.7	40.66	36.4	-	36.35	36.2	41.37	37.28

Πίνακας 30. Αποτελέσματα στην βιβλιογραφία για το σύνολο δεδομένων *I* του Τορόντο, οι παραλλαγές *a* (αριστερή στήλη) και *e* (δεξιά στήλη).

data set	Carter et al. [51] (1996)	Caramia et al. [45] (2001)	Merlot et al. [104] (2003)	Terashima-Marin et al. [138] (1999)	Ross et al. [128] (2004)
car91	28	28	30	130	283
car92	28	28	31	285	542
ear83	22	22	24	723	958
hec92	17	17	18	154	224
kfu93	19	19	21	223	226
lse91	17	17	18	221	263
rye92	21	21	22	671	832
pur93	35	36	-	-	-
sta83	13	13	13	821	1058
tre92	20	20	21	586	604
uta92	32	30	32	594	855
ute92	10	10	11	902	967
yor83	19	19	23	708	758

Πίνακας 31. Αποτελέσματα στην βιβλιογραφία για το σύνολο δεδομένων *I* του Τορόντο, οι παραλλαγές *c* (άνω μέρος) και *d* (κάτω μέρος).

dataset	car92	car91	kfu93	tre92	uta92	pur93
Burke et al (1996) [38]	81	331	974	3	772	-
Caramia et al (2001) [45]	74	268	912	2	680	-
Di Gaspero&Schearf (2001) [71]	88	424	512	4	554	-
Merlot et al (2003) [104]	31	158	247	0	334	-
Abdullah et al (2006) [3]	37	278	548	0	300	-
Abdullah et al (2006) [4]	47	525	206	4	310	-
Burke et al (1998) [40]	2218	-	3256	-	2440	-
Burke&Newall (1999) [35]	1665	-	1388	-	-	63824
Di Gaspero&Schearf (2001) [71]	3048	-	1733	-	-	123935
Merlot et al (2003) [104]	1744	-	1082	-	-	-
Burke&Newall (2004) [37]	1775	-	1422	-	-	97237
Burke et al (2004) [16]	1506	-	1321	-	-	-

Ο Πίνακας 32 παρουσιάζει τα αποτελέσματα από τις διαφορετικές προσεγγίσεις που εφαρμόζονται στα σύνολα δεδομένων *a* και *b* του Νότινχαμ (βλ. Πίνακα 26) και στα σύνολα δεδομένων *I* και *II* της Μελβούρνης (βλ. Πίνακας 26) στην βιβλιογραφία.

Πίνακας 32. Αποτελέσματα για τα σύνολα δεδομένων *a* και *b* του Νότινχαμ και τα σύνολα δεδομένων *I* και *II* της Μελβούρνης. Οι τιμές με έντονους χαρακτήρες αντιπροσωπεύουν τα καλύτερα αποτελέσματα που αναφέρθηκαν. Τα "-" αντιπροσωπεύουν το αντίστοιχο πρόβλημα που δεν έχει ελεγχθεί ή μία εφικτή λύση δεν μπορεί να ληφθεί.

Nottingham	Burke et al [38] (1996)	Di Gaspero & Schearf [71] (2001)	Caramia et al [45] (2001)	Merlot et al [104] (2003)	Abdullah et al [4] (2006)
<i>a</i> (26 slots)	53	11	44	2	18
<i>a</i> (23 slots)	269	123	-	88	-
	Burke & Newall [35] (1999)	Di Gaspero & Schearf [71] (2001)	Merlot et al [104] (2003)	Burke & Newall [37] (2004)	Burke et al [16] (2004)
<i>b</i>	519	751	401	545	384
Melbounre	Merlot et al [104] (2003)				
<i>I</i>	1072				
<i>II</i>	1115				

Οι Πίνακες 29-32 απεικονίζουν επίσης ποια από τις μεθόδους είναι πιο αποτελεσματική από την άποψη της ποιότητας λύσης. Τα πολύ καλύτερα αποτελέσματα παρουσιάζονται με έντονους χαρακτήρες. Δεν έχουμε παραθέσει υπολογιστικούς χρόνους για τους λόγους αυτούς. Πρώτον, πολλά από αυτά τα έγγραφα δεν αναφέρουν τους σχετικούς χρόνους. Δεύτερον, οι συγκρίσεις σε πολύ διαφορετικές πλατφόρμες είναι αδύνατες. Τρίτον, ο χρονοπρογραμματισμός εξετάσεων είναι ένα πρόβλημα, το οποίο είναι σχεδόν πάντα να αντιμετωπιστεί εβδομάδες ή μήνες πριν από το χρονοδιάγραμμα που θα χρησιμοποιηθεί. Έτσι, σίγουρα δεν είναι ένα πρόβλημα χρόνου και υπάρχουν πολλά πραγματικά σενάρια όπου θα ήταν απολύτως λογικό να αφήσει έναν αλγόριθμο να τρέχει τη διάρκεια της νύχτας ή ακόμα και πάνω από ένα Σαββατοκύριακο.

