



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

**ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΙΠΤΑΜΕΝΩΝ ΣΕ ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΣΥΝΕΔΡΙΕΣ ΥΠΟ
ΠΕΡΙΟΡΙΣΜΟΥΣ ΔΥΝΑΜΙΚΟΤΗΤΑΣ, ΕΙΣΟΡΡΟΠΗΣΗΣ, ΠΡΟΤΙΜΗΣΕΩΝ
ΚΑΙ ΛΗΞΗΣ ΠΤΗΤΙΚΗΣ ΙΚΑΝΟΤΗΤΑΣ**

υπό

ΓΡΙΒΑΚΗ ΜΑΡΙΑΝΝΑ ΚΑΙ ΡΑΠΟΤΙΚΑ ΑΛΙΚΗ

Επιβλέπων Καθηγητής: Κοζανίδης Γεώργιος

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Διπλώματος Μηχανολόγου Μηχανικού

2019

© 2019 ΓΡΙΒΑΚΗ ΜΑΡΙΑΝΝΑ, ΡΑΠΟΤΙΚΑ ΑΛΙΚΗ

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

**Πρώτος Εξεταστής
(Επιβλέπων)**

Δρ. Γεώργιος Κοζανίδης
Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής

Δρ. Γεώργιος Λυμπερόπουλος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο
Θεσσαλίας

Τρίτος Εξεταστής

Δρ. Δημήτριος Παντελής
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο
Θεσσαλίας

ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτα από όλα θα θέλαμε να ευχαριστήσουμε θερμά όλους εκείνους που με έμμεσο ή άμεσο τρόπο μας βοήθησαν στην εκπόνηση της διπλωματικής μας εργασίας.

Πιο συγκεκριμένα ευχαριστούμε θερμά τον επιβλέποντα καθηγητή κ. Γεώργιο Κοζανίδη, Αναπληρωτή Καθηγητή του τμήματος Μηχανολόγων Μηχανικών της Πολυτεχνικής σχολής του Πανεπιστημίου Θεσσαλίας, που μας έδωσε την ευκαιρία να ασχοληθούμε με τον κλάδο των αερομεταφορών αλλά και με τα προβλήματα ανάθεσης του ανθρώπινου δυναμικού. Επίσης οι συμβουλές του, η τεχνογνωσία του αλλά και η καθοδήγηση του στα θέματα προγραμματισμού μας ήταν απαραίτητα εργαλεία για την υλοποίηση της διπλωματικής μας εργασίας.

Στην συνέχεια θα θέλαμε να ευχαριστήσουμε τα υπόλοιπα δυο μέλη της εξεταστικής επιτροπής της διπλωματικής εργασίας μας, κ. Γεώργιο Λυμπερόπουλο, Καθηγητή του Τμήματος Μηχανολόγων Μηχανικών και κ. Δημήτριο Παντελή Αναπληρωτή Καθηγητή του ίδιου τμήματος για τον χρόνο που αφιέρωσαν για την ανάγνωση της εργασίας. Επίσης επιθυμούμε να ευχαριστήσουμε θερμά και τον κ. Ιωάννη Λυχναρόπουλο Δρ Μαθηματικό του τμήματος Μηχανολόγων Μηχανικών για την πολύτιμη βοήθειά του και την παροχή του απαραίτητου λογισμικού για την περάτωση της εργασίας μας.

Τέλος, θα θέλαμε να ευχαριστήσουμε τις οικογένειες μας για την αγάπη και την κατανόηση που μας έχουν δείξει αλλά και για την υλική και ηθική συμπαράσταση τους κατά τη διάρκεια των σπουδών μας.

ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΙΠΤΑΜΕΝΩΝ ΣΕ ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΣΥΝΕΔΡΙΕΣ ΥΠΟ ΠΕΡΙΟΡΙΣΜΟΥΣ ΔΥΝΑΜΙΚΟΤΗΤΑΣ, ΕΞΙΣΟΡΡΟΠΗΣΗΣ, ΠΡΟΤΙΜΗΣΕΩΝ ΚΑΙ ΛΗΞΗΣ ΠΤΗΤΙΚΗΣ ΙΚΑΝΟΤΗΤΑΣ.

ΓΡΙΒΑΚΗ ΜΑΡΙΑΝΝΑ και ΡΑΠΟΤΙΚΑ ΑΛΙΚΗ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών

Επιβλέπων Καθηγητής Δρ. Γεώργιος Κοζανίδης

Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο
Θεσσαλίας

ΠΕΡΙΛΗΨΗ

Η αεροπορική βιομηχανία είναι μια από τις πιο ανταγωνιστικές βιομηχανίες που υπάρχουν στις μέρες μας, πράγμα που αναγκάζει τις αεροπορικές εταιρείες συνεχώς να εξελίσσονται. Η αποτελεσματική λειτουργία του πληρώματος αποτελεί ένα κρίσιμο σημείο για κάθε αεροπορική εταιρεία. Παρουσιάζει λοιπόν μεγάλο ενδιαφέρον ο σωστός προγραμματισμός της εκπαίδευσης του ανθρώπινου δυναμικού, ο οποίος είναι αναγκαίος για την ασφάλεια των αερομεταφορών.

Στην εργασία μας παρουσιάζουμε ένα μαθηματικό μοντέλο που στηρίζεται στις αρχές του Μεικτού Ακέραιου Προγραμματισμού (MILP). Το μοντέλο αυτό δημιουργήθηκε για να βελτιστοποιήσει την ανάθεση ιπτάμενων σε εκπαιδευτικές συνεδρίες που αντιστοιχούν σε συνεδρίες προσομοιωτή ή σε τμήματα θεωρίας. Στο μοντέλο θεωρούμε ως δεδομένα εισόδου τα χαρακτηριστικά του κάθε ιπτάμενου και της κάθε συνεδρίας και λαμβάνουμε υπόψη ορισμένους περιορισμούς. Αυτοί οι περιορισμοί αναφέρονται στην δυναμικότητα, στην ισορροπία που θα πρέπει να υπάρχει μεταξύ των τμημάτων, στην ικανοποίηση των προτιμήσεων του κάθε ιπτάμενου με σειρά αυστηρής αρχαιότητας αλλά και στην λήξη της πτητικής ικανότητας του καθενός. Έτσι καταστρώνοντας τον κατάλληλο κώδικα για το μαθηματικό μας μοντέλο, στόχος μας είναι να ελαχιστοποιήσουμε τους ιπτάμενους που δεν θα ανατεθούν σε καμία εκπαιδευτική συνεδρία.

Λέξεις κλειδιά: ιπτάμενοι, εκπαιδευτικές συνεδρίες, βελτιστοποίηση, πρόβλημα ανάθεσης, προτιμήσεις, αρχαιότητα, λήξη πτητικής ικανότητας

OPTIMAL ASSIGNMENT OF FLIGHT CREW TO TRAINING SESSIONS UNDER RESTRICTIONS OF DYNAMIC CAPACITY, BALANCE, PREFERENCES AND ENDING OF FLYING CAPACITY

GRIVAKH MARIANNA AND RAPOTIKA ALIKH

University of Thessaly, Department of Mechanical Engineering

Supervisor: Dr. George Kozanidis

Associate Professor, Department of Mechanical Engineering, University of Thessaly

ABSTRACT

Nowadays, airline industry is one of the most competitive industries that exists, which is forcing them to go along constantly. The efficient operation of the crew is a crucial point for each airline industry. There is therefore a great deal of interest for properly human resources training planning, which is essential for safety of air transport.

In our project, we present a mathematical model based on the principles of Mixed Integer Linear Programming (MILP). This model was created to optimize the assignment of the crew to training sessions, corresponding to simulator or theory sessions. Moreover, we consider the characteristics of each member of the crew and session as input data and receive some restrictions. These limitations refer to the capacity, the balance that must be achieved between the sections, the satisfaction of each trainee preferences in seniority and the ending of everyone's flying ability. Thus, designing the appropriate code for our mathematical model, our aim is to minimize the percentage of flight crew that will not be assigned to any training session.

Key words: flight crew, training sessions, optimization, assignment problem, preferences, seniority, ending of flight capacity.

Περιεχόμενα

1. ΕΙΣΑΓΩΓΗ.....	1
1.1 Θεωρητικό υπόβαθρο	1
1.2 Σκοπός Εργασίας	3
2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ.....	4
3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	6
3.1 Περιγραφή δεδομένων	7
3.2 Περιγραφή της λύσης του προβλήματος	10
4. ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ	12
4.1 Δεδομένα εισόδου	12
4.2 Μεταβλητές απόφασης	13
4.2.1 Δυαδικές μεταβλητές απόφασης	13
4.2.2 Ακέραιες μεταβλητές απόφασης	13
4.3 Μαθηματικό Μοντέλο	13
4.3.1 Αντικειμενική συνάρτηση	14
4.3.2 Περιορισμοί	14
4.3.3 Προσδιορισμός συντελεστών κόστους των μεταβλητών απόφασης	18
5. ΤΑ ΣΤΑΔΙΑ ΤΗΣ ΕΠΙΛΥΣΗΣ.....	22
6. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ C++.....	25
7. ΕΦΑΡΜΟΓΗ ΤΟΥ ΚΩΔΙΚΑ (ΠΕΙΡΑΜΑΤΑ)	42
7.1 Πρώτο πείραμα	42
7.2 Δεύτερο πείραμα.....	51
7.3 Τρίτο πείραμα.....	58
8. ΕΠΙΛΟΓΟΣ.....	64
8.1 Σύνοψη και αποτελέσματα	64
8.2 Μελλοντικές επεκτάσεις.....	64
9. ΒΙΒΛΙΟΓΡΑΦΙΑ	66

ΕΙΚΟΝΕΣ

εικόνα 3 1: Σχηματική απεικόνιση της λίστας των ιπτάμενων	8
εικόνα 3 2: Σχηματική απεικόνιση της λίστας των εκπαιδευτικών συνεδριών	9
Εικόνα 4 1:Σχηματική απεικόνιση της Penaltylist.....	21
Εικόνα 5 1:Τα στάδια της επίλυσης	24
Εικόνα 6 2:Ο κώδικας για το seniority	26
Εικόνα 6 1: Ο κώδικας για το Expiration	26
Εικόνα 6 3: Ο κώδικας για το position	27
Εικόνα 6 4: Ο κώδικας για την λίστα των εφικτών συνεδριών	27
Εικόνα 6 5:Ο κώδικας για την λίστα των προτιμήσεων.....	28
Εικόνα 6 6: Η ρουτίνα unit_problem()	29
Εικόνα 6 7:Δημιουργία μεταβλητών απόφασης	30
Εικόνα 6 8:Δημιουργία περιορισμών ανάθεσης	31
Εικόνα 6 9: Ο κώδικας για το sim_capacity	32
Εικόνα 6 10: Total capacity	33
Εικόνα 6 11:Ο κώδικας για cockpitcapacity, είναι ίδιος και για cabin με position =3 ή 4.....	34
Εικόνα 6 12: Ο κώδικας για cp,fo,purser,facapacity . Είναι ο ίδιος και χρησιμοποιούμε διαφορετικό position	34
Εικόνα 6 13: Ο κώδικας για το fix_expiration.....	36
Εικόνα 6 14: Ο κώδικας για το fix_bids	37
Εικόνα 6 15: Ο κώδικας για το class_balance.....	39
Εικόνα 6 16: Κ επίλυση και τα αποτελέσματα	40
Εικόνα 6 17: Το κλείσιμο του προβλήματος.....	41

ΠΙΝΑΚΕΣ

Πίνακας 4 1: Σύνολα του μαθηματικού μοντέλου	12
Πίνακας 4 2: Δείκτες του μαθηματικού μοντέλου	12
Πίνακας 4 3: Συμβολισμοί των παραμέτρων του μαθηματικού μοντέλου	12
Πίνακας 7 1: Τα χαρακτηριστικά των ιπτάμενων παράδειγμα 1	42
Πίνακας 72:Λίστα των προτιμήσεων των ιπτάμενων και το σύνολο των εφικτών συνεδριών παράδειγμα 1.....	43
Πίνακας 7 3: Η χωρητικότητα για κάθε συνεδρία παράδειγμα 1	44
Πίνακας 7 4: Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 1	49
Πίνακας 7 5: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 1	50
Πίνακας 7 6:Χαρακτηριστικά ιπτάμενων παράδειγμα 2	51
Πίνακας 7 7:Λίστα με τις προτιμήσεις των ιπτάμενων και το σύνολο των εφικτών συνεδριών παράδειγμα 2.....	52
Πίνακας 7 8:Η χωρητικότητα για κάθε συνεδρία παράδειγμα 2	52
Πίνακας 7 9: Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 2	57
Πίνακας 7 10: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 2.....	58
Πίνακας 7 11: Χαρακτηριστικά ιπτάμενων παράδειγμα 3	59
Πίνακας 7 12:Λίστα προτιμήσεων και σύνολο εφικτών συνεδριών παράδειγμα 3 ..	59
Πίνακας 7 13:Χωρητικότητα για κάθε συνεδρία παράδειγμα 3	59
Πίνακας 7 14:Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 3.....	62
Πίνακας 7 15: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 3.....	63

1.ΕΙΣΑΓΩΓΗ

1.1 Θεωρητικό υπόβαθρο

Ορισμός Επιχειρησιακής Έρευνας

Η Επιχειρησιακή Έρευνα αποτελεί έναν επιστημονικό κλάδο των Εφαρμοσμένων Μαθηματικών, που χρησιμοποιεί μεθόδους στατιστικής ανάλυσης, μαθηματικών μοντέλων και μαθηματικής βελτιστοποίησης ώστε να προκύψουν βέλτιστες λύσεις σε σύνθετα προβλήματα λήψης αποφάσεων.(1) Αντικείμενό της είναι να καθορίσει τον καλύτερο σχεδιασμό και να συντονίσει ένα σύστημα υπό συνθήκες που απαιτούν την κατανομή πολλών παραγωγικών πόρων.

Ιστορική αναδρομή

Οι πρώτες δραστηριότητες της Επιχειρησιακής Έρευνας ξεκίνησαν στην Αγγλία κατά την διάρκεια του Β΄ Παγκοσμίου Πολέμου, όταν μια ομάδα Βρετανών επιστημόνων ξεκίνησαν την λήψη αποφάσεων όσον αφορά τη διαχείριση του πολεμικού υλικού στον καλύτερο δυνατό βαθμό. Ειδικότερα επιστήμονες και μηχανικοί ανέλυσαν διάφορα στρατιωτικού τύπου προβλήματα όπως για παράδειγμα, ανέπτυξαν αποτελεσματικές μεθόδους ώστε να χρησιμοποιήσουν ραντάρ τα οποία είχαν ανακαλυφθεί εκείνη την εποχή, διαχειρίστηκαν καλύτερα τις νηοπομπές και τα υποβρύχια, διαχειρίστηκαν τις επιθέσεις με βόμβες και γενικά βελτιστοποίησαν τις στρατιωτικές επιχειρήσεις. Μετά τον Β΄ Παγκόσμιο Πόλεμο αυτές οι ιδέες που αρχικά αφορούσαν μόνο τις στρατιωτικές δραστηριότητες οδήγησαν σε ένα νέο επιστημονικό πεδίο που αναπτύχθηκε ραγδαία κυρίως στις Ηνωμένες Πολιτείες της Αμερικής ενώ κατά την διάρκεια των δεκαετιών των 1950 και 1960 αναπτύχθηκαν οι περισσότεροι αλγόριθμοι και μέθοδοι που χρησιμοποιούνται ακόμα και σήμερα.(2)

Επιχειρησιακή Έρευνα και Βελτιστοποίηση

Όπως γίνεται αντιληπτό η Επιχειρησιακή Έρευνα είναι στενά συνδεδεμένη με την έννοια της Βελτιστοποίησης. Με τον όρο Βελτιστοποίηση αναφερόμαστε στην αναζήτηση βέλτιστων παραμέτρων ενός συνήθως περίπλοκου συστήματος. Στα μαθηματικά διατυπώνεται ένα πρόβλημα βελτιστοποίησης σαν πρόβλημα ελαχιστοποίησης ή μεγιστοποίησης μιας συνάρτησης μίας ή πολλών μεταβλητών.(3)

Μεθοδολογία

Η Επιχειρησιακή Έρευνα ακολουθεί μια ορισμένη διαδικασία για τον προσδιορισμό της βέλτιστης λύσης. (4)Τα βασικά στάδια για την βελτιστοποίηση ενός συστήματος είναι:

1. Ορισμός προβλήματος
2. Κατασκευή μοντέλου
3. Επίλυση μοντέλου
4. Επικύρωση μοντέλου
5. Εφαρμογή της λύσης

Διάφορες τεχνικές

Στην Επιχειρησιακή Έρευνα δεν διαθέτουμε κάποια απλή τεχνική για την επίλυση όλων των μαθηματικών μοντέλων. Αντίθετα ο τύπος και η πολυπλοκότητα του μαθηματικού μοντέλου δείχνουν την μέθοδο επίλυσης.