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Η έρευνα χρονοπρογραμματισμού ξεκίνησε με απλές διαδοχικές τεχνικές στη δεκαετία του 1960. Τεχνικές με βάση περιορισμούς εμφανίστηκαν αργότερα και εξακολουθούν να διαδραματίζουν σημαντικό ρόλο στον χρονοπρογραμματισμό σήμερα. Οι πρόσφατες έρευνες σε χρονοπρογραμματισμό εξετάσεων κυριαρχούνται από μετά-ευρετικά και τις εντάξεις / υβριδοποιήσεις τους με μία ποικιλία τεχνικών, συμπεριλαμβανομένων πολλών από τις πρώιμες τεχνικές. Οι τεχνικές με βάση την τοπική αναζήτηση, πολυκριτηριακές τεχνικές και προσεγγίσεις που έχουν ως στόχο να είναι πιο γενικές από τις πιο καινούριες έχουν παρουσιάσει επίσης ενδιαφέροντα αποτελέσματα.

Οι πρόσφατες καινοτομίες έχουν χρησιμοποιήσει διαφορετικούς μηχανισμούς στον χρονοπρογραμματισμό εξετάσεων και καλύπτουν μια ποικιλία νέων τεχνικών, συμπεριλαμβανομένων της Μεταβλητής Αναζήτησης Γειτονιάς, την Επαναληπτική Τοπική Αναζήτηση, GRASP, και υπέρ-ευρετικά με σκοπό την ανάπτυξη πιο ισχυρών, αποτελεσματικών, αποδοτικών και πιο γενικών προσεγγίσεις.

Στην επόμενη ενότητα θα αξιοποιήσουμε την παραπάνω συζήτηση για να επισημάνουμε μια σειρά από συμπεράσματα και να υποβάλουμε κάποιες ιδέες για μελλοντική έρευνα που παράγεται από αυτά τα συμπεράσματα.

Τα Μετά-ευρετικά έχουν προσελκύσει τη μεγαλύτερη προσοχή στην έρευνα του χρονοπρογραμματισμού εξετάσεων.

Εκτός από μια ολοκληρωμένη αντιμετώπιση της Tabu αναζήτησης, της Προσομοιωμένης Ανόπτησης, των Γενετικών Αλγόριθμων και διάφορων υβριδίων, κάποιες νέες τεχνικές χρονοπρογραμματισμού εξετάσεων έχουν παρουσιαστεί. Για παράδειγμα, οι GRASP και η επαναλαμβανόμενη Τοπική Αναζήτηση φτιάχτηκαν με παρόμοια ιδέα της εξερεύνησης ευρύτερων περιοχών του χώρου αναζήτησης με τη χρήση μίας άπληστης τεχνική αναζήτησης

για να μειώσει τον κίνδυνο να κολλήσει σε τοπικό βέλτιστο. Η Μεταβλητή Αναζήτηση Γειτονιάς ξεφεύγει από τα τοπικά βέλτιστα με την αλλαγή μεταξύ των χώρων αναζήτησης που ορίζονται από διαφορετικές δομές γειτονιάς. Η έρευνα Μεγάλης γειτονιάς πληροί αυτό με την επέκταση της ευελιξίας των κινήσεων εντός του χώρου αναζήτησης. Εν περιλήψει, οι τεχνικές αυτές επεκτείνουν την ιδέα της διευκόλυνσης της αναζήτησης για να διαφύγει από τα τοπικά βέλτιστα με μια ποικιλία τρόπων και έχουν αποκτήσει πολλά υποσχόμενη πρόοδο σε ένα ευρύ φάσμα των προβλημάτων χρονοπρογραμματισμού εξετάσεων. Η ανάπτυξη αυτών των νέων τεχνικών έχει ανοίξει μια μεγάλη ποικιλία από νέες κατευθύνσεις έρευνας, όπως η διερεύνηση εναλλακτικών δομών γειτονιάς, νέες multi-start τεχνικές, τα θέματα υβριδισμού, εναλλακτικούς παρόχους και πολλά άλλα. Ένας από τους βασικούς στόχους της έρευνας είναι να παρέχει μια κατάλληλη ισορροπία μεταξύ της εξερεύνησης και της εκμετάλλευση σε αλγόριθμους αναζήτησης.

Εκτεταμένη μελέτη απαιτείται επίσης για το πώς να καθορίσει τις κατάλληλες ρυθμίσεις παραμέτρων για τις μετά-ευρετικές μεθόδους. Ο προσδιορισμός της κατάλληλης αρχικοποίησης μεθόδων και σε βάθος ανάλυση των συνεπειών της εκκίνησης σε μια σειρά των μετά-ευρετικών είναι ένα άλλο σημαντικό θέμα της έρευνας χρονοπρογραμματισμού εξετάσεων. Θεωρητικά ζητήματα (όπως η μεταβατική φάση) και τεχνικές πολλαπλών κριτηρίων αντιπροσωπεύουν άλλες σημαντικές κατευθύνσεις στην έρευνα των μετά-ευρετικών. Εξελικτικές μέθοδοι και άλλες τεχνικές βασισμένες στον πληθυσμό αντιπροσωπεύουν ένα σημαντικό ποσοστό της μετά-ευρετικής βιβλιογραφίας σχετικά με χρονοπρογραμματισμό εξετάσεων. Υπάρχουν πολλές κατευθύνσεις έρευνας που δημιουργούνται με την εξέταση του υβριδισμού των μετά-ευρετικών μεθόδων ιδιαίτερα μεταξύ μεθόδους με βάση τον πληθυσμό και άλλες προσεγγίσεις. Μια μελέτη της κωδικοποίησης θεμάτων αντιπροσωπεύει μια νέα και πολλά υποσχόμενη κατεύθυνση και στους εξελικτικούς αλγόριθμους και στην υπέρ-ευρετική έρευνα.

Περισσότεροι Γενικές και Προσαρμοστικές τεχνικές έχουν επίσης μελετηθεί.

Τα Υπέρ-ευρετικά που ασχολούνται με την αναζήτηση των κατάλληλων ευρετικών περισσότερο από τις λεπτομέρειες του προβλήματος, για τις πραγματικές λύσεις, οι οποίες ήταν το επίκεντρο των παραδοσιακών αλγορίθμων αναζήτησης. Αυτό ανοίγει μια νέα κατεύθυνση της έρευνας και αντιπροσωπεύει πολλές δυνατότητες και στις δύο πρακτικές εφαρμογές και θεωρητική μελέτη. Προσαρμοστικές τεχνικές έχουν επίσης πρόσφατα εμφανιστεί όπου οι πληροφορίες που συλλέγονται κατά τη διάρκεια της επίλυσης του προβλήματος χρησιμοποιούνται για να καθοδηγήσουν την αναζήτηση. Κάποια εργασία έχει πραγματοποιηθεί στις τεχνικές με βάση τη γνώση όπου η εμπειρία από τις προηγούμενες λύσεις προβλημάτων οδηγεί την αναζήτηση. Περαιτέρω έρευνα των τεχνικές με βάση τη γνώση έχει την υπόσχεση για την υποστήριξη της ανάπτυξης των πιο ουσιαστικά γενικών προσεγγίσεων. Ο στόχος είναι να ασχοληθεί αυτόματα με διάφορα προβλήματα με δυναμικό τρόπο, έτσι ώστε να μην καταβληθούν επιπλέον προσπάθειες για να τελειοποιήσουν την προσέγγιση. Η ανάλυση των ευρετικών / τεχνικών σχετικά με τη φύση του χώρου αναζήτησης θα μπορούσε να είναι επωφελής. Είναι γενικά αποδεκτό ότι λίγα είναι γνωστά σχετικά με τη φύση των χώρων αναζήτησης, ειδικά για τα σύνθετα προβλήματα του πραγματικού κόσμου, όπως χρονοπρογραμματισμός εξεταστικής.

Οι υβριδισμοί των διαφορετικών τεχνικών έχουν διερευνηθεί στην κατάρτιση προγράμματος εξεταστικής.