Ενδεικτικά αναφέρονται κάποιες από τις τεχνικές αυτές:

- Ο γραμμικός προγραμματισμός
- Ο μη-γραμμικός προγραμματισμός
- Ο ακέραιος προγραμματισμός
- Ο δυναμικός προγραμματισμός
- Ο μεικτός ακέραιος προγραμματισμός

Μεικτός ακέραιος προγραμματισμός

Ένα γραμμικό πρόβλημα μικτού-ακέραιου προγραμματισμού (MILP) είναι ένα μαθηματικό πρόγραμμα με γραμμικούς περιορισμούς στο οποίο απαιτούνται ορισμένες μεταβλητές για την ανάληψη ακέραιων τιμών. Οι ακέραιες μεταβλητές μπορεί να προέρχονται από τη φύση των προϊόντων. Τα μικτά ακέραια γραμμικά προβλήματα επιλύονται χρησιμοποιώντας την ίδια τεχνολογία με τα ακέραια (ή αντίστροφα). Για παράδειγμα, ένας αλγόριθμος branch-and-bound μπορεί να εκμεταλλευτεί την γραμμική χαλάρωση και η διαδικασία διακλάδωσης να εφαρμόζεται μόνο σε ακέραιες μεταβλητές. Παρόλο που τα MILP είναι δύσκολο να επιλυθούν γενικά, τα τελευταία δέκα χρόνια έχει σημειωθεί ραγδαία αύξηση της ποσότητας και της ποιότητας λογισμικού, τόσο για εμπορική χρήση όσο και για μη, που έχει σχεδιαστεί για να λύσει τα MILP.(5)

Έχοντας κατανοήσει όλα αυτά τα στοιχεία και χρησιμοποιώντας τις κατάλληλες τεχνικές της Επιχειρησιακής Έρευνας θα δοκιμάσουμε να λύσουμε το πρόβλημα της εργασίας μας.

1.2 Σκοπός Εργασίας

Η αποτελεσματική λειτουργία μιας αεροπορικής εταιρείας απαιτεί τον κατάλληλο προγραμματισμό αλλά και τα απαραίτητα εργαλεία και λογισμικά που χρησιμεύουν στις πολύπλοκες και μεγάλες δραστηριότητες μιας επιχείρησης. Τέτοιες δραστηριότητες όπως ο προγραμματισμός των πτήσεων, η ανάθεση πληρώματος σε πτήσεις και η δρομολόγηση των αεροσκαφών χρησιμοποιούν λογισμικά που αποφέρουν βέλτιστες λύσεις σε σύντομο χρονικό διάστημα.

Στις μέρες μας όμως η ραγδαία ανάπτυξη της αεροπορικής βιομηχανίας σε ένα εξαιρετικά ανταγωνιστικό παγκόσμιο δίκτυο μεταφορών ανάγκασε τις αεροπορικές εταιρείες να πραγματοποιούν σημαντικές επενδύσεις σε τέτοιου είδους λογισμικά και εργαλεία. Έτσι στράφηκαν στην χρήση της επιχειρησιακής έρευνας και σε τεχνικές βελτιστοποίησης ώστε να δημιουργήσουν τα κατάλληλα συστήματα για τον καλύτερο έλεγχο, την καλύτερη λειτουργία και συνεπώς την επικράτησή τους απέναντι στους ανταγωνιστές.

Ένα σοβαρό πρόβλημα που αντιμετωπίζουν οι αεροπορικές εταιρείες είναι ότι με την ανάπτυξη των συστημάτων ελέγχου το ανθρώπινο δυναμικό αργεί να προσαρμοστεί σε αυτά με αποτέλεσμα να δημιουργούνται ατυχήματα. Τόσο το προσωπικό στο πιλοτήριο όσο και στην καμπίνα θα πρέπει να είναι έτοιμοι ώστε ανά πάσα στιγμή να μπορούν να αντιμετωπίζουν οποιοδήποτε πρόβλημα συμβεί. Τη λύση ενός τέτοιου προβλήματος σε μια αεροπορική εταιρεία μπορεί να την δώσει η επιχειρησιακή έρευνα με μοντέλα βελτιστοποίησης που επεξεργάζονται τον βέλτιστο προγραμματισμό και την βέλτιστη κατανομή του προσωπικού σε τμήματα εκπαίδευσης. Ένα τέτοιο μοντέλο λοιπόν επεξεργάζεται, αναλύεται και παρουσιάζεται στην εργασία μας.

2. ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Με πρωταρχικό στόχο τη μείωση του λειτουργικού κόστους των αεροπορικών εταιρειών, έχουν πραγματοποιηθεί πολλές μελέτες καθώς και κατασκευαστεί μαθηματικά μοντέλα που αφορούν τον βέλτιστο προγραμματισμό του ιπτάμενου προσωπικού για εκπαίδευση. Με την επίλυση των μαθηματικών αυτών μοντέλων, παρουσιάστηκε σημαντική μείωση του κόστους των εταιρειών που τα εφάρμοσαν.

Τον Ιούνιο του 1981 ο M. Shapiro (6) περιγράφει ένα υπολογιστικό μοντέλο που κατασκεύασε ο ίδιος για τις Αμερικανικές Αερογραμμές (American Airlines). Το μοντέλο αφορούσε την εκπαίδευση των ιπτάμενων και τη μείωση του κόστους. Τα αποτελέσματα έδειξαν ότι είναι πιο αποδοτικό η εκπαίδευση να πραγματοποιείται μετά την απομάκρυνση των ιπτάμενων από το πρόγραμμα πτήσεων και όχι όταν έχουν ρεπό. Με τη χρήση αυτού του μοντέλου η American Airlines εξοικονόμησε σημαντικά ποσά από την εκπαίδευση του ιπτάμενου προσωπικού, της τάξεως των \$250,000/χρόνο.

Το 1995, ο R.P. Brown (7) αναπτύσσει στη διατριβή του ένα μοντέλο δι-κριτήριου μεικτού αέριου προγραμματισμού που αφορά την πολεμική αεροπορία και πιο συγκεκριμένα το Σώμα Πεζοναυτών. Το μοντέλο αυτό συσχετίζει τον προγραμματισμό της εκπαίδευσης των πεζοναυτών με την με την ετοιμότητά τους στη μάχη. Λαμβάνοντας δεδομένο ότι όλοι οι πεζοναύτες έχουν ίσες ευκαιρίες και για χρονικό διάστημα 90 ημερών, το μοντέλο μεγιστοποιεί μία συνάρτηση που αφορά την ετοιμότητα των πεζοναυτών και τη δίκαιη κατανομή των εργασιών που τους αναλογούν.

Το 1998 σε έρευνα που πραγματοποιήθηκε από τους Gang Yu, S. Dugan και M. Argüelo (8) στην πολιτική αεροπορία παρήγαγαν ένα πλήρες μοντέλο ανάθεσης πιλότων σε εκπαιδευτικές συνεδρίες. Μία ευρετική λύση του παραπάνω μοντέλου εφαρμόστηκε από την εταιρεία Continental Airlines, πράγμα που οδήγησε σε εξοικονόμηση πόρων τάξεως 6 εκατομμυρίων δολαρίων για κάθε χρόνο και αυτό βελτιώνοντας μόνο τον προγραμματισμό εκπαίδευσης των πιλότων. Δεν μπορούμε να παραβλέψουμε την ανάλυση των Gamache et al. (9) όπου εμπλουτίστηκε το σύστημα δίνοντας τη δυνατότητα επιλογής προτιμήσεων. Ο συνολικός σχεδιασμός ανθρώπινων πόρων λειτουργεί σύμφωνα με ένα σύστημα προτιμήσεων κατά το οποίο οι ιπτάμενοι επιλέγουν την προτίμησή τους με τη σειρά, βάσει της αρχαιότητάς τους (από τον αρχαιότερο στο νεότερο). Συνεπώς με δεδομένο την αυστηρή τήρηση της αρχαιότητας επιλύεται το πρόβλημα, όπως θα δούμε και στην δικιά μας ανάλυση.

Το 2003 ο AslanDavut (10) χρησιμοποιεί έναν ευρετικό αλγόριθμο με σκοπό τον προγραμματισμό των πτήσεων της μοίρας αξιοποιώντας τους διαθέσιμους πόρους και τηρώντας τα αυστηρά χρονοδιαγράμματα.

Οι Thengvall και Pachon (11) επεκτείνουν το μοντέλο που αναπτύχθηκε στην εργασία των Gamacheetal και παράγουν προγράμματα εκπαίδευσης που καλύπτουν τόσο τις μεταθέσεις πιλότων αλλά και τις αναθέσεις τους σε προγράμματα κατάρτισης εφαρμόζοντας ένα σύστημα προτιμήσεων (bidawardsystem). Ένα πρώτο επίπεδο εφικτότητας προκύπτει από την επίλυση ενός μοντέλου Μεικτού Ακέραιου Προγραμματισμού (ΜΑΠ) με γραμμική χαλάρωση, ενώ στη συνέχεια και επιλύεται το μοντέλο του ΜΑΠ για την εύρεση της βέλτιστης εφικτής λύσης.

Μια τελευταία βιβλιογραφική αναφορά είναι αυτή της Hábel (12) η οποία στη διατριβή της προτείνει την επίλυση του προβλήματος μεταθέσεων και εκπαίδευσης σε δύο φάσεις. Στην πρώτη φάση γίνεται ο καθορισμός του χρονοδιαγράμματος των μεταθέσεων με μηνιαία ακρίβεια, ενώ σε δεύτερη φάση προτείνεται ένα ολοκληρωμένο πρόγραμμα εκπαίδευσης. Οι ιπτάμενοι έχουν την δυνατότητα να επιλέξουν τις προτιμήσεις τους και στην τελική επιλογή λαμβάνονται υπόψη οι περιορισμοί αρχαιότητας.

3.ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Τα τελευταία χρόνια επιστήμονες και εταιρείες εργάζονται πάνω στην βελτίωση και στην εύρεση νέων τεχνολογιών και συστημάτων που θα αναβαθμίσουν όλους τους βιομηχανικούς τομείς. Ένας τέτοιος τομέας είναι και η αεροπορική βιομηχανία. Οι αεροπορικές εταιρείες λόγω της αύξησης της τεχνολογίας διαθέτουν καινούριο εξοπλισμό, λογισμικά και συστήματα ελέγχου ακόμα πιο αξιόπιστα. Εξαιτίας του μεγάλου ανταγωνισμού καλούνται να προσαρμόζονται στο καθετί νέο και να εστιάζουν στην βελτίωση τους.

Ένας σημαντικός κλάδος που απασχολεί όλες τις αεροπορικές εταιρείες είναι η συνεχής εκπαίδευση του ιπτάμενου προσωπικού. Η εκπαίδευση, οι ικανότητες και τα προσόντα του πληρώματος είναι απαραίτητο κομμάτι για την εταιρεία καθώς αποτελούν εγγύηση για την αποτελεσματικότητα και την ασφάλεια των αερομεταφορών. Επίσης η γνώση και η προσαρμογή σε όλο και πιο σύγχρονες μεθόδους μπορεί να διασφαλίσει την αποφυγή ατυχήματος αλλά και την γρήγορη αντιμετώπιση οποιουδήποτε προβλήματος εν ώρα πτήσης.

Το πρόγραμμα εκπαίδευσης του προσωπικού χωρίζεται σε δυο κατηγορίες. Η πρώτη είναι ο Πλήρης Προσομοιωτής Πτήσης (FFS)(13). Ένας **προσομοιωτής πτήσης** είναι μια συσκευή που δημιουργεί τεχνητά μια πτήση των αεροσκαφών και το περιβάλλον στο οποίο πετάει που χρησιμεύει για την πιλοτική εκπαίδευση. Περιλαμβάνει την αναπαραγωγή του λογισμικού, δηλαδή τον τρόπο με τον οποίο πετούν τα αεροσκάφη, τον τρόπο με τον οποίο αντιδρούν στις εφαρμογές ελέγχου πτήσης, τις επιπτώσεις άλλων συστημάτων αεροσκαφών και τον τρόπο με τον οποίο αντιδρούν τα αεροσκάφη σε εξωτερικούς παράγοντες όπως η πυκνότητα του αέρα, οι αναταράξεις και η διάτμηση του ανέμου. Η δεύτερη κατηγορία είναι τα Classrooms, όπου εκεί οι ιπτάμενοι και του πιλοτήριου και της καμπίνας εκπαιδεύονται θεωρητικά στις νέες τεχνολογίες και μεθόδους πτήσης.

Βέβαια είναι κατανοητό ότι το κόστος για την εκπαίδευση του πληρώματος είναι μεγάλο και πως θα πρέπει να δοθεί ιδιαίτερη βάση στον προγραμματισμό της χρονικής διάρκειας των μαθημάτων. Όμως καθίσταται απαραίτητο όλες οι αεροπορικές εταιρείες να διαθέτουν υπαλλήλους που θα πληρούν όλα τα κριτήρια που επιβάλλει η Διεθνής Υπηρεσία Αερομεταφορών (IATA=International Air-Transport Association).

3.1 Περιγραφή δεδομένων

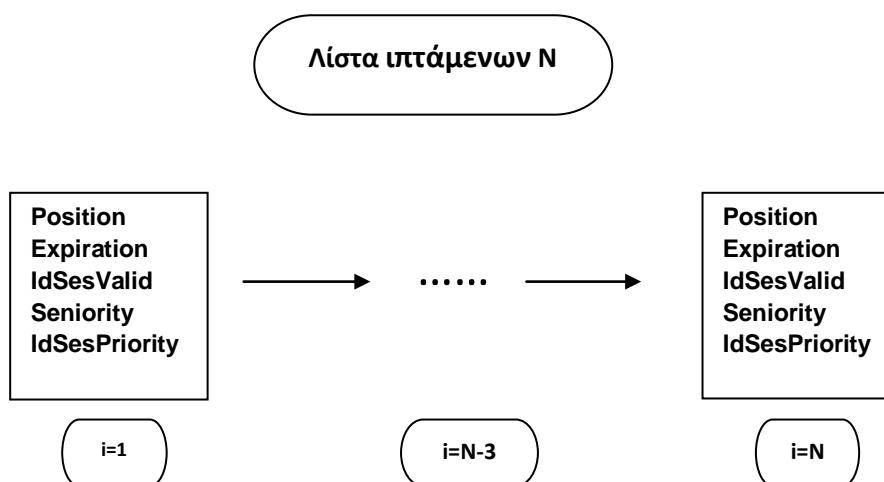
Το πρόβλημα που καλούμαστε να λύσουμε είναι αρκετά δύσκολο με πολύπλοκους υπολογισμούς. Όμως τα δεδομένα και οι συνιστώσες που χρησιμοποιούμε μπορούν να ποσοτικοποιηθούν εύκολα και το μετατρέπουν σε ένα αρκετά υπολογίσιμο. Τα δεδομένα του προβλήματος είναι τα παρακάτω.

Η λίστα με το ιπτάμενο προσωπικό: τα πρώτα στοιχεία που μας δίνονται είναι μια λίστα στην οποία αναγράφεται το προσωπικό της αεροπορικής εταιρείας το οποίο θα ανατεθεί στις συνεδρίες. Κάθε άτομο αυτής της λίστας συνοδεύεται με τα παρακάτω χαρακτηριστικά

- **Θέση (position).** Κάθε άτομο αντιστοιχεί σε συγκεκριμένη θέση σε κάθε εκπαιδευτική συνέδρια. Οι θέσεις προορίζονται για τέσσερις κατηγορίες ιπτάμενων οι δυο για το πλήρωμα του πιλοτήριου που είναι οι κυβερνήτες (**cp**) και οι συγκυβερνήτες (**fo**) και οι άλλοι δυο για το πλήρωμα της καμπίνας, τους αεροσυνοδούς (**flight attendants**) και τους αρχιφροντιστές (**pursers**). Στο πρόβλημα μας η θέση με την τιμή 1 πάει στον κυβερνήτη, η 2 στον συγκυβερνήτη, η 3 στον αρχιφροντιστή και η 4 στον αεροσυνοδό. Με την ολοκλήρωση του προβλήματος ο κάθε ιπτάμενος θα πρέπει να έχει ανατεθεί στην σωστή θέση στην συνέδρια ανάλογα με την τιμή που έχει.
- **Η λίστα με τις εφικτές εκπαιδευτικές συνεδρίες (IdSesValid).** Αυτή η λίστα περιλαμβάνει τις εκπαιδευτικές συνεδρίες στις οποίες είναι εφικτό να ανατεθεί κάθε ιπτάμενος. Αυτή η λίστα καθορίζεται ανάλογα με το εργασιακό πρόγραμμα του κάθε ιπτάμενου όπως πτήσεις, άδειες κ.α. έτσι ώστε να μπορεί ανά πάσα στιγμή να παρακολουθήσει την συνέδρια. Έτσι στο πρόβλημα μας με την ένδειξη true δηλώνεται η εφικτότητα και με την false η μη εφικτότητα για την κάθε συνέδρια.
- **Λήξη (expiration).** Σύμφωνα με το καταστατικό της κάθε αεροπορικής εταιρείας οι ιπτάμενοι θα πρέπει να παρακολουθούν κάθε εκπαιδευτική συνέδρια σε ένα συγκεκριμένο χρονικό διάστημα ώστε να μην υπάρχουν κενά στο πρόγραμμα των πτήσεων. Έτσι η λήξη της ισχύος της πτητικής ικανότητας του κάθε ιπτάμενου προσδιορίζεται είτε τον παρόντα μηνά είτε σε ένα μηνά είτε σε 2 μήνες. Στο πρόβλημα μας με την τιμή -1 αναφερόμαστε στους ιπτάμενους που η πτητική τους ικανότητα λήγει τον τρέχοντα μηνά, με 0 στους ιπτάμενους που η πτητική τους ικανότητα λήγει σε ένα μηνά και με 1 στους ιπτάμενους που η πτητική τους ικανότητα λήγει σε δυο μήνες. Καθώς αναλύουμε το πρόβλημα θα πρέπει να δοθεί μεγάλη

προσοχή έτσι ώστε ο ιπτάμενος με τιμή -1 να ανατεθεί οπωσδήποτε σε μια συνέδρια του τρέχοντος μηνός, με την προϋπόθεση ότι είναι εφικτή ανεξαρτήτως της προτίμησης του. Οι υπόλοιποι ανατίθενται ανάλογα των προτιμήσεων τους.

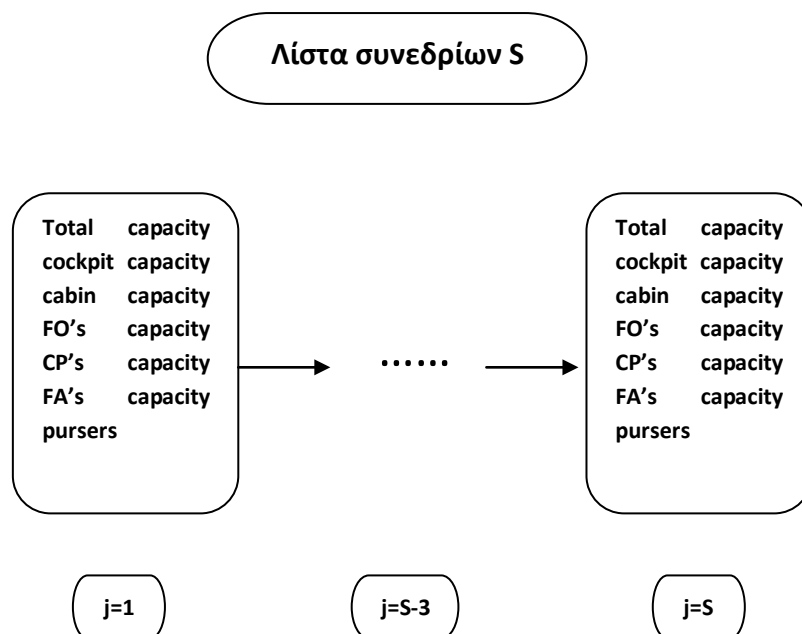
- **Αρχαιότητα (seniority).** Κάθε ιπτάμενος ξεχωρίζει από έναν άλλον με βάση την αρχαιότητα του η οποία καθορίζεται από την αεροπορική εταιρεία λαμβάνοντας υπόψιν τα χρόνια υπηρεσίας και των αριθμό των πτήσεων. Η αρχαιότητα συμβολίζεται με ένα ακέραιο αριθμό ο οποίος είναι μοναδικός για κάθε ιπτάμενο. Όσο μεγαλύτερη είναι αυτή η τιμή τόσο μικρότερος είναι ο βαθμός της αρχαιότητας του ιπτάμενου.
- **Προτεραιότητα προτιμήσεων (IdSesPriority).** Κάθε ιπτάμενος δημιουργεί μια λίστα στην οποία υπάρχουν η προτιμήσεις του για κάθε εφικτή εκπαιδευτική συνέδρια. Σ' αυτήν την λίστα υπάρχει ένας ακέραιος αριθμός που δηλώνει την προτίμηση του στην συγκεκριμένη συνεδρία. Ο ιπτάμενος με την τιμή 0 δηλώνει ότι δεν επιθυμεί να ανατεθεί στην συγκεκριμένη συνέδρια ή ότι δεν είναι εφικτή για αυτόν ενώ με θετικούς ακέραιους αριθμούς δηλώνει την προτεραιότητα των επιθυμιών του. Αυτοί οι θετικοί αριθμοί κυμαίνονται από το 1 έως το 8 και όσο αυξάνονται τόσο μικραίνει η προτεραιότητα των προτιμήσεων του. Υπάρχει η δυνατότητα πολλές συνεδρίες να έχουν προτίμηση 0 από έναν ιπτάμενο όμως δεν γίνεται δυο ή περισσότερες συνεδρίες να έχουν την ίδια θετική προτίμηση.



εικόνα 3 1: Σχηματική απεικόνιση της λίστας των ιπτάμενων

Επίσης σημαντικό δεδομένο εισόδου είναι και η **λίστα με τις εκπαιδευτικές συνεδρίες** η οποία περιλαμβάνει τις διαθέσιμες συνεδρίες που πραγματοποιούνται κάθε μήνα. Αυτή η λίστα έχει ένα χαρακτηριστικό την χωρητικότητα η οποία δημιουργεί ορισμένους περιορισμούς κατά την λύση του προβλήματος.

- **Χωρητικότητα (capacity).** Η ανάθεση των ιπτάμενων στις συνεδρίες εξαρτάται ανάλογα με την χωρητικότητα της. Στην περίπτωση του προσομοιωτή πτήσης η ανάθεση γίνεται με τέτοιο τρόπο ώστε ποτέ σε ένα μάθημα να μην υπάρχουν ταυτόχρονα δυο κυβερνήτες ή 2 συγκυβερνήτες. Στην περίπτωση των classrooms είναι λίγο πιο περίπλοκο. Αρχικά η συνολική χωρητικότητα σε μια συνέδρια καθορίζεται ανάλογα με τον αριθμό των ιπτάμενων N και ποτέ δεν πρέπει να ξεπερνάμε αυτόν τον αριθμό. Επίσης υπάρχουν περιορισμοί για την χωρητικότητα οι οποίοι μας δίνουν τον μέγιστο αριθμό των ιπτάμενων της καμπίνας και του πιλοτήριου που θα ανατεθούν σε κάθε συνέδρια. Τέλος θα πρέπει να δίνεται προσοχή και στον αριθμό των τεσσάρων ειδικοτήτων που θα μπαίνουν σε μια συνέδρια. Οι περιορισμοί που μας δίνει η χωρητικότητα είναι πολύ ισχυροί και απαραβίαστοι και είναι αυτοί που καθορίζουν σε μεγάλο βαθμό την λύση του προβλήματος.



εικόνα 3 2:Σχηματική απεικόνιση της λίστας των εκπαιδευτικών συνεδριών

3.2 Περιγραφή της λύσης του προβλήματος

Εφόσον καταγράψαμε και αναλύσαμε τα δεδομένα εισόδου που μας δίνονται για την επίλυση του προβλήματος, στην συνέχεια θα αναφερθούμε στα βασικά χαρακτηριστικά που θα πρέπει να έχει η απαιτούμενη βέλτιστη λύση. Κύριο μέλημα αλλά και βασικός μας στόχος κατά την επίλυση είναι η βέλτιστη αξιοποίηση των ανθρώπινων και υλικών πόρων λαμβάνοντας πάντα σοβαρά υπόψιν μας τους περιορισμούς που δημιουργούνται από την αρχαιότητα και τις προτιμήσεις του κάθε ιπτάμενου αλλά και από την χωρητικότητα της κάθε συνεδρίας.

Αρχικά το πρόβλημα επεξεργάζεται τα μέλη του πληρώματος που η λήξη της πτητικής τους ικανότητας είναι τον τρέχοντα μήνα ($expiration=-1$). Κάθε ιπτάμενος με λήξη πτητικής ικανότητας -1 θα ανατεθεί σε μια συνεδρία που είναι εφικτή για αυτόν. Εφόσον ολοκληρωθεί η ανάθεση τους οι άδειες θέσεις που θα υπάρχουν θα καλυφτούν πρώτα από τους ιπτάμενους που η πτητική τους ικανότητα λήγει τον επόμενο μήνα ($expiration=0$) και οι υπόλοιπες από τους ιπτάμενους που η πτητική τους ικανότητα λήγει τον μεθεπόμενο μήνα ($expiration=1$). Στο τέλος αυτής της διαδικασίας θα πρέπει να έχουν καλυφτεί στο μέγιστο οι θέσεις στις εκπαιδευτικές συνεδρίες, δηλαδή να μην υπάρχει ιπτάμενος που να μην έχει ανατεθεί όταν υπάρχουν διαθέσιμες θέσεις.

Βασική προϋπόθεση για την ανάθεση των ιπτάμενων στις συνεδρίες είναι η τήρηση των περιορισμών χωρητικότητας που αφορούν το σύνολο των ιπτάμενων, το πλήρωμα καμπίνας και πιλοτήριου και των διαφορετικών ειδικοτήτων των εκπαιδευμένων.

Ο κώδικάς μας ακολουθεί τους περιορισμούς αρχαιότητας. Έχοντας ως δεδομένο ότι ένας ιπτάμενος με λήξη της πτητικής του ικανότητας -1 θα ανατεθεί σε μια εφικτή συνεδρία αλλά και πως όλη η διαδικασία ολοκληρώνεται όταν έχουν καλυφτεί όλες οι διαθέσιμες θέσεις, το πρόγραμμα μας προσπαθεί να ικανοποιήσει την μεγαλύτερη επιθυμία των αρχαιότερων ιπτάμενων.

Αρχικά ελέγχεται η πρώτη επιθυμία του αρχαιότερου ιπτάμενου με πτητική ικανότητα που λήγει τον παρόντα μήνα και εφόσον είναι εφικτή ανατίθεται στην συγκεκριμένη συνεδρία. Αυτή η διαδικασία συνεχίζεται και για τους υπόλοιπους ιπτάμενους κατά σειρά αρχαιότητας. Όμως ένα πρόβλημα που μπορεί να αντιμετωπίσουμε είναι να βρεθούμε σε μια κατάσταση όπου η πρώτη επιθυμία κάποιου ιπτάμενου (πάντα με $expiration=-1$) να έχει καλυφτεί από έναν αρχαιότερο ιπτάμενο. Σ' αυτήν την περίπτωση για να μην οδηγηθούμε στην μη ανάθεση ενός υπαλλήλου με $expiration=-1$ ή στη μη κάλυψη μια θέσεις, ελέγχουμε την δεύτερη επιθυμία ή την τρίτη και ούτω καθεξής μέχρι να γίνει ανάθεση σε μια εφικτή

συνεδρία χωρίς πάντα να επηρεαζόμαστε από τις επιθυμίες των ιπτάμενων που είναι κάτω στην λίστα της αρχαιότητας. Τέλος εάν καμία από τις προτιμήσεις του δεν ικανοποιηθεί τότε είμαστε αναγκασμένοι να τον αναθέσουμε σε οποιαδήποτε εφικτή συνεδρία που υπάρχει κενή θέση.

Όταν ολοκληρωθεί η ανάθεση όλων των ιπτάμενων με $expiration=-1$ οι κενές θέσεις που θα περισσέψουν θα καλυφτούν αρχικά από ιπτάμενους με $expiration = 0$. Και σ' αυτήν την περίπτωση ελέγχουμε πάλι τις επιθυμίες των υπαλλήλων σύμφωνα με την αρχαιότητα τους. Ξεκινώντας από τον παλαιότερο προς τον νεότερο προσπαθούμε να τους αναθέσουμε σε κενές εφικτές θέσεις ανάλογα με την σειρά των προτιμήσεων τους. Εάν και πάλι δεν καλύπτονται από καμία προτίμηση τότε οι ιπτάμενοι θα ανατεθούν και αυτοί στις υπόλοιπες διαθέσιμες θέσεις των εφικτών συνεδρίων. Τέλος αν μετά και την ανάθεση των ιπτάμενων με $expiration=0$ υπάρχουν ακόμα άδειες θέσεις τότε θα προχωρήσουμε στον έλεγχο και των ιπτάμενων με $expiration=1$. Στις θέσεις αυτές θα ανατεθούν ιπτάμενοι μόνο κατά βάση των επιθυμιών τους και πάντα ακολουθώντας τους περιορισμούς αρχαιότητας και αν βρεθούν ιπτάμενοι που οι προτιμήσεις τους δεν συμπίπτουν με καμία διαθέσιμη εφικτή θέση ή έχουν καλυφτεί όλες οι θέσεις των συνεδρίων τότε η ανάθεση τους θα μεταφερθεί τον επόμενο μήνα που θα έχει αλλάξει η λήξη της πτητικής τους ικανότητας σε 0.

Τέλος το πρόγραμμα μας προσπαθεί να κρατήσει μια ισορροπία (**balance**) ανάμεσα στον αριθμό των ιπτάμενων που θα ανατεθούν σε κάθε εκπαιδευτική συνεδρία. Έχοντας πάντα υπόψιν μας τις προτιμήσεις των ιπτάμενων και χωρίς να τις προσπερνάμε προσπαθούμε οι συνεδρίες να έχουν σχετικά ίδιες θέσεις κατειλημμένες και να μην υπάρχει συσσώρευση εκπαιδευτών σε μια συνεδρία και πολλές κενές θέσεις σε άλλες. Έτσι στόχος πάντα είναι να δίνεται μια ισορροπημένη λύση στο πρόβλημα μας.

4. ΜΑΘΗΜΑΤΙΚΗ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Στο παρόν κεφάλαιο αναπτύσσουμε το μαθηματικό μοντέλο βελτιστοποίησης που είναι απαραίτητο για την επίλυση του προβλήματος ανάθεσης ιπτάμενων σε εκπαιδευτικές συνεδρίες προσομοιωτή και classroom. Επιπλέον, θα περιγραφούν τα δεδομένα εισόδου, θα πραγματοποιηθεί μαθηματική αποτύπωση της αντικειμενικής συνάρτησης και των περιορισμών του προβλήματος.

4.1 Δεδομένα εισόδου

Μέσω της μορφοποίησης εξάγουμε ένα πρόγραμμα σύμφωνα με το οποίο θα καθορίζεται σε ποια εκπαιδευτική συνεδρία θα ανατεθεί ο κάθε ιπτάμενος. Οι μαθηματικοί συμβολισμοί που χρησιμοποιούμε για την αποτύπωση του προβλήματος είναι οι εξής:

Σύνολα	
N	Σύνολο ιπτάμενων (κυβερνήτες, συγκυβερνήτες, αεροσυνοδοί, αρχιφροντιστές)
S	Σύνολο εκπαιδευτικών συνεδριών

Πίνακας 4 1: Σύνολα του μαθηματικού μοντέλου

Δείκτες	
i	Ιπτάμενος (κυβερνήτης, συγκυβερνήτης, αεροσυνοδός, αρχιφροντιστής)
j	Εκπαιδευτική συνεδρία

Πίνακας 4 2: Δείκτες του μαθηματικού μοντέλου

Παράμετροι	
c_{ij}	Συντελεστής κόστους μεταβλητών απόφασης x_{ij} της αντικειμενικής συνάρτησης
b_i	Συντελεστής κόστους των μεταβλητών απόφασης y_i της αντικειμενικής συνάρτησης
c_w	Συντελεστής κόστους της μεταβλητής απόφασης w της αντικειμενικής συνάρτησης
P	Συντελεστής κόστους των μεταβλητών απόφασης $v_{pos,j}$ της αντικειμενικής συνάρτησης που είναι ίδιος για αυτές τις μεταβλητές
Expiration _i	Λήξη ισχύος πτητικής ικανότητας του ιπτάμενου i (-1 αν λήγει τον τρέχον μήνα, 0 αν λήγει τον επόμενο μήνα, 1 αν λήγει τον μεθεπόμενο μήνα)
Position _i	Θέση του ιπτάμενου (1 για κυβερνήτη, 2 για συγκυβερνήτη, 3 για αεροσυνοδό και 4 για αρχιφροντιστή)
Seniority _i	Αρχαιότητα του ιπτάμενου i (μοναδικός ακέραιος, όσο μικρότερος τόσο μεγαλύτερη η αρχαιότητα του ιπτάμενου)
Priority _{ij}	Τιμή προτεραιότητας που θέτει ο κάθε ιπτάμενος i για να εκφράσει επιθυμία για κάθε συνεδρία j (0 αν δεν έχει εκφράσει επιθυμία, 1 για την πρώτη επιθυμία, 2 για την δεύτερη κ.ο.κ.)

Πίνακας 4 3: Συμβολισμοί των παραμέτρων του μαθηματικού μοντέλου

4.2 Μεταβλητές απόφασης

Μετά την ολοκλήρωση της καταγραφής των απαραίτητων δεδομένων εισόδου μπορούμε να συνεχίσουμε με την διαδικασία του σχεδιασμού. Σε αυτή την ενότητα παραθέτονται οι μεταβλητές απόφασης που χρησιμοποιούμε για την μαθηματική διατύπωση του προβλήματος.

4.2.1 Δυαδικές μεταβλητές απόφασης

Η πρώτη κατηγορία δυαδικών μεταβλητών αφορά την ανάθεση κάθε ιπτάμενου i ($i \in N$) σε κάθε εκπαιδευτική συνεδρία j ($j \in S$), αρκεί φυσικά να ανήκει στη λίστα εφικτών συνεδριών του ιπτάμενου i .

$$x_{ij} = \begin{cases} 1, & \text{αν ο ιπτάμενος } i \text{ ανατεθεί στην εκπαιδευτική συνεδρία } j \\ 0, & \text{αν όχι} \end{cases}$$

Η δεύτερη κατηγορία αφορά τη μη ανάθεση των ιπτάμενων. Αυτή η μεταβλητή απόφασης ορίζεται για του ιπτάμενους των οποίων η πτητική ικανότητα λήγει τον επόμενο είτε τον μεθεπόμενο μήνα.

$$y_i = \begin{cases} 1, & \text{αν ο ιπτάμενος } i \text{ δεν ανατεθεί στην εκπαιδευτική συνεδρία του τρέχοντος μήνα} \\ 0, & \text{αν ανατεθεί} \end{cases}$$

4.2.2 Ακέραιες μεταβλητές απόφασης

Η ακέραια αυτή μεταβλητή δίνει τον αριθμό των ιπτάμενων που δεν έχουν ανατεθεί σε κάποια εκπαιδευτική συνεδρία.

$$w = \text{συνολικός αριθμός ιπτάμενων που δεν ανατίθεται στις συνεδρίες}$$

4.3 Μαθηματικό Μοντέλο

Ολοκληρώνοντας και τη διαδικασία ορισμού των μεταβλητών απόφασης, σειρά έχει η περιγραφή της αντικειμενικής συνάρτησης καθώς και των περιορισμών που την πλαισιώνουν.