Μολονότι διάφοροι συγγραφείς έχουν ευνοήσει διαφορετικές προσεγγίσεις, έχει παρατηρηθεί ότι υβριδικές προσεγγίσεις είναι συνήθως ανώτερες των καθαρών αλγορίθμων. Για παράδειγμα, όλες οι πρόσφατες εργασίες για τις τεχνικές με βάση τους περιορισμούς

αντιπροσωπεύουν υβριδισμό με άλλες τεχνικές. Ωστόσο, στις περισσότερες των περιπτώσεων, μεθοδολογίες υβριδοποιούνται σε ένα διαδοχικό τρόπο, αντί να ενσωματωθούν αποτελεσματικά. Περισσότερη δουλειά πρέπει να γίνει όχι απλά για να συνδυαστούν αλλά να ενσωματωθούν διαφορετικές μεθοδολογίες αποτελεσματικά. Για παράδειγμα, στους μιμητικούς αλγόριθμους, τοπική αναζήτηση χρησιμοποιείται συνεργατικά μετά από κάθε γενιά. Σε ένα υπέρ-ευρετικό, μια προσέγγιση που ακολουθήθηκε είναι ότι τα χαμηλού επίπεδου ευρετικά αναζητούνται και συνδυάζονται προσαρμοστικά κατά τη διάρκεια της επίλυσης του προβλήματος. Περαιτέρω σε βάθος ανάλυση και έρευνα μπορεί να υποστηρίξει το σχεδιασμό και την ανάπτυξη πιο ισχυρών τεχνικών.

Τα προβλήματα αναφοράς χρονοπρογραμματισμού εξετάσεων έχουν σχηματιστεί πλήρως και δοκιμαστεί από μια σειρά προσεγγίσεων.

Πρόσφατες state-of-the-art προσεγγίσεις στην έρευνα χρονοπρογραμματισμού εξετάσεων έχουν διεξάγει συγκρίσεις των προβλημάτων συγκριτικής μέτρησης επιδόσεων, που έχουν εμφανιστεί κατά τη διάρκεια των τελευταίων δέκα ετών. Αυτό έχει οδηγήσει σε θεμελιώδεις εξελίξεις στην έρευνα χρονοπρογραμματισμού εξετάσεων. Ωστόσο, τα προβλήματα εξακολουθούν να παριστάνουν απλοποιημένες εκδόσεις του προβλήματος. Στο ευρύτερο πλαίσιο του προγραμματισμού της έρευνας, υπάρχει έχει πολύ πρόσφατη συζήτηση για το "χάσμα μεταξύ θεωρίας και πράξης». Το ίδιο ισχύει και για την έρευνα χρονοπρογραμματισμού εξετάσεων. Μια σημαντική ερευνητική κατεύθυνση αντιπροσωπεύεται από την εξερεύνηση μεγάλου φάσματος των θεμάτων της έρευνας που ανοίγουν εξετάζοντας τα υψηλά επίπεδα πολυπλοκότητας που δημιουργούνται από τον πραγματικό κόσμο προβλημάτων. Επιπλέον, ακόμα δεν υπάρχει μία ευρέως αποδεκτή καθολική μορφή δεδομένων και τυπικές γλώσσες χρονοπρογραμματισμού. Η καθιέρωση των μέτρων ποιότητας με πρότυπες τεχνικές τόσο στην ποιότητα λύσης (αντικειμενικές λειτουργίες) και του υπολογιστικού χρόνου για τα προβλήματα χρονοπρογραμματισμού

εξετάσεων απαιτεί πολύ δουλειά και είναι ζωτικής σημασίας σε συνδυασμό με το σχηματισμό των σημείων αναφοράς. Οι απαιτήσεις για την ανάπτυξη των αυτόματων εργαλείων για την υποστήριξη του χρονοπρογραμματισμού για να εξοικονομήσουν σημαντικό χρόνο ανάπτυξης, εξακολουθούν να υφίστανται. Για να ενθαρρυνθεί η ανάπτυξη, χτίζουμε ένα αρχείο όπου συλλέγονται προβλήματα αναφοράς χρονοπρογραμματισμού εξετάσεων, μαζί με μία ενημερωμένη κατηγοριοποιημένη βιβλιογραφία χρονοδιαγράμματος (μετά το 1995).