4.3.1 Αντικειμενική συνάρτηση

Η μαθηματική μορφή της αντικειμενικής συνάρτησης είναι η εξής:

$$\min \sum_{i=1}^N \sum_{j=1}^S c_{ij} x_{ij} + \sum_{i=1}^N b_i y_i + c_w w + (S-1)p \sum_{j=1}^S v_{pos,j}$$

Στόχος της αντικειμενικής συνάρτησης είναι να ελαχιστοποιηθεί το κόστος ανάθεσης των ιπτάμενων στις εκπαιδευτικές συνεδρίες έτσι ώστε να πετύχουμε την μέγιστη δυνατή ικανοποίησή τους, τηρώντας σειρά αυστηρής αρχαιότητας. Η μεταβλητή ανάθεσης x_{ij} δεν ορίζεται όταν κάποια συνεδρία j δεν ανήκει στη λίστα των εφικτών συνεδριών ενός ιπτάμενου i . Συνεχίζοντας στην ίδια λογική, η μεταβλητή μη ανάθεσης y_i δεν ορίζεται αν η πτητική ικανότητα ενός ιπτάμενου i λήγει τον τρέχων μήνα.

Όσον αφορά τη μορφή της αντικειμενικής συνάρτησης, ο πρώτος όρος των αθροισμάτων αφορά τις μεταβλητές ανάθεσης ενός ιπτάμενου i στις εκπαιδευτικές συνεδρίες. Ο δεύτερος όρος αφορά τις μεταβλητές μη ανάθεσης για τον τρέχων μήνα και τέλος, ο τρίτος όρος αναφέρεται στον συνολικό αριθμό των ιπτάμενων που δεν ανατίθενται στις εκπαιδευτικές συνεδρίες. Ο τελευταίος όρος αφορά τις ακέραιες τιμές ισορροπίας.

Οι συντελεστές των μεταβλητών απόφασης ή συντελεστές κόστους αυξάνονται όσο αυξάνονται και οι τιμές προτεραιότητας. Πρωταρχικό μέλημα είναι να δίνεται προτεραιότητα στην ικανοποίηση των ιπτάμενων με μεγαλύτερο βαθμό αρχαιότητας. Σημαντικό είναι να αναφερθεί ότι αν ο ιπτάμενος i ανατεθεί στην συνεδρία j , η οποία θα είναι η πρώτη του επιλογή (δηλ. τιμή προτεραιότητας=1) τότε ο συντελεστής κόστους θα ελαχιστοποιηθεί, δηλαδή θα είναι 0. Γενικότερα, όσο μικρότερος είναι ο συντελεστής κόστους, τόσο μεγαλύτερη θα είναι η ικανοποίηση του ιπτάμενου i όταν ανατεθεί σε μία συνεδρία j . Ο στόχος του προβλήματος όμως είναι να μείνουν όσο το δυνατόν λιγότεροι ιπτάμενοι εκτός των εκπαιδευτικών συνεδριών άρα πρέπει να ελαχιστοποιηθεί και το κόστος της μεταβλητής w , έτσι ώστε να υπάρχουν όσο το δυνατόν λιγότερες κενές θέσεις στις συνεδρίες.

4.3.2 Περιορισμοί

1. Περιορισμοί ανάθεσης

$$x_{i1} + x_{i2} + \dots + x_{ij} + y_i = 1$$

Σε περίπτωση που η μεταβλητή μη ανάθεσης y_i είναι 0 ο ιπτάμενος πρέπει να ανατεθεί σε μία από τις εφικτές συνεδρίες. Αν η y_i ισούται με 1 τότε ο ιπτάμενος i δεν θα περάσει από εκπαίδευση τον τρέχοντα μήνα. Όσον αφορά τους ιπτάμενους

που πρέπει να ανατεθούν τον τρέχοντα μήνα σε εκπαιδευτικές συνεδρίες (δηλ. $expiration=-1$), αν δεν υπάρχει εφικτή θα μείνουν εκτός των

2. Περιορισμοί χωρητικότητας για τον προσομοιωτή

Οι εκπαιδευτικές συνεδρίες του προσομοιωτή αφορούν μόνο το πλήρωμα του πιλοτηρίου, δηλαδή τους κυβερνήτες και τους συγκυβερνήτες.

Όσον αφορά τους κυβερνήτες για κάθε εκπαιδευτική συνεδρία ισχύει ο περιορισμός

$$x_{1j} + x_{2j} + \dots + x_{nj} \leq \text{MaxSimCP}[j]$$

Οι όροι στο πρώτο μέλος περιλαμβάνουν τους ιπτάμενους που έχουν εφικτή συνεδρία την j . Το δεύτερο μέλος εκφράζει τον μέγιστο αριθμό των κυβερνητών που μπορούν να βρίσκονται στο πιλοτήριο σε μία συνεδρία j . Ο περιορισμός εξασφαλίζει ότι ο αριθμός των κυβερνητών που θα ανατεθούν στην συγκεκριμένη συνεδρία δεν ξεπερνάει τη μέγιστη χωρητικότητα του προσομοιωτή.

Με όμοιο τρόπο λειτουργεί ο περιορισμός και για τους συγκυβερνήτες. Για κάθε εκπαιδευτική συνεδρία j ισχύει ο περιορισμός

$$x_{1j} + x_{2j} + \dots + x_{nj} \leq \text{MaxSimFO}[j]$$

Οι όροι στο αριστερό μέλος αφορούν τους συγκυβερνήτες που έχουν εφικτή συνεδρία την j . Στο δεξί μέλος βρίσκεται ο αριθμός που δηλώνει τη μέγιστη χωρητικότητα του προσομοιωτή σε συγκυβερνήτες. Συνολικά ο περιορισμός εξασφαλίζει το ότι ο αριθμός των συγκυβερνητών που έχουν εφικτή τη συνεδρία j να μην ξεπερνά τη μέγιστη χωρητικότητα.

3. Περιορισμοί χωρητικότητας τάξης

Σε αυτή την περίπτωση τις εκπαιδευτικές συνεδρίες πρέπει να τις παρακολουθήσουν όλοι οι ιπτάμενοι ανεξάρτητα αν πρόκειται για πλήρωμα πιλοτηρίου ή πλήρωμα καμπίνας.

Για κάθε εκπαιδευτική συνεδρία ισχύει ο περιορισμός

$$x_{1j} + x_{2j} + \dots + x_{nj} \leq \text{MaxTotCap}[j]$$

Στο αριστερό μέλος βρίσκονται όλα τα μέλη του πληρώματος που έχουν εφικτή την εκπαιδευτική συνεδρία j . Με τον περιορισμό αυτό εξασφαλίζεται ότι ο αριθμός των εκπαιδευόμενων δεν θα ξεπερνά την μέγιστη χωρητικότητα της τάξης.

Σειρά έχει ο περιορισμός που αφορά το πλήρωμα καμπίνας.

$$x_{1j} + x_{2j} + \dots + x_{nj} \leq \text{MaxCabinCap}[j]$$

Οι όροι του αριστερού μέλους αφορούν το προσωπικό καμπίνας (αεροσυνοδούς και αρχιφροντιστές) που έχει εφικτή συνεδρία την j . Ο περιορισμός εξασφαλίζει ότι ο αριθμός των ιπτάμενων που προορίζονται για αυτή την τάξη δεν θα ξεπεράσει τον διαθέσιμο αριθμό των θέσεων.

Ομοίως ο περιορισμός που αφορά του πλήρωμα του πιλοτηρίου είναι ο εξής

$$x_{1j} + x_{2j} + \dots + x_{nj} \leq \text{MaxCockpitCap}[j]$$

Οι όροι του αριστερού μέλους αφορούν το προσωπικό καμπίνας που έχει εφικτή συνεδρία τη j . Το άθροισμα τους δεν πρέπει να υπερβαίνει τον διαθέσιμο αριθμό θέσεων της τάξης που προορίζεται αποκλειστικά για αυτήν την κατηγορία.

Σε μία προσπάθεια να πάμε την ανάλυση ένα βήμα παραπέρα, δημιουργούμε περιορισμούς χωρητικότητας που αφορούν την κάθε θέση ξεχωριστά.

$$\sum_{i \in N: \text{position}_i = \text{pos}} x_{ij} \leq \text{MaxClassCrew}[\text{pos}][j]$$

Με αυτόν των τρόπο δημιουργούνται τέσσερις ομάδες περιορισμών. Η πρώτη αφορά τους κυβερνήτες ($\text{pos}=1$), η δεύτερη τους συγκυβερνήτες ($\text{pos}=2$), η τρίτη τους αεροσυνοδούς ($\text{pos}=3$) και η τέταρτη τους αρχιφροντιστές ($\text{pos}=4$). Με τον περιορισμό αυτόν εξασφαλίζεται ότι δεν θα παραβιαστεί η χωρητικότητα για κάθε εκπαιδευτική συνεδρία j για κάθε μία από τις 4 θέσεις.

4. Περιορισμός ιπτάμενων που δεν θα ανατεθούν

$$\sum_{i \in N} y_i = w$$

Στην προσπάθεια να μείνουν όσο το δυνατόν λιγότεροι ιπτάμενοι εκτός των εκπαιδευτικών συνεδριών ορίζεται ο παραπάνω περιορισμός. Στο αριστερό μέλος βρίσκονται όλοι οι ιπτάμενοι οι οποίοι δεν μπόρεσαν να ανατεθούν σε κάποια συνεδρία. Το w είναι μία ακέραια μεταβλητή απόφασης. Το σύνολο N περιλαμβάνει όλους τους ιπτάμενους.

5. Περιορισμός αρχαιότητας για τον επόμενο μήνα.

Η ανάθεση των ιπτάμενων πρέπει να γίνει ακολουθώντας αυστηρή αρχαιότητα. Ο κάθε ιπτάμενος ανάλογα με την αρχαιότητά του θα ανατεθεί στη μεγαλύτερη προτίμηση από τις εφικτές που του αντιστοιχούν. Οι ιπτάμενοι των οποίων η πτητική ικανότητα λήγει τον τρέχοντα ή τον επόμενο μήνα θα πρέπει να ανατεθούν

σε εκπαιδευτικές συνεδρίες ακόμα και αν έχουν δηλώσει μηδενική προτίμηση, αρκεί βέβαια η συνεδρία αυτές να είναι εφικτές.

Άρα για κάθε ιπτάμενο i του οποίου η πτητική ικανότητα λήγει τον τρέχοντα ($exp=-1$) είτε τον επόμενο ($exp=0$) μήνα και για κάθε εφικτή συνεδρία j για την οποία δεν έχει εκφράσει επιθυμία ($priority=0$) :

$$x_{i1} + x_{i2} + \dots + x_{ij} \geq \frac{\sum_{k \in K} x_{kj}}{|K|}$$

Το αριστερό μέλος περιλαμβάνει εκτός από τη j , και όλες τις άλλες εφικτές εκπαιδευτικές συνεδρίες του ιπτάμενου i . Στο σύνολο K εμπεριέχονται όλοι οι αρχαιότεροι ιπτάμενοι που έχουν την ίδια θέση με τον i , έχουν εφικτή την συνεδρία j με τιμή προτεραιότητας 0 και η πτητική τους ικανότητα λήγει τον τρέχοντα είτε τον επόμενο μήνα. Περιλαμβάνονται επίσης και οι ιπτάμενοι που η πτητική τους ικανότητα λήγει τον μεθεπόμενο μήνα και δεν έχουν δηλώσει καμία προτίμηση για τη συνεδρία j .

Ο περιορισμός αυτός εξασφαλίζει ότι κάθε ιπτάμενος i του οποίου η πτητική ικανότητα λήγει τον τρέχοντα μήνα θα ανατεθεί σε μία εκπαιδευτική συνεδρία j , ακόμα και αν δεν έχει εκφράσει επιθυμία γι' αυτήν, πριν από οποιονδήποτε άλλον παλαιότερο ιπτάμενο i του οποίου η πτητική ικανότητα λήγει τον επόμενο μήνα και δεν έχει δηλώσει επιθυμία για τη συγκεκριμένη εκπαιδευτική συνεδρία j . Οι υπόλοιπες εφικτές συνεδρίες, εκτός από τη j , του ιπτάμενου i μπαίνουν στο αριστερό μέλος ώστε να μην είναι ενεργός ο περιορισμός σε περίπτωση που ο ιπτάμενος i έχει ανατεθεί ήδη σε κάποια άλλη εφικτή συνεδρία. Το $|K|$ εκφράζει τον αριθμό των στοιχείων που βρίσκονται στο σύνολο K .

6. Περιορισμός αρχαιότητας για τον μεθεπόμενο μήνα

Ο περιορισμός αυτός αφορά κάθε ιπτάμενο του οποίου η πτητική ικανότητα λήγει τον μεθεπόμενο μήνα ($exp=1$) και για κάθε εφικτή συνεδρία j που δεν έχει δηλώσει προτίμηση ($priority=0$). Όμοια λοιπόν με τον προηγούμενο περιορισμό

$$x_{i1} + x_{i2} + \dots + x_{ij} \geq \frac{\sum_{k \in K} x_{kj}}{|K|}$$

Το αριστερό μέλος του περιορισμού περιλαμβάνει εκτός από την j και όλες τις άλλες εφικτές συνεδρίες του i ιπτάμενου. Το σύνολο K στο δεξιό μέλος είναι το πλήθος των αρχαιότερων ιπτάμενων που έχουν την ίδια θέση με τον i ιπτάμενο, έχουν εφικτή την συνεδρία j με τιμή προτεραιότητας 0 και η πτητική τους ικανότητα λήγει τον μεθεπόμενο μήνα.

Ο περιορισμός εξασφαλίζει ότι κάθε ιπτάμενος του οποίου η πτητική ικανότητα λήγει τον επόμενο μήνα θα ανατεθεί σε μία εκπαιδευτική συνεδρία που δεν έχει

εκφράσει επιθυμία πριν από κάποιον άλλο αρχαιότερο ιπτάμενο ίδιας θέσης που επίσης δεν έχει δηλώσει προτίμηση για τη συγκεκριμένη συνεδρία και έχει λήξη πτητικής ικανότητα 2 μήνες ($\text{exp}=1$). Στο αριστερό μέλος θα μπουν και οι υπόλοιπες εφικτές συνεδρίες του κάθε ιπτάμενου έτσι ώστε ο περιορισμός να είναι ανενεργός σε περίπτωση που ο ιπτάμενος i έχει ήδη ανατεθεί σε κάποια άλλη εκπαιδευτική συνεδρία.

7. Περιορισμός ισορροπίας

Οι περιορισμοί αναφέρονται μόνο σε συνεδρίες που αφορούν τα τμήματα θεωρίας, τάξεις. Για κάθε θέση λοιπόν και κάθε ζεύγος εκπαιδευτικών συνεδριών j και k ακολουθεί ο περιορισμός:

$$\sum_{i \in L} x_{ij} + v_{ij} = \sum_{r \in R} x_{rk} + v_{rk}$$

Το σύνολο L περιλαμβάνει όλους τους ιπτάμενους ίδιας θέσης που έχουν εφικτή τη συνεδρία j . Το σύνολο R περιλαμβάνει όλους τους ιπτάμενους ίδιας θέσης που έχουν εφικτή τη συνεδρία k . Στόχος του περιορισμού είναι να αναθέσουμε, όσο αυτό είναι δυνατό, σε κάθε συνεδρία τον ίδιο αριθμό εκπαιδευόμενων. Οι μεταβλητές v_{ij} και v_{rk} είναι ακέραιες μεταβλητές απόφασης, μοναδικές για κάθε ζεύγος συνεδριών και μοναδικές για κάθε θέση. Συμμετέχουν στην αντικειμενική συνάρτηση με μικρό συντελεστή κόστους, ο οποίος είναι για όλες κοινός. Θετική τιμή μπορεί να πάρει μέχρι μία μεταβλητή και η άλλη θα είναι 0. Η θετική τιμή της μεταβλητής εκφράζει την διαφορά των ιπτάμενων μεταξύ των συνεδριών i και j , για κάθε θέση χωριστά. Ο περιορισμός αυτός βοηθάει στο να βρεθεί λύση τέτοια ώστε ο αριθμός των ιπτάμενων που έχουν ανατεθεί να διαφέρει ελάχιστα έως καθόλου από συνεδρία σε συνεδρία. Τα παραπάνω γίνονται με τέτοιο τρόπο ώστε να μην επηρεάζονται οι προτιμήσεις των ιπτάμενων.

4.3.3 Προσδιορισμός συντελεστών κόστους των μεταβλητών απόφασης

Έχει ήδη αναφερθεί στις προηγούμενες ενότητες ότι σε κάθε ιπτάμενο αντιστοιχεί μία λίστα εφικτών συνεδριών στις οποίες μπορεί να ανατεθεί. Για κάθε μία από αυτές τις συνεδρίες ο κάθε ιπτάμενος έχει εκφράσει την προτίμηση του με έναν αύξοντα ακέραιο αριθμό. Για την συνεδρία που επιθυμεί περισσότερο θα χρησιμοποιήσει τον αριθμό 1, για την δεύτερη τον αριθμό 2 κ.ο.κ. Επίσης, αν δεν έχει προτίμηση για κάποιες από αυτές μπορεί να χρησιμοποιήσει τον αριθμό 0. Όσο μεγαλώνει ο ακέραιος αριθμός τόσο λιγότερο θέλει ο κάθε ιπτάμενος να ανατεθεί σε αυτήν την συνεδρία και τόσο μεγαλύτερη θα είναι η δυσαρέσκεία του αν τελικά ανατεθεί σε αυτή. Ακόμα μεγαλύτερη δυσαρέσκεια στον ιπτάμενο θα προκαλέσει αν ανατεθεί σε μία συνεδρία που έχει δηλώσει προτίμηση 0.

Θα δημιουργήσουμε λοιπόν μία λίστα με όνομα PenaltyList αποτελούμενη από κόμβους. Ο κάθε κόμβος αντιπροσωπεύει από μία εφικτή συνεδρία για κάθε ιπτάμενο, δηλαδή αναφέρονται στις μεταβλητές ανάθεσης x_{ij} . Δημιουργούμε επίσης κόμβους για τους ιπτάμενους που δεν θα ανατεθούν σε κάποια συνεδρία, μεταβλητές μη ανάθεσης y_i . Τέλος, χρειαζόμαστε ακόμη έναν κόμβο που αντιστοιχεί στην ακέραια μεταβλητή w , η οποία δηλώνει τον συνολικό αριθμό των ιπτάμενων που δεν θα ανατεθούν σε κάποια συνεδρία τον τρέχοντα μήνα. Σειρά έχει η ταξινόμηση των κόμβων ώστε να δοθούν τιμές στους συντελεστές κόστους της αντικειμενικής συνάρτησης.

Οι κόμβοι της PenaltyList που αφορούν τον ίδιο ιπτάμενο είναι γειτονικοί. Οι αρχαιότεροι ιπτάμενοι τοποθετούνται στο τέλος της λίστας και ο τελευταίος κόμβος αντιστοιχεί πάντα στη μεταβλητή w . Για κάθε ιπτάμενο i πρώτα τοποθετείται η εφικτή συνεδρία με τη μικρότερη θετική τιμή, δηλαδή αυτή για τη οποία έχει δηλώσει τη μεγαλύτερη προτίμηση και ακολουθούν οι υπόλοιπες με αύξουσα σειρά προτεραιότητας. Έπειτα, σε περίπτωση που υπάρχει μεταβλητή μη ανάθεσης τοποθετείται ο αντίστοιχος κόμβος. Τέλος, τοποθετούνται οι συνεδρίες που έχουν τιμή προτεραιότητας 0 σε τυχαία σειρά. Η διάταξη περιγράφεται στο Σχήμα 4.1

Αρχικά δημιουργούμε ένα διάνυσμα $\max_priority[i]$ το οποίο θα περιέχει τη μέγιστη τιμή προτεραιότητας του κάθε ιπτάμενου, δηλαδή την επιλογή που θέλει λιγότερο από όλες τις άλλες. Στο επόμενο βήμα δημιουργούμε μία βοηθητική μεταβλητή με το όνομα $priority$ και την εισάγουμε σε κάθε κόμβο. Στην περίπτωση που ένας ιπτάμενος i δεν έχει δηλώσει προτίμηση για μία συνεδρία j , αλλά είναι εφικτή τότε η μεταβλητή $priority$ πάρει την τιμή 0. Αν ένας ιπτάμενος i έχει δηλώσει θετική προτεραιότητα $k > 0$ για μία συνεδρία j , τότε η μεταβλητή $priority$ πάρει την τιμή $\max_priority[i] + k - 2$. Για τους ιπτάμενους που η πτητική τους ικανότητα λήγει τον μεθεπόμενο μήνα η μεταβλητή $priority$ πάρει την τιμή 1, που αντιστοιχεί στον κόμβο για τη μη ανάθεση του ιπτάμενου αυτού. Η μεταβλητή $priority$ δεν χρειάζεται να οριστεί για τον κόμβο που αντιστοιχεί η μεταβλητή w .

Για να γίνει η ταξινόμηση πρώτα τοποθετούμε τη μεταβλητή w στο τέλος της λίστας. Προς το τέλος της λίστας τοποθετούνται οι κόμβοι των αρχαιότερων ιπτάμενων με φθίνουσα σειρά κατά seniority. Οι κόμβοι του ίδιου ιπτάμενου τοποθετούνται σε φθίνουσα σειρά κατά $priority$. Αφού τελειώσαμε με την ταξινόμηση της λίστας, προχωράμε στον προσδιορισμό των συντελεστών κόστους.

Για τον πρώτο ιπτάμενο ο συντελεστής κόστους που αντιστοιχεί στη συνεδρία με τη μικρότερη τιμή $priority$ (μεγαλύτερη επιθυμία) παίρνει την τιμή 0. Η συνεδρία με την αμέσως μικρότερη τιμή $priority$ έχει μεγαλύτερο συντελεστή κόστους και στις περιπτώσεις όπου ο ιπτάμενος δεν έχει δηλώσει προτεραιότητα, οι μεταβλητές κόστους θα έχουν ακόμα μεγαλύτερες τιμές αλλά ίδιες μεταξύ τους. Με τον ίδιο τρόπο συνεχίζουμε και για τον αμέσως επόμενο σε αρχαιότητα ιπτάμενο, δηλαδή

αυξάνουμε τον συντελεστή κόστους όσο μειώνεται η προτίμηση. Η μόνη διαφορά είναι ότι για να ικανοποιηθεί η αυστηρή αρχαιότητα, το άθροισμα των προηγούμενων συντελεστών κόστους δεν πρέπει να υπερβαίνει τον τρέχοντα συντελεστή.

Με τον τρόπο αυτό εξασφαλίζεται ότι οι μέγιστες εφικτές επιθυμίες των αρχαιότερων ιπτάμενων θα ικανοποιηθούν ακόμη και αν αυτό στοιχίσει στους ιπτάμενους με μικρότερη αρχαιότητα. Άρα θα πρέπει ο πρώτος θετικός συντελεστής κόστους κάθε ιπτάμενου i να είναι μεγαλύτερος από το άθροισμα των μέγιστων συντελεστών κόστους των νεότερων ιπτάμενων, ενώ κάθε επόμενος συντελεστής που αφορά τον ίδιο ιπτάμενο θα πρέπει να αυξάνει τουλάχιστον κατά το άθροισμα αυτό σε σχέση με τον αμέσως προηγούμενο.

Όσον αφορά τη μεταβλητή w , ο συντελεστής κόστους της πρέπει να είναι μεγαλύτερος από το άθροισμα των συντελεστών κόστους όλων των ιπτάμενων. Βασικό στόχος άλλωστε είναι να μειωθούν στο ελάχιστο δυνατό οι ιπτάμενοι που δεν θα ανατεθούν σε εκπαιδευτικές συνεδρίες τον τρέχοντα μήνα.

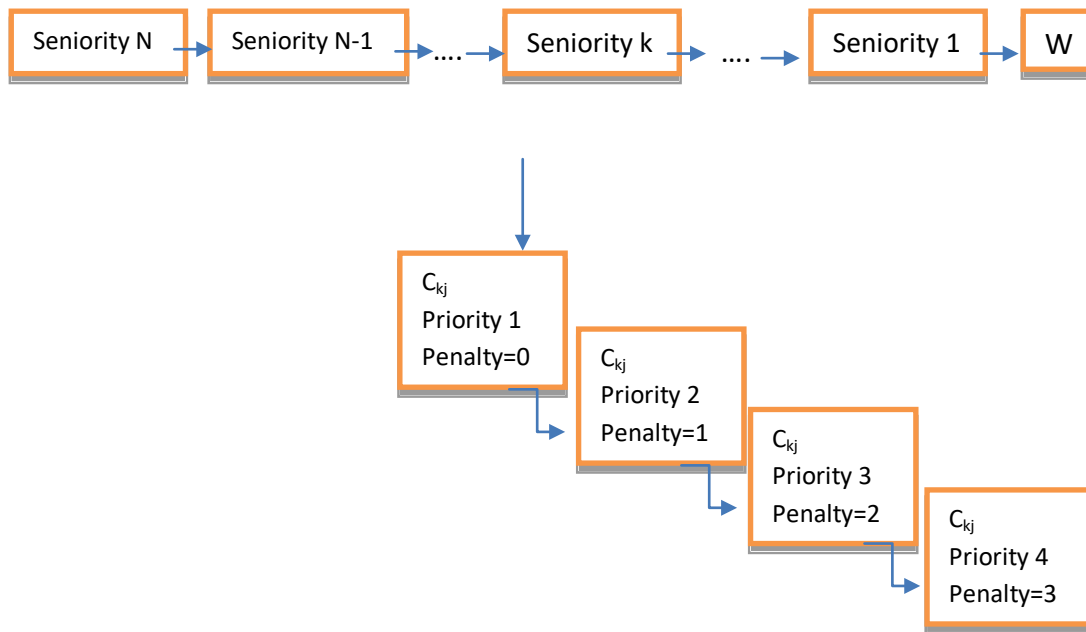
Σε περίπτωση που το πλήθος των ιπτάμενων και των εκπαιδευτικών συνεδριών είναι πολύ μεγάλο, οι συντελεστές κόστους μπορεί να λάβουν τόσο μεγάλες τιμές που να δημιουργήσουν προβλήματα στην επίλυση. Για αυτό το λόγο μία ενδεικνυόμενη λύση είναι η επίλυση του προβλήματος σε διαδοχικά στάδια.

Αρχικά ορίζουμε τιμές σε έναν συγκεκριμένο αριθμό συντελεστών κόστους, ξεκινώντας από το τέλος της λίστας προς την αρχή. Προτιμούμε μικρές τιμές έτσι ώστε να μην προκύψουν προβλήματα λόγω πολύ μεγάλων αριθμών. Όλους τους υπόλοιπους συντελεστές τους θέτουμε ίσους με 0. Λύνουμε το πρόβλημα κρατώντας κλειδωμένες τις αναθέσεις των ιπτάμενων. Η διαδικασία επαναλαμβάνεται διαδοχικά για όλους τους συντελεστές, μέχρι την εξάντληση της λίστας και την εύρεση της βέλτιστης λύσης.

Σημαντικό είναι να ορίσουμε συντελεστές σε όλους τους κόμβους που συσχετίζονται με έναν ιπτάμενο. Για αυτό το λόγο ο αριθμός των συντελεστών κόστους θα πρέπει να προσαρμοστεί κατάλληλα ώστε να συμπεριληφθούν όλοι οι συντελεστές κόστους του τελευταίου από αριστερά ιπτάμενου που θα είναι ενεργός στο συγκεκριμένο στάδιο επίλυσης.

Στο τέλος κάθε επανάληψης, κλειδώνουμε τις μεταβλητές που έχουν πάρει τιμή 1 και αντιστοιχούν σε ανάθεση, εισάγοντας έναν περιορισμό που επιβάλλει αυτή την τιμή. Οι περιορισμοί αυτοί ενσωματώνονται στο μοντέλο έτσι ώστε να χρησιμοποιούνται και για τις επόμενες επαναλήψεις. Οι μεταβλητές απόφασης που αντιστοιχούν στις τελευταίες προτιμήσεις για τον κάθε ιπτάμενο δεν χρειάζεται να κλειδωθούν ακόμη κι αν πάρουν την τιμή 1. Αυτό συμβαίνει επειδή έτσι κι αλλιώς ο

ιπάμενος αυτός θα πάει στην τελευταία προτίμηση οπότε δεν χρειάζεται να στερηθεί τη θέση αυτή ένας νεότερος ιπάμενος που την έχει υψηλότερα στη λίστα προτεραιοτήτων του.



Εικόνα 4 1: Σχηματική απεικόνιση της Penaltylist

5. ΤΑ ΣΤΑΔΙΑ ΤΗΣ ΕΠΙΛΥΣΗΣ

Έχοντας λοιπόν συγκεντρώσει όλα τα δεδομένα εισόδου και τα ζητούμενα ξεκινάμε να λύσουμε το πρόβλημα ακολουθώντας μια ιεραρχία, δηλαδή η διαδικασία επίλυσης του προβλήματος γίνεται σε ορισμένα διαδοχικά στάδια.

Όπως αναφερθήκαμε στα προηγούμενα κεφάλαια κάθε ιπτάμενος έχει κάποια συγκεκριμένα χαρακτηριστικά όπως η θέση του, η λήξη της πτητικής του ικανότητας, η λίστα με τις εφικτές συνεδρίες και οι προτιμήσεις του. Κάθε ιπτάμενος έχει δημιουργήσει τη δική του λίστα με τις προτιμήσεις του για τις συνεδρίες όπου είναι εφικτές για εκείνον. Την μεγαλύτερη του επιθυμία την δηλώνει στη λίστα με τον αριθμό 1 ενώ με τιμή προτεραιότητας 2,3,4 μέχρι και 8 δηλώνει τις συνεδρίες με φθίνοντα αριθμό επιθυμίας. Έτσι όσο αυξάνεται η τιμή προτεραιότητας μειώνεται η επιθυμία για την συγκεκριμένη συνεδρία. Ακόμα έχει την δυνατότητα να δηλώσει μια εφικτή εκπαιδευτική συνεδρία με τιμή προτεραιότητας ίση με 0 η οποία σημαίνει ότι δεν εκδηλώνει καμία επιθυμία για αυτήν την συνεδρία. Τέλος, γνώρισμα που διαδραματίζει μεγάλο ρόλο στην επίλυση είναι και η αρχαιότητα του κάθε ιπτάμενου.

Με βάση αυτά τα χαρακτηριστικά το πρώτο πράγμα που κάνουμε είναι να δημιουργήσουμε μια λίστα με την ανάθεση και τη μη ανάθεση των ιπτάμενων στις εκπαιδευτικές συνεδρίες. Αρχικά ελέγχουμε για κάθε ιπτάμενο ποιες από τις εκπαιδευτικές συνεδρίες είναι εφικτές για αυτόν. Για τις συγκεκριμένες συνεδρίες αν ο ιπτάμενος έχει δηλώσει τιμή προτεραιότητας κάποια θετική τιμή και όχι την τιμή μηδέν τότε θα γίνει η ανάθεση του στην συνεδρία. Βέβαια μπορούμε να προχωρήσουμε σε αυτήν την ανάθεση μόνο υπό την προϋπόθεση ότι η πτητική ικανότητα του ιπταμένου λήγει τον παρόντα μήνα ή τον επόμενο μήνα και όχι τον μεθεπόμενο μήνα. Στην αντίθετη περίπτωση όπου δεν ισχύουν τα παραπάνω τότε ο συγκεκριμένος ιπτάμενος δεν θα ανατεθεί σε καμία συνεδρία.

Το επόμενο στάδιο στην διαδικασία της επίλυσης είναι να καθοριστεί η σωστή χωρητικότητα για κάθε εκπαιδευτική συνεδρία. Τόσο στον προσομοιωτή όσο και στα τμήματα θεωρίας η χωρητικότητα είναι συγκεκριμένη. Για παράδειγμα στον προσομοιωτή υπάρχει ορισμένος αριθμός από κυβερνήτες και συγκυβερνήτες και τα τμήματα θεωρίας δέχονται συγκεκριμένους αριθμούς από την κάθε ειδικότητα και από τα πληρώματα καμπίνας και πιλοτήριου αντίστοιχα. Συνεπώς αυτό που καθορίζουμε στο συγκεκριμένο στάδιο είναι πως όλοι οι ιπτάμενοι που έχουν ανατεθεί σε μια συγκεκριμένη συνεδρία δεν θα πρέπει να ξεπερνούν σε αριθμό την χωρητικότητα αυτής.

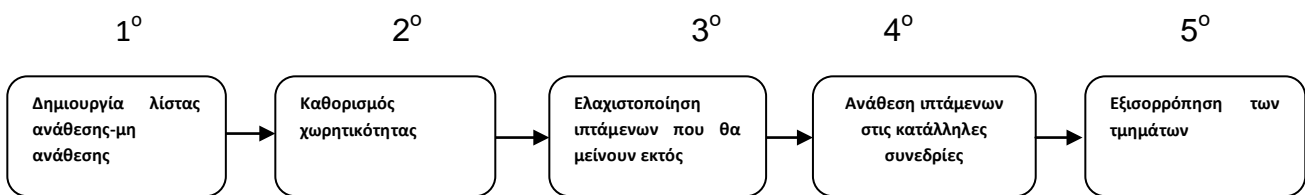
Το τρίτο σε σειρά στάδιο στην ιεραρχία είναι η ελαχιστοποίηση του αριθμού των ιπτάμενων με συγκεκριμένη λήξη της πτητικής τους ικανότητας οι όποιοι θα μείνουν εκτός, δεν θα υπάρξει δηλαδή ανάθεση. Κάθε ιπτάμενος έχει συγκεκριμένη λήξη της πτητικής του ικανότητας όπως τον παρόντα μήνα ($expr=-1$), τον επόμενο μήνα ($expr=0$) ή το μεθεπόμενο μήνα ($expr=1$). Η διαδικασία ξεκινά με τους ιπτάμενους που έχουν λήξη πτητικής ικανότητας με τιμή -1 και ελέγχουμε ποιοι από αυτούς θα μείνουν εκτός. Στην συνέχεια λύνουμε το πρόβλημα και θα πάρουμε μια εφικτή τιμή που θα μας δίνει τον αριθμό αυτών των ιπτάμενων. Έτσι θα έχουμε βρει τον βέλτιστο αριθμό ιπτάμενων στους οποίους δεν θα γίνει ανάθεση. Αυτός ο αριθμός θα ισχύει και για τα επόμενα στάδια της επίλυσης πράγμα που σημαίνει ότι στην συνέχεια δεν θα μπορούμε να αφήσουμε εκτός περισσότερους ιπτάμενους από όσους βρήκαμε σ' αυτό το στάδιο. Αφού ολοκληρώσουμε αυτήν την διαδικασία για τους ιπτάμενους με λήξη τον παρόντα μήνα θα την εκτελέσουμε ξανά πρώτα για αυτούς με λήξη τον επόμενο μήνα και τέλος για εκείνους με λήξη τον μεθεπόμενο μήνα. Έτσι θα έχουμε βρει και θα έχουμε κατοχυρώσει το σύνολο των ιπτάμενων που δεν θα ανατεθούν και για τις τρεις κατηγορίες.

Στο τέταρτο στάδιο θα προχωρήσουμε στην ανάθεση των ιπτάμενων στις κατάλληλες εκπαιδευτικές συνεδρίες. Θεωρώντας σταθερά και αδιαπραγμάτευτα τα αποτελέσματα από το προηγούμενο στάδιο της διαδικασίας αυτό που θέλουμε να πετύχουμε είναι κάθε ιπτάμενος, ανάλογα με την αρχαιότητα του (πρώτα ο πιο παλιός), να ανατεθεί στην μεγαλύτερη προτίμηση του από αυτές που είναι εφικτό να πάρει. Ξεκινώντας από τον πιο παλιό ελέγχω ποιες προτιμήσεις του είναι εφικτές. Για όσες είναι εφικτές βρίσκω την συνεδρία που έχει εκδηλώσει μεγαλύτερη προτίμηση. Αν η ανάθεση του σε αυτήν την συνεδρία είναι εφικτή τότε προχωράμε στην ανάθεση, αλλιώς ελέγχουμε την επόμενη μέχρι να βρεθεί μια εφικτή και να κλειδώσουμε την ανάθεση. Την ίδια διαδικασία εκτελούμε και για τους υπόλοιπους ιπτάμενους από τον παλαιότερο προς τον νεότερο και βρίσκω και για αυτούς την εφικτή ανάθεσή τους σε κάποια συνεδρία χωρίς να παραβιάζεται φυσικά το προηγούμενο στάδιο. Όπως και πριν τα αποτελέσματα και αυτού του σταδίου κλειδώνονται και επηρεάζουν την συνέχεια της επίλυσης.

Το τελευταίο στάδιο της επίλυσης του προβλήματος είναι να δημιουργήσουμε μια ισορροπία όσον αφορά τους ιπτάμενους σε κάθε συνεδρία. Έχοντας βρει από τα προηγούμενα στάδια, και θεωρώντας τα δεδομένα, τις συνεδρίες που θα ανατεθεί ο κάθε ιπτάμενος αλλά και τον αριθμό των υπαλλήλων που θα μείνουν εκτός θα προσπαθήσουμε να αναδιατάξουμε το σύνολο των εκπαιδευόμενων έτσι ώστε κάθε συνεδρία να έχει τις ίδιες κατειλημμένες θέσεις. Ελέγχοντας μόνο τα τμήματα θεωρίας και όχι τον προσομοιωτή βρίσκουμε τους υπόλοιπους ιπτάμενους που δεν έχουν ανατεθεί σε κάποια συνεδρία και τους τοποθετούμε σε τμήματα που είναι βέβαια εφικτά για αυτούς με τέτοιο τρόπο ώστε να υπάρχει ισορροπία μεταξύ των ατόμων του κάθε τμήματος και όχι συσσώρευση σε συγκεκριμένες συνεδρίες. Το

αποτέλεσμα που προκύπτει σε αυτό το στάδιο είναι η μεγίστη και η ελάχιστη απόκλιση από τον αριθμό των ιπτάμενων που θα ανατεθούν σε κάθε συνεδρία έτσι ώστε να μας δώσει μια ισορροπημένη λύση.

Συνεπώς ακολουθώντας την ιεραρχία των σταδίων κατά την επίλυση του προβλήματος θα μας δοθεί η βέλτιστη εφικτή λύση ελέγχοντας όλες τις παραμέτρους. Θα γίνει δηλαδή η βέλτιστη ανάθεση όλων των ιπτάμενων στις κατάλληλες εκπαιδευτικές συνεδρίες.



Εικόνα 5 1: Τα στάδια της επίλυσης

6. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ C++

Σε αυτό το κεφάλαιο περιγράφουμε την δομή του κώδικα που αναπτύξαμε για τη επίλυση του παραπάνω μαθηματικού μοντέλου. Η γλώσσα προγραμματισμού για την ανάπτυξη της δομής του κώδικα είναι η C++ και για την βελτιστοποίηση χρησιμοποιήσαμε το λογισμικό πακέτο βελτιστοποίησης IBM ILOG CPLEX Optimization Studio. Η σύνθεση και η επίλυση του κώδικα πραγματοποιήθηκε στο προγραμματιστικό περιβάλλον MicrosoftVisualStudio 2010.

Έχοντας ολοκληρώσει την γραφή του κώδικα προχωράμε στην εκτέλεση του και το πρώτο που θα εκτελεστεί είναι το κομμάτι εκείνο του κώδικα που βρίσκεται μέσα στην συνάρτηση **main()**. Αυτή η συνάρτηση περιέχει τις βασικότερες υπορουτίνες, που συνθέτουν τον κώδικα, σε σειρά και βρίσκεται συνήθως στο τέλος του προγράμματος. Οι εξής υπορουτίνες που καλούνται στην συνάρτηση main() είναι :

1. GenerateRandomData()
2. ReadData()
3. init_problem()
4. clock()
5. multiple_choice()
6. Sim_capacity()
7. Class_capacity()
8. fix_expiration(-1)
9. fix_expiration(0)
10. fix_expiration(1)
11. fix_bids()
12. Class_balance()
13. evaluate_results()
14. PrintProblem(env, lp, "myprob.lp", NULL)
15. Clock()
16. FreeAll()

Στη συνέχεια περιγράφουμε και αναλύουμε την δομή και την λειτουργία κάθε υπορουτίνας ξεχωριστά. Πριν την έναρξη της επίλυσης γνωρίζουμε πάντα τον συνολικό αριθμό ιπτάμενων N και τον συνολικό αριθμό εκπαιδευτικών συνεδριών S .

1. GenerateRandomData()

Όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο κάθε ιπτάμενος έχει κάποια συγκεκριμένα χαρακτηριστικά τα οποία θεωρούνται δεδομένα για την επίλυση του προβλήματος. Με την ρουτίνα GenerateRandomData() εισάγουμε τα δεδομένα στον

κώδικα χωρίς όμως να ανταποκρίνονται στην πραγματικότητα αλλά αναπαράγονται από γεννήτριες τυχαίων αριθμών. Έτσι σαν δεδομένα εισόδου λαμβάνονται:

- η λίστα με τους ιπτάμενους, **N** στο σύνολο , που ανατεθούν στις εκπαιδευτικές συνεδρίες.
- Η λίστα με τις εκπαιδευτικές συνεδρίες, **S** στο σύνολο που πραγματοποιούνται κάθε μήνα.
- Η λήξη της πτητικής ικανότητας κάθε ιπτάμενου (**expiration**) που δηλώνεται με τις τιμές -1,0,1. Με την τιμή -1 αναφερόμαστε στον παρόντα μήνα, με 0 στον επόμενο μήνα και με 1 στον μεθεπόμενο μήνα.

```
for (i = 0; i <= N - 1; i++) {
    random = (double)(rand()) / (double)RAND_MAX; //
    random in (0,1)
    random = 3 * random;
    j = ceil(random);
    if (j == 1) Expiration[i] = -1;
    elseif (j == 2) Expiration[i] = 0;
    elseif (j == 3) Expiration[i] = 1;
}
```

Εικόνα 6 1: Ο κώδικας για το Expiration

- Η αρχαιότητα (**seniority**) του κάθε ιπτάμενου που δηλώνεται με ένα ακέραιο αριθμό. Με την τιμή 1 αναφερόμαστε στον παλαιότερο ιπτάμενο και κάθε φορά η τιμή αυξάνεται κατά μια μονάδα. Με τιμή ίση με N αναφερόμαστε στο νεότερο.

```
for (i = 0; i <= N - 1; i++)
    SeniorityUsed[i] = false;
for (i = 0; i <= N - 1; i++) {
    random = (double)(rand()) / (double)RAND_MAX;
    random = random * N;
    Seniority[i] = ceil(random);
    while (SeniorityUsed[Seniority[i] - 1] == true) {
        random = (double)(rand()) / (double)RAND_MAX;
        random = random * N;
        Seniority[i] = ceil(random);
    }
    SeniorityUsed[Seniority[i] - 1] = true;
}
```

Εικόνα 6 2: Ο κώδικας για το seniority

- Η θέση (**position**) του κάθε ιπτάμενου όπου με την τιμή 1 δηλώνεται ο κυβερνήτης (**FO**) ,με την τιμή 2 ο συγκυβερνήτης (**CP**), με την τιμή 3 οι αρχιφροντιστές (**purser**)και με την τιμή 4 οι αεροσυνοδοί (**FA**).

```

for (i = 0; i<= N - 1; i++) {
if (issim == false)
{
    random = (double)(rand())/(double)RAND_MAX;
    random = random * 4;
    Position[i] = ceil(random);
}
else//if IsSim == true mpaineistosim
{
    random = (double)(rand())/(double)RAND_MAX;
    random = random * 2;
    Position[i] = ceil(random);
}
}

```

Εικόνα 6 3: Ο κώδικας για το position

- Η λίστα με τις εφικτές συνεδρίες του κάθε ιπτάμενου (**IdSesValid**) δηλαδή εκείνες τις συνεδρίες που μπορεί να ανατεθεί ο καθένας.

```

for (i = 0; i<= N - 1; i++) {           // gia to N
for (j = 0; j <= S - 1; j++) {         //gia to S
random = (double)(rand()) / (double)RAND_MAX;           if
(random <= 0.75)
{
    IdSesValid[i][j] = true;
}

else
{
    IdSesValid[i][j] = false;
}
}
}
}

```

Εικόνα 6 4: Ο κώδικας για την λίστα των εφικτών συνεδριών

- Η λίστα με τις προτιμήσεις του κάθε ιπτάμενου (**priority**) όπου με την τιμή 1 δηλώνει την μεγαλύτερη προτίμηση του και μπορεί να δηλώσει μέχρι 8 συνεδρίες. Με την τιμή μηδέν δηλώνονται οι συνεδρίες στις οποίες δεν έχει προτίμηση.

```

for (i = 0; i <= N - 1; i++) {
for (j = 0; j <= S - 1; j++) {
if (IdSesValid[i][j] == true)
{
random = (double)(rand()) / (double)RAND_MAX;
random = random * 8;
IdSesPriority[i][j] = ceil(random);
if (j > 0)
{
for (k = 1; k <= j; k++) {
while (IdSesPriority[i][j - k] == IdSesPriority[i][j]) {
random = (double)(rand()) / (double)RAND_MAX;
random = random * 8;
IdSesPriority[i][j] = ceil(random);
}
}
}
}
else
{
IdSesPriority[i][j] = 0;
}
}
}
}

```

Εικόνα 6 5:0 κώδικας για την λίστα των προτιμήσεων

2. ReadData()

Αυτή η ρουτίνα δημιουργήθηκε έτσι ώστε να εισαγάγουμε χειροκίνητα πραγματικά δεδομένα εισόδου και όχι από γεννήτριες τυχαίων αριθμών. Χρησιμοποιώντας την μεταβλητή userandom καθορίζουμε ποια από τις ρουτίνες εισαγωγής δεδομένων θα επιλέξουμε.

3. init_problem()

Στην ρουτίνα `init_problem()` διαμορφώνουμε το περιβάλλον της CPLEX, με την εντολή `CPXopenCPLEX`, στο οποίο θα εργαστούμε για να αναπτύξουμε τον κώδικα. Αυτή η εντολή μας επιστρέφει έναν δείκτη (`env`) ο οποίος δείχνει το περιβάλλον που δημιουργήθηκε. Αμέσως μετά μέσα σε αυτό το περιβάλλον δημιουργούμε ένα πρόβλημα της CPLEX με τη εντολή `CPXcreatprob` η οποία όταν εκτελεστεί μας δίνει τον δείκτη (`lp`) ο οποίος δείχνει το πρόβλημα που μόλις δημιουργήσαμε. Το αρχικό

πρόβλημα που δημιουργούμε και καταγράφουμε είναι ένα πρόβλημα ελαχιστοποίησης με μηδενικό αριθμό περιορισμών και μεταβλητών απόφασης.

```
boolinit_problem()  
{  
  if (init_env(&env) == false)  
    returnfalse;  
  if (init_lp(env, &lp, "myprob") == false)  
    returnfalse;  
  returntrue;  
}
```

Εικόνα 6 6: Η ρουτίνα `unit_problem()`

4. `clock()`

Με την εντολή αυτή ορίζουμε το σημείο εκκίνησης της καταγραφής και της καταμέτρησης του χρόνου επίλυσης του προβλήματος.

5. `multiple_choice()`

Μέσα σε αυτήν την ρουτίνα καλούμε δυο άλλες ξεχωριστά οι οποίες χρησιμεύουν στη δημιουργία της αντικειμενικής συνάρτησης και των αρχικών περιορισμών ανάθεσης και μη ανάθεσης.

➤ `AddobjVars()`

Με αυτήν την συνάρτηση εισαγάγουμε στο κώδικα τις μεταβλητές απόφασης που αφορούν την ανάθεση x_{ij} και την μη ανάθεση y_j των ιπτάμενων στις εκπαιδευτικές συνεδρίες. Για κάθε ιπτάμενο σαρώνουμε όλη την λίστα με τις εκπαιδευτικές συνεδρίες του και βρίσκουμε αυτές για τον οποίο είναι εφικτές. Για κάθε εφικτή εκπαιδευτική συνέδρια εισαγάγουμε μια μεταβλητή απόφασης με την εντολή `CPXnewcols`. Με την εντολή αυτή καθορίζουμε τον συντελεστή της μεταβλητής που αρχικά είναι μηδέν. Επίσης καθορίζουμε τα όρια της μεταβλητής που στο πρόβλημα μας θα είναι από 0 μέχρι 1, τον τύπο της μεταβλητής όπως δυαδική που την χρησιμοποιούμε εδώ αλλά και τέλος το άνομα που θέλουμε να δώσουμε στην μεταβλητή. Αυτό που κάνουμε στη συνέχεια είναι να ορίσουμε έναν αύξοντα αριθμό σε κάθε μεταβλητή ο οποίος θα είναι και ο δείκτης που θα έχει μέσα στην αντικειμενική συνάρτηση. Αυτό

το πετυχαίνουμε καλώντας τη ρουτίνα *CPXgetnumcols* η οποία μας δίνει πρόσβαση στον αριθμό των μεταβλητών που έχουν εισαχθεί στο πρόβλημα. Έτσι για κάθε ιπτάμενος που έχει εφικτές συνεδρίες και η λήξη της πτητικής του ικανότητας είναι τον παρόντα μήνα αλλά και έχει δηλώσει προτιμήσεις δημιουργείται μια αντίστοιχη μεταβλητή απόφασης για την ανάθεση του x_{ij} με τον αντίστοιχο δείκτη της. Στην αντίθετη περίπτωση αν δεν ισχύουν οι παραπάνω περιορισμοί δημιουργείται μια μεταβλητή απόφασης για την μη ανάθεση y_i με τον αντίστοιχο δείκτη της.

```

boolAddobjVars()
{
doubleobj, lb, ub;
charxctype, colname;
inti,j, ccnt;

obj = 0;
lb = 0;
ub = 1;
xctype = 'B';
ccnt = 1;
colname = NULL;

for (i=0; i<= N-1; i++){
    if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xctype, NULL) == false) returnfalse;
    Cplexidx[i][0] = CPXgetnumcols(env, lp) - 1;
    for (j=0; j<= S-1; j++){
        if (IdSesValid[i][j] == false) continue;
        if (Expiration[i] == 1 &&IdSesPriority[i][j] <= 0 ) continue;
        if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xctype, NULL) == false) returnfalse;
        Cplexidx[i][j+1] = CPXgetnumcols(env, lp) - 1;
    }
}

returntrue;
}

```

Εικόνα 6 7:Δημιουργία μεταβλητών απόφασης

➤ addconstrVars()

Με την ρουτίνα αυτή εισάγουμε τους περιορισμούς ανάθεσης των ιπτάμενων. Έχοντας εισάγει τις μεταβλητές απόφασης για την ανάθεση και την μη ανάθεση δομούμε τους περιορισμούς ανάθεσης για τον κάθε ιπτάμενο. Αρχικά δημιουργούμε τους περιορισμό με την εντολή *CPXnewrows*. Αυτή η εντολή έχει σαν παραμέτρους το περιβάλλον της *CPLEX* (*env*), το πρόβλημα της *CPLEX* (*lp*), το σύνολο των περιορισμών που προστίθεται (εδώ έχουμε 1), το δεξί μέρος του περιορισμού (*rhs=1*), το αριθμητικό τελεστή (*sense=B*) και το όνομα του περιορισμού. Η *CPXnewrows* λειτουργεί σε συνδυασμό με την εντολή *Chgcoef* η οποία δίνει στον περιορισμό το σύνολο των μεταβλητών που υπάρχουν στο αριστερό μέλος του περιορισμού. Έτσι για κάθε ιπτάμενο θα

δημιουργηθεί ένας περιορισμός ανάθεσης με τις αντίστοιχες μεταβλητές απόφασης για την ανάθεση και την μη ανάθεση.

```
booladdconstrVars()
{
doublerhs, obj, ub, lb ;
char sense, *rowname, colname, xtype;
inti, j, ccnt;

obj = 0;
lb = 0;
ub = 1;
xctype = 'B';
ccnt = 1;
colname = NULL;
rowname = (char*)malloc(10*sizeof(char));
rhs = 1;
sense = 'E';
//newvalue=1

for (i=0; i<=N-1; i++){
if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xctype, NULL) == false) returnfalse;
Cplexidx[i][0] = CPXgetnumcols(env, lp) - 1;
sprintf(rowname, "Assign%d" , i);
if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
Chgcoef(env, lp,i ,Cplexidx[i][0] ,1);
for (j=0; j<=S-1; j++){
if (IdSesValid[i][j] == false) continue;
if (Expiration[i] == 1 &&IdSesPriority[i][j] <= 0 ) continue;
if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xctype, NULL)==false)return false;
Cplexidx[i][j+1] = CPXgetnumcols(env, lp) - 1;
Chgcoef(env, lp,i ,Cplexidx[i][j+1] ,1);

}

}
free(rowname);
returntrue;
}
```

Εικόνα 6 8:Δημιουργία περιορισμών ανάθεσης

6. Sim_capacity()

Με αυτήν την ρουτίνα εισάγουμε τους περιορισμούς χωρητικότητας για κάθε συνεδρία μόνο για προσομοιωτή. Όπως έχουμε αναφέρει στον προσομοιωτή υπάρχουν μόνο 2 θέσεις κενές και με αυτούς τους περιορισμούς εξασφαλίζουμε ότι το πολύ ένας κυβερνήτης ή συγκυβερνήτης θα ανατεθεί σε κάθε εκπαιδευτική συνεδρία. Η διαδικασία που ακολουθούμε είναι να εντοπίσουμε και να μετρήσουμε για κάθε συνεδρία τους κυβερνήτες (position=1) οι οποίοι έχουν εφικτή την εκάστοτε εκπαιδευτική συνεδρία και να καταγράψουμε τις αντίστοιχες μεταβλητές ανάθεσής τους. Έτσι με την εντολή CPXnewrows έχοντας ως αριθμητικό τελεστή <= και δεξί μέλος ίσο με 1 εισάγουμε για κάθε εκπαιδευτική συνεδρία τον αντίστοιχο περιορισμό χωρητικότητας. Η ίδια διαδικασία εκτελείται ξανά και για τους

συγκυβερνήτες (position=2) όπου προχωράμε και εδώ στην εισαγωγή των αντίστοιχων περιορισμών για κάθε συνεδρία. Ο έλεγχος επαναλαμβάνεται για όλες τις συνεδρίες της λίστας μας.

```
boolSim_capacity()
{
    if (issim == false) returntrue;
    doublerhs;
    char sense,*rowname, colname;
    inti, j;

    rhs = 1;
    sense = 'L';
    colname = NULL;
    rowname = (char*)malloc(10*sizeof(char));

    for (j=0; j<=S-1; j++){
        sprintf(rowname, "sim1_cap%d",j);
        if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname)==false)returnfalse;
        for (i=0; i<=N-1; i++){
            if (Position[i] != 1) continue;
            if ( Cplexidx[i][j+1] == -1) continue;
            Chgcoef(env, lp,CPXgetnumrows(env,lp)-1 , Cplexidx[i][j+1] ,1);
        }
    }

    for (j=0; j<=S-1; j++){
        sprintf(rowname, "sim2_cap%d", j);
        if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
        for (i=0; i<=N-1; i++){
            if (Position[i] != 2) continue;
            if ( Cplexidx[i][j+1] == -1) continue;
            Chgcoef(env, lp,CPXgetnumrows(env,lp)-1 , Cplexidx[i][j+1] ,1);
        }
    }
    free(rowname);
    returntrue;
}
```

Εικόνα 6 9: Οκώδικας για το sim_capacity

7. Class_capacity

Στην περίπτωση των τμημάτων θεωρίας οι περιορισμοί χωρητικότητα είναι διαφορετικοί. Υπάρχουν περιορισμοί για την συνολική χωρητικότητα, για τον αριθμό του πληρώματος πιλοτήριου και καμπίνας αλλά και για το αριθμό των ιπτάμενων για κάθε ειδικότητα ξεχωριστά.

Αρχικά με αυτήν την υπορουτίνα εισάγουμε τους περιορισμούς που μας εξασφαλίζουν ότι ο συνολικός αριθμός ιπτάμενων που θα ανατεθούν να μην υπερβαίνουν τις διαθέσιμες θέσεις. Δημιουργούμε λοιπόν έναν περιορισμό με την εντολή CPXnewrows η οποία έχει ως αριθμητικό τελεστή <= και δεξί μέλος ίσο με N/S*1.5 και ελέγχουμε ποιοι πιλότοι την έχουν δηλώσει ως εφικτή και εισάγουμε

την αντίστοιχη μεταβλητή στον περιορισμό. Αυτή η διακόςια επαναλαμβάνεται για κάθε εκπαιδευτική συνεδρία ξεχωριστά.

```
x= N*1.5/S;
del = false;

rhs=ceil(x);
for (j=0; j<=S-1; j++){
    sprintf(rowname, "tot_cap%d", j);
    if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname)==false)returnfalse;
    for (i=0; i<=N-1; i++){
        if ( Cplexidx[i][j+1] == -1) continue;
        del = true;
        Chgcoef(env, lp, CPXgetnumrows(env,lp)-1 , Cplexidx[i][j+1] ,1);
    }
    if (del == false) {
        if (DelRows (env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumrows(env,lp)-1) == false)
            returnfalse;
    }
}
}
```

Εικόνα 6 10: Totalcapacity

Αμέσως μετά δημιουργούμε τους περιορισμούς χωρητικότητας που μας εξασφαλίζουν ότι σε κάθε συνεδρία θα ανατεθούν τόσοι υπάλληλοι καμπίνας και τόσοι υπάλληλοι πιλοτήριου όσες είναι οι διαθέσιμες θέσεις που δίνονται σε κάθε συνεδρία για την αντίστοιχη κατηγορία. Βρίσκουμε και μετράμε δηλαδή για κάθε μια συνεδρία τους εκπαιδευόμενους που έχουν θέση κυβερνήτη (**position=1**) ή συγκυβερνήτη (**position=2**) και που έχουν εφικτή την εκάστοτε συνεδρία και καταγράφουμε τις αντίστοιχες μεταβλητές ανάθεσης. Έπειτα με την εντολή CPXnewrows με αριθμητικό τελεστή \leq και δεξί μέλος $0.75*(N/S*1.5)$ αυτή την φορά ,εισάγουμε για κάθε συνεδρία τον αντίστοιχο περιορισμό χωρητικότητας. Η ίδια διαδικασία επαναλαμβάνεται ξανά με τις ίδιες εντολές και για τους υπάλληλους καμπίνας με την διάφορα ότι τώρα βρίσκουμε και μετράμε τις μεταβλητές ανάθεσης για τους ιπτάμενους με θέση αρχιφροντιστη (**position=3**) ή αεροσυνοδού (**position=4**).

```

// cockpit capacity
rhs = ceil(0.75 * x);
for (j=0; j<= S-1; j++){
    del = false;
    sprintf(rowname, "Cockpit%d", j);
    if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
    for (i=0; i<=N-1; i++){
        if (Cplexidx[i][j+1] == -1) continue;
        if (Position[i] != 1 && Position[i] != 2) continue;
        del = true;
        Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[i][j+1], 1 );
    }
    if (del == false) {
        if (DelRows (env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumrows(env,lp)-1) == false)
returnfalse;
    }
}
}

```

Εικόνα 6 11:Ο κώδικας για cockpitcapacity, είναι ίδιος και για cabin μεposition =3 ή 4

Τέλος δημιουργούμε και τους περιορισμούς χωρητικότητας για κάθε ειδικότητα ξεχωριστά οι οποίοι μας εξασφαλίζουν ότι δεν θα μπουν περισσότεροι ιπτάμενοι από κάθε ειδικότητας από όσες είναι οι διαθέσιμες θέσεις. Για παράδειγμα σε κάθε συνεδρία βρίσκουμε και μετράμε τους κυβερνήτες (**position=1**) που έχουν εφικτή την εκάστοτε συνεδρία και καταγράφουμε τις αντίστοιχες μεταβλητές ανάθεσης τους στα τμήματα θεωρίας. Πάλι με την εντολή CPXnewrows με τον ίδιο αριθμητικό τελεστή αλλά με δεξί μέλος και πάλι διαφορετικό και ίσο με $0.35*(N/S*1.5)$ δημιουργούμε για κάθε εκπαιδευτική συνεδρία ξεχωριστά τους αντίστοιχους περιορισμούς για την χωρητικότητα. Εφόσον ολοκληρωθεί η διαδικασία για τους κυβερνήτες εκτελείτε ξανά η ίδια και για τις άλλες 3 ειδικότητες χρησιμοποιώντας τις ίδιες εντολές για την δημιουργία των περιορισμών (**position=2, 3, 4**)

```

rhs = ceil(0.35 * x); //giakatheenaxorista
//cp
for (j=0; j<= S-1; j++){
    del = false;
    sprintf(rowname, "CP_cap%d", j);
    if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
    for (i=0; i<=N-1; i++){
        if (Cplexidx[i][j+1] == -1) continue;
        if (Position[i] != 1) continue;
        del = true;
        Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[i][j+1], 1 );
    }
    if (del == false) {
        if (DelRows (env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumrows(env,lp)-1) == false)
returnfalse;
    }
}
}

```

Εικόνα 6 12: Ο κώδικας για cp,fo,purser,facapacity . Είναι ο ίδιος και χρησιμοποιούμε διαφορετικό position

8,9,10. `fix_expiration(intexp)`

Χρησιμοποιώντας αυτή την ρουτίνα προσπαθώ να ελαχιστοποιήσω τον αριθμό των ιπτάμενων με `expiration -1 ή 0 ή 1` που θα μείνουν εκτός ,δεν ανατεθούν δηλαδή σε καμία εκπαιδευτική συνεδρία. Ορίζω ένα δείκτη `exp` με τιμές `-1,0,1` ο οποίος καθορίζει ποια είναι λήξη της πτητικής ικανότητας που εξετάζει η ρουτίνα. Εξετάζοντας αρχικά την περίπτωση όπου η λήξη της πτητικής ικανότητας των ιπτάμενων είναι το παρόντα μήνα, το πρώτο βήμα που κάνουμε είναι να εισάγουμε μια νέα μεταβλητή απόφασης που μετράει πόσοι εκπαιδευόμενοι θα μείνουν εκτός από τις συνεδρίες. Την μεταβλητή αυτή την εισάγουμε με την εντολή `CPXnewcols` στην οποία δηλώνουμε τον τύπο της μεταβλητής η οποία είναι `integer`, το κάτω όριο ίσο με `0`, το πάνω όριο ίσο με `CPX_INFBOUND` και με συντελεστή στην αντικειμενική συνάρτηση ίσο με `1000000`. Αμέσως μετά για κάθε ιπτάμενο με `exp=-1` βρίσκουμε τις μεταβλητές μη ανάθεσης. Έτσι δημιουργούμε ένα καινούριο περιορισμό ο οποίος μας δηλώνει ότι η νέα μεταβλητή που εισάγαμε πρέπει να είναι ίση με το άθροισμα των μεταβλητών απόφασης για την μη ανάθεση. Αξίζει να σημειωθεί ότι στην περίπτωση όπου δεν βρεθεί κανένας ιπτάμενος με `exp=-1` τότε θα πρέπει να σβήσουμε την μεταβλητή και τον περιορισμό που εισάγαμε με τις εντολές `CPXdelcols` και `CPXdelrows`. Στη συνέχεια αφού δημιουργήσαμε τον περιορισμό θα λύσουμε το πρόβλημα με την εντολή `CPXmipopt`, η οποία είναι μια εντολή που λύνει ένα πρόβλημα μεικτού ακέραιου προγραμματισμού. Αφού λύσουμε το πρόβλημα καλούμε την εντολή `CPXgetx` η οποία μας δίνει την λύση, δηλαδή τον βέλτιστο ελάχιστο αριθμό ιπτάμενων που θα μείνουν εκτός. Έτσι έχοντας πάρει την βέλτιστη λύση εισάγουμε και ένα δεύτερο περιορισμό όπου στην τελευταία μεταβλητή απόφασης που εισάγαμε δίνουμε την τιμή που μας έδωσε η λύση. Με αυτή τη διαδικασία λοιπόν έχουμε κατοχυρώσει και για τα επόμενα στάδια της επίλυσης πόσοι ιπτάμενοι με `exp=-1` δεν θα ανατεθούν σε καμία εκπαιδευτική συνεδρία. Η ίδια ακριβώς μέθοδος θα επαναληφθεί και για λήξη πτητικής ικανότητας τον επόμενο μήνα με `exp=0` αλλά και για τον μεθεπόμενο μήνα με `exp=1`.

```

bool fix_expiration(int exp)
{
    double rhs, obj, ub, lb, sol;
    char sense, *rowname, colname, xtype;
    int i, ccnt, status;
    bool delvar;

    obj = 1000000;
    lb = 0;
    ub = CPX_INFBOUND;
    xtype = 'I';
    ccnt = 1;
    colname = NULL;
    rowname = (char*)malloc(10*sizeof(char));
    rhs = 0;
    sense = 'E';
    delvar = false;

    if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xtype, NULL) == false) return false;
    sprintf(rowname, "fix_exp%d", exp);
    if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) return false;
    for (i=0; i<=N-1; i++){
        if (Expiration[i] != exp) continue;
        delvar = true;
        Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[i][0], -1);
    }
    if (delvar == false) {
        if (DelCols (env, lp, CPXgetnumcols(env, lp) -1, CPXgetnumcols(env, lp) -1) == false) return false;
        if (DelRows (env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumrows(env,lp)-1) == false) return false;
    }
    Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, CPXgetnumcols(env, lp) - 1, 1);
    if (Mipopt(env, lp) == false) return false;
    status = CPXgetx (env,lp, &sol, CPXgetnumcols(env, lp)-1, CPXgetnumcols(env, lp)-1);
    if (status != 0){
        print_error("CPXgetx");
        return false;
    }

    rhs = sol;
    sprintf(rowname, "fix_sol%d", exp);
    if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) return false;
    Chgcoef(env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumcols(env, lp) - 1, 1);
    if (delvar == false) {
        if (DelRows (env, lp, CPXgetnumrows(env,lp)-1, CPXgetnumrows(env,lp)-1) == false) return false;
    }
    Chgcoef(env, lp, -1, CPXgetnumcols(env, lp) - 1, 0);

    free(rowname);
    return true;
}

```

Εικόνα 6 13: Ο κώδικας για το fix_expiration

11. Fix_bids()

Σε αυτή την ρουτίνα θα προχωρήσουμε στη εισαγωγή ενός περιορισμού που θα μας δίνει την ανάθεση των ιπτάμενων στις κατάλληλες εκπαιδευτικές συνεδρίες. Θεωρώντας δεδομένο τον αριθμό των ιπτάμενων που θα μείνουν εκτός αυτό που θέλουμε να πετύχουμε είναι πως κάθε ιπτάμενος ανάλογα με την αρχαιότητα του (πρώτα ο πιο παλιός) θα ανατεθεί στην μεγαλύτερη προτίμηση του από αυτές που

είναι εφικτό να πάρει. Αρχικά κατατάσσω τους ιπτάμενους από τον παλαιότερο προς τον νεότερο αλλά και τις προτιμήσεις του από την μεγαλύτερη προς την μικρότερη στην μικρότερη. Έτσι ξεκινώντας από τον παλαιότερο βρίσκουμε τις συνεδρίες οι οποίες είναι εφικτές για τον εκάστοτε ιπτάμενο και ελέγχουμε τις προτιμήσεις του. Για την μεγαλύτερη προτίμηση του λύνω το πρόβλημα με CPXmipopt και παίρνω μια λύση όπου με την εντολή **CPXgetstate** ελέγχω αν είναι εφικτή για το πρόβλημα μας. Αν η τιμή του δείκτη **solstat** είναι διάφορη του 103 τότε έχουμε πάρει μια εφικτή λύση και θα εισαχτεί ένα καινούριος περιορισμός με την αντίστοιχη μεταβλητή απόφασης για την ανάθεση που θα ταιριάζει στην λύση. Στην περίπτωση όμως που το **solstat** είναι ίσο με 103 τότε δεν θα εισαχθεί ο περιορισμός και θα πάμε να ελέγξουμε και τις άλλες προτιμήσεις του. Η διαδικασία αυτή θα εκτελεστεί επαναληπτική για κάθε ιπτάμενο κατά παλαιότητα.

```

bool fix_bids()
{
int m, i, j, k, max_prior[N], Sort_seniority[N];
for (m=1; m <=N; m++){
    for (i=0; i<=N-1; i++){
        if (Seniority[i] != m) continue;
        Sort_seniority[m-1] = i;
    }
}

double rhs;
char sense, *rowname;
int solstat;
rhs = 1;
sense = 'E';
rowname = (char*)malloc(10*sizeof(char));

for (i=0; i<=N-1; i++){
    max_prior[i] = 0;
    for (j=0; j<=S-1; j++){
        if (IdSesPriority[i][j] >= max_prior[i]) {
            max_prior[i] = IdSesPriority[i][j];
        }
    }
}

for (i=0; i<=N-1; i++){ //giakatheiptameno
    for (k=1; k<=max_prior[Sort_seniority[i]]; k++){ //protereeotita
        for (j=0; j<=S-1; j++){ //session
            if (IdSesPriority[Sort_seniority[i]][j] == 0) continue;
            if (IdSesPriority[Sort_seniority[i]][j] != k) continue;
            sprintf(rowname, "fix_bids%d", i);
            if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) return false;
            Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[Sort_seniority[i]][j+1], 1);
        }
        if (Mipopt(env, lp) == false) return false;
        solstat = CPXgetstat(env, lp);
        if (solstat == 103) { //mh efikto
            break;
        }
    }
    if (solstat != 103) break;
}
}
free(rowname);
return true;
}

```

Εικόνα 6 14: Ο κώδικας για το fix_bids

12. Class_balance()

Αυτή η ρουτίνα αναφέρεται μόνο σε συνεδρίες που αφορούν τα τμήματα θεωρίας. Με τη συνάρτηση `class_balance` εισάγουμε στο πρόβλημα περιορισμούς με τους οποίους προσπαθούμε όσο είναι δυνατό να αναθέτουμε σε κάθε συνεδρία τον ίδιο αριθμό εκπαιδευόμενων. Έχοντας βρει δηλαδή από την συνάρτηση `fix_expiration()` τον αριθμό των ιπτάμενων που θα μείνουν εκτός τώρα θέλω να τους αναθέσω σε συνεδρίες με τέτοιο τρόπο έτσι ώστε να υπάρχει ισορροπία μεταξύ των τμημάτων. Στον συγκεκριμένο πρόβλημα θεωρούμε δεδομένο ότι οι συνεδρίες έχουν ίδια χαρακτηριστικά και δεν διαφέρουν η μια από την άλλη και συνεπώς ορίζω για κάθε εκπαιδευτικό τμήμα μια μέση τιμή ($avg=N/S$) για το σύνολο των ατόμων που μπορούν να εντεθούν σε αυτή.

Αρχικά εισάγω μια νέα μεταβλητή απόφασης με την εντολή `CPXnewcols` η οποία μετράει την μέγιστη απόκλιση από το άνω και το κάτω όριο. Αυτή η μεταβλητή είναι ακέραια, το κάτω όριο της είναι το 0, το πάνω όριο είναι `CPX_INFBOUND` και ο συντελεστής της στην αντικειμενική συνάρτηση ίσος με 1000. Αφού δημιουργήσαμε αυτήν την μεταβλητή στην συνέχεια για κάθε συνεδρία ξεχωριστά βρίσκουμε τους ιπτάμενους που θα γίνει η ανάθεση τους σ' αυτήν και αθροίζουμε τις αντίστοιχες μεταβλητές απόφασης που αφορούν την ανάθεση. Αφού έχουμε βρει αυτό το άθροισμα εισάγουμε έναν νέο περιορισμό για κάθε συνεδρία. Αυτός ο περιορισμός μας λέει ότι η καινούρια μεταβλητή που εισάγαμε δηλαδή η μέγιστη απόκλιση θα είναι μεγαλύτερη ίση (\geq) από το άθροισμα των ιπτάμενων για τους οποίους έχει γίνει ανάθεση μείον την μέση τιμή του συνόλου των ατόμων που μπορούν να ανατεθούν στο τμήμα εκπαίδευσης. Αυτόν ο περιορισμός αναφέρεται στον άνω όριο την μέγιστης απόκλισης (**upperbound**).

Η ίδια επαναληπτική διαδικασία θα εκτελεστεί και για το κάτω όριο της μέγιστης απόκλισης (**lowerbound**). Θα εισάγουμε ξανά για κάθε εκπαιδευτική συνεδρία των τμημάτων θεωρίας ένα περιορισμό που θα δηλώνει ότι η νέα μεταβλητή απόφασης, δηλαδή η μέγιστη απόκλιση, θα είναι μεγαλύτερη ίση (\geq) από την μέση τιμή του συνόλου των ιπτάμενων που μπορούν να ανατεθούν μείον το άθροισμα των ιπτάμενων που έχει γίνει η ανάθεση τους σε μια συγκεκριμένη συνεδρία. Συνεπώς αυτή η επαναληπτική διαδικασία με τους αντίστοιχους περιορισμούς δημιουργήθηκε έτσι ώστε να υπάρχουν σχετικά ισομερή τμήματα εκπαίδευσης.

```

boolClass_balance()
{
if (issim == true) returntrue;
doublerhs, obj, ub, lb, avg;
char sense, *rowname, colname, xtype;
inti, j ,ccnt;

obj = 1000;
lb = 0;
ub = CPX_INFBOUND;
xctype = 'I';
ccnt = 1;
colname = NULL;
rowname = (char*)malloc(10*sizeof(char));
avg = N/S;
sense = 'G';

for (j=0; j<=S-1; j++){
if (NewCols(env, lp, ccnt, &obj, &lb, &ub, &xctype, NULL) == false) returnfalse;
rhs = -avg-1;
sprintf(rowname, "up_bal%d", j);
if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, CPXgetnumcols(env, lp) - 1, 1);
for (i=0; i<=N-1; i++){
if (Cplexidx[i][j+1] == -1) continue;
Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[i][j+1], -1);
}

rhs = avg-1;
sprintf(rowname, "low_bal%d", j);
if (Newrows(env, lp, 1, &rhs, &sense, NULL, &rowname) == false) returnfalse;
Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, CPXgetnumcols(env, lp) - 1, 1);
for (i=0; i<=N-1; i++){
if (Cplexidx[i][j+1] == -1) continue;
Chgcoef(env, lp, CPXgetnumrows(env, lp) - 1, Cplexidx[i][j+1], 1);
}
}
free(rowname);
returntrue;
}

```

Εικόνα 6 15: Ο κώδικας για το class_balance

13. evaluate_results()

Με αυτήν την ρουτίνα επιλύουμε το πρόγραμμα που δομήθηκε από τις παραπάνω ρουτίνες και βλέπουμε τα αποτελέσματα που προκύπτουν. Για να φτάσουμε στην λύση πρώτα καλούμε την εντολή **CPXmipopt**, η οποία χρησιμοποιείται για να μας λύσει το πρόβλημα του Μεικτού Ακεραίου Προγραμματισμού. Αφού έχουμε λύσει το πρόβλημα στη συνέχεια καλούμε την εντολή **CPXgetstat**, η οποία μας δίνει πρόσβαση στη λύση και βλέπουμε αν είναι εφικτή ή όχι. Αφού εκτελέσουμε αυτές τις εντολές στη συνέχεια με την βοήθεια της εντολής **CPXgetx**, η οποία προσεγγίζει τις τιμές της λύσης, εκτυπώνουμε σε ένα αρχείο τα αποτελέσματα που παρουσιάζουν τις συνεδρίες στις οποίες ανατίθενται οι ιπτάμενοι και ποιοι ιπτάμενοι δεν ανατίθενται για εκπαίδευση (αν υπάρχουν).


```

boolevaluate_results()
{
intsolstat, num_var, i , j,status;
double *sol;

if (Mipopt(env, lp) == false) returnfalse;

solstat = CPXgetstat(env, lp);
if (solstat == 103) { //mhefikto
    print_error("Problem infeasible");
}

num_var = CPXgetnumcols(env, lp);
sol = (double*)malloc(num_var *sizeof(double));
status = CPXgetx (env,lp, sol , 0, num_var-1);
if (status != 0){
    print_error("CPXgetx");
returnfalse;
}

for (i=0; i<=N-1; i++){
    if (sol[Cplexidx[i][0]] > 0.5) continue;
    printf(" ID%d remains unassign\n",i);
    for (j=1; j<=S; j++){
        if (sol[Cplexidx[i][j]]> 0.5) continue;
        printf("ID%d is assign to session %d\n",i,j);
    }
}
free(sol);
returntrue;
}

```

Εικόνα 6 16: Κ επίλυση και τα αποτελέσματα

14. Print Problem(env, lp, "myprob.lp", NULL)

Σε αυτήν την συνάρτηση καλούμε την ρουτίνα **CPXwriteprob()**, η οποία χρησιμοποιείται για να γράψει το αντικείμενο των προβλημάτων CPLEX LP σε ένα αρχείο σε μια συγκεκριμένη μορφή. Όταν χρησιμοποιείται αυτή η ρουτίνα, το τρέχον πρόβλημα είναι γραμμένο σε ένα αρχείο στον υπολογιστή μας.

15. Clock()

Με την εντολή αυτή ορίζουμε το σημείο παύσης της καταγραφής και καταμέτρησης του χρόνου επίλυσης του προβλήματος.

16. FreeAll()

Στη τελευταία ρουτίνα που καλούμε εκτελώντας τη εντολή **CPXfreeprob** αφαιρούμε το πρόβλημα της CPLEX από το περιβάλλον της CPLEX και απελευθερώνουμε την αντίστοιχη μνήμη, που χρησιμοποιήθηκε κατά το στήσιμο και την εκτέλεση του κώδικα. Επίσης, με την εντολή **CPXcloseCPLEX** απελευθερώνουμε όλες τις δομές δεδομένων, που σχετίζονται με την CPLEX.

```
boolFreeAll()  
{  
    if ( FreeProb( env, &lp) == false) returnfalse;  
    if ( FreeEnv(&env) == false) returnfalse;  
    returntrue;  
}
```

Εικόνα 6 17: Το κλείσιμο του προβλήματος

Εάν μετά την εκτέλεση του κώδικα δεν μας δίνεται καμία εφικτή λύση τότε το πρόγραμμα τερματίζει και τυπώνει ένα μήνυμα στην οθόνη ("**Problem infeasible**"), που μας ενημερώνει ότι το πρόβλημα μας δεν είναι εφικτό

7. ΕΦΑΡΜΟΓΗ ΤΟΥ ΚΩΔΙΚΑ (ΠΕΙΡΑΜΑΤΑ)

7.1 Πρώτο πείραμα

Το πρώτο παράδειγμα αφορά την ανάθεση δεκαεπτά (**N=17**) ιπτάμενων σε έξι (**S=6**) εκπαιδευτικές συνεδρίες. Στην αρχή παρουσιάζονται τα χαρακτηριστικά που έχει κάθε ιπτάμενος αλλά και η χωρητικότητα της κάθε συνεδρίας.

N	EXPIRATION	SENIORITY	POSITION
1	-1	7	1 (CP)
2	0	8	2(FO)
3	-1	15	1 (CP)
4	1	2	1 (CP)
5	-1	16	3 (PURSER)
6	-1	11	2 (FO)
7	-1	4	4 (FA)
8	0	6	2 (FO)
9	0	12	3 (PURSER)
10	1	13	1 (CP)
11	-1	14	4 (FA)
12	1	3	3 (PURSER)
13	-1	1	2 (FO)
14	0	9	2 (FO)
15	0	17	1 (CP)
16	-1	10	1 (CP)
17	-1	5	3 (PURSER)

Πίνακας 7 1: Τα χαρακτηριστικά των ιπτάμενων παράδειγμα 1

S N	1	2	3	3	5	6	ΣΥΝΟΛΟ ΑΦΙΚΤΩΝ ΣΥΝΕΔΡΙΩΝ
1	1	3	0	2	4	5	3
2	0	4	2	3	1	0	3
3	1	0	4	0	2	3	5
4	3	0	2	1	0	0	3
5	0	0	3	2	0	1	3
6	1	0	2	4	0	3	4
7	2	0	1	3	4	5	3
8	1	4	6	5	3	2	1
9	6	4	3	2	1	5	1
10	3	4	5	1	2	6	2
11	0	1	0	0	0	0	3
12	2	1	0	3	0	0	1
13	1	0	0	0	2	3	2
14	4	1	0	2	3	0	4
15	1	0	0	0	0	2	6
16	1	4	6	2	5	3	1
17	0	1	3	2	0	4	2

Πίνακας 7 2: Λίστα των προτιμήσεων των ιπτάμενων και το σύνολο των εφικτών συνεδριών παράδειγμα 1

	ΓΙΑ ΚΑΘΕ ΣΥΝΕΔΡΙΑ
ΣΥΝΟΛΙΚΗ ΧΩΡΗΤΙΚΟΤΗΤΑ	5
ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΑΜΠΙΝΑΣ	4
ΧΩΡΗΤΙΚΟΤΗΤΑ ΠΙΛΟΤΗΡΙΟΥ	4

ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΥΒΕΡΝΗΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΣΥΓΚΥΒΕΡΝΗΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΡΧΙΦΡΟΝΤΙΣΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΕΡΟΣΥΝΟΔΩΝ	2

Πίνακας 7 3: Η χωρητικότητα για κάθε συνεδρία παράδειγμα 1

Αφού τρέξαμε τον κώδικα για το παράδειγμα 1 πήραμε την παρακάτω αντικειμενική συνάρτηση και τους αντίστοιχους περιορισμούς.

✓ Η αντικειμενική συνάρτηση είναι:

Minimize

obj: $0 x_1 + 0 x_2 + 0 x_3 + 0 x_4 + 0 x_5 + 0 x_6 + 0 x_7 + 0 x_8 + 0 x_9 + 0 x_{10} + 0 x_{11} + 0 x_{12} + 0 x_{13} + 0 x_{14} + 0 x_{15} + 0 x_{16} + 0 x_{17} + 0 x_{18} + 0 x_{19} + 0 x_{20} + 0 x_{21} + 0 x_{22} + 0 x_{23} + 0 x_{24} + 0 x_{25} + 0 x_{26} + 0 x_{27} + 0 x_{28} + 0 x_{29} + 0 x_{30} + 0 x_{31} + 0 x_{32} + 0 x_{33} + 0 x_{34} + 0 x_{35} + 0 x_{36} + 0 x_{37} + 0 x_{38} + 0 x_{39} + 0 x_{40} + 0 x_{41} + 0 x_{42} + 0 x_{43} + 0 x_{44} + 0 x_{45} + 0 x_{46} + 0 x_{47} + 0 x_{48} + 0 x_{49} + 0 x_{50} + 0 x_{51} + 0 x_{52} + 0 x_{53} + 0 x_{54} + 0 x_{55} + 0 x_{56} + 0 x_{57} + 0 x_{58} + 0 x_{59} + 0 x_{60} + 0 x_{61} + 0 x_{62} + 1000 x_{128} + 1000 x_{129} + 1000 x_{130} + 1000 x_{131} + 1000 x_{132} + 1000 x_{133}$

- ✓ Οι περιορισμοί για την ανάθεση και την μην ανάθεση των υπτάμενων. Η πρώτη μεταβλητή απόφασης αφορά την μη ανάθεση.

Assign0#0: $x_{63} + x_{64} + x_{65} + x_{66} = 1$
Assign1#1: $x_{67} + x_{68} + x_{69} + x_{70} = 1$
Assign2#2: $x_{71} + x_{72} + x_{73} + x_{74} + x_{75} + x_{76} = 1$
Assign3#3: $x_{77} + x_{78} = 1$
Assign4#4: $x_{79} + x_{80} + x_{81} + x_{82} = 1$
Assign5#5: $x_{83} + x_{84} + x_{85} + x_{86} + x_{87} = 1$
Assign6#6: $x_{88} + x_{89} + x_{90} + x_{91} = 1$
Assign7#7: $x_{92} + x_{93} = 1$
Assign8#8: $x_{94} + x_{95} = 1$
Assign9#9: $x_{96} + x_{97} + x_{98} = 1$
Assign10#10: $x_{99} + x_{100} + x_{101} + x_{102} = 1$
Assign11#11: $x_{103} + x_{104} = 1$
Assign12#12: $x_{105} + x_{106} + x_{107} = 1$
Assign13#13: $x_{108} + x_{109} + x_{110} + x_{111} + x_{112} = 1$
Assign14#14: $x_{113} + x_{114} + x_{115} + x_{116} + x_{117} + x_{118} + x_{119} = 1$
Assign15#15: $x_{120} + x_{121} = 1$
Assign16#16: $x_{122} + x_{123} + x_{124} = 1$

Γιατους CP:

sim1_cap0#17: $x_{64} + x_{72} + x_{114} + x_{121} \leq 1$
sim1_cap1#18: $x_{65} + x_{73} + x