Εν ολίγοις, είναι δυνατό να εξαχθούν ορισμένα συμπεράσματα από μια σε βάθος επισκόπηση της βιβλιογραφίας χρονοπρογραμματισμού εξετάσεων κατά τα τελευταία δέκα χρόνια. Πρώτον, υπήρξε ένας σημαντικός αριθμός των επιτυχιών της έρευνας σε αυτό το διάστημα. Δεύτερον, η τρέχουσα τεχνογνωσία παρέχει μια ισχυρή πλατφόρμα από μια σειρά σημαντικών ερευνητικών κατευθύνσεων. Τρίτον, η μελλοντική έρευνα απαιτεί ιδιαίτερη έμφαση στην πολυπλοκότητα των προβλημάτων πραγματικών κόσμου και αυτό απαιτεί τη δημιουργία περισσότερων σημείων αναφοράς που προέρχονται από προβλήματα του πραγματικού κόσμου. Τέταρτον, η αύξηση του επιπέδου της γενικότητας των συστημάτων υποστήριξης αποφάσεων (συμπεριλαμβανομένων για χρονοπρογραμματισμό εξετάσεων) αντιπροσωπεύει ένα αναδυόμενο θέμα. Τέλος, αξίζει να σημειωθεί ότι επιτυχής εργασίες στον χρονοπρογραμματισμό εξετάσεων έχουν συγγραφεί από τους ερευνητές από μια σειρά πειθαρχικού υπόβαθρου και ιδιαίτερα στο περιβάλλον της Επιχειρησιακής Έρευνας και Τεχνητής Νοημοσύνης. Η διεπιστημονική συνεργασία είναι ζωτικής σημασίας για την επιστημονική πρόοδο στον τομέα. Είναι σαφές από την ανάλυση αυτή της βιβλιογραφίας ότι το μέλλον της έρευνας χρονοπρογραμματισμό εξετάσεων είναι διεπιστημονικό.

Βιβλιογραφία

1. An Investigation of a Genetic Algorithm and Sequential Local Search Approach for Curriculum-based Course Timetabling Problems (Salwani Abdullah , Hamza Turabieh , Barry McCollum, Edmund K Burke)
2. Applying the threshold accepting metaheuristic to curriculum based course timetabling - A contribution to the second International timetabling competition ITC 2007 (Martin Josef Geiger)
3. Benchmarking Curriculum-Based Course Timetabling: Formulations, Data Formats, Instances, Validation, and Results (Fabio De Cesco, Luca Di Gaspero, Andrea Schaerf)
4. Multi-criteria Curriculum-Based Course Timetabling – A Comparison of a Weighted Sum and a Reference Point Based Approach (Martin Josef Geiger)
5. Solving the Course Timetabling Problem with a Hybrid Heuristic Algorithm (Zhipeng Lü and Jin-Kao Hao)
6. Curriculum based course timetabling: new solutions to Udine benchmark instances (Gerald Lach, Marco E. Lübbecke)
7. A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling (Sadaf Naseem Jat, Shengxiang Yang)

8. A Modular Multiphase Heuristic Solver for Post Enrolment Course Timetabling (Marco Chiarandini, Chris Fawcett, and Holger H. Hoos)
9. A time-dependent metaheuristic algorithm for post enrolment-based course timetabling (Rhyd Lewis)
10. An Elitist-Ant System for Solving the Post-Enrolment Course Timetabling Problem (Ghaith M. Jaradat and Masri Ayob)
11. Solving the post enrolment course timetabling problem by ant colony optimization (Clemens Nothegger, Alfred Mayer, Andreas Chwatal, Günther R. Raidl)
12. Post Enrolment based Course Timetabling: A Description of the Problem Model used for track Two of the Second International Timetabling Competition (Rhydian Lewis, Ben Paechter, Barry McCollum)
13. Local search and constraint programming for the post enrolment-based course timetabling problem (Hadrien Cambazard, Emmanuel Hebrard, Barry O'Sullivan, Alexandre Papadopoulos)
14. A survey of search methodologies and automated system development for examination timetabling (R. Qu, E.K. Burke, B. McCollum, L.T.G. Merlot, S.Y. Lee)
15. An integrated hybrid approach to the examination timetabling problem (Hamza Turabieh, Salwani Abdullah)
16. A Memetic Algorithm for University Exam Timetabling (E.K. Burke, J.P. Newall and R.F. Weare)
17. Scatter search technique for exam timetabling (Nashat Mansour, Vatche Isahakian, Iman Ghalayini)

18. Roulette Wheel Graph Colouring for Solving Examination Timetabling Problems
(Nasser R. Sabar, Masri Ayob, Graham Kendall, and Rong Qu)
19. ITC2007 solver description: a hybrid approach (Tomáš Müller)
20. University Timetabling (Sanja Petrovic and Edmund Burke)
21. Practice and Theory of Automated Timetabling VI (Edmund K. Burke, Hana Rudová)
22. A Survey of Automated Timetabling (A. SCHAERF)
23. A New Model for Automated Examination Timetabling (Barry McCollum, Paul McMullan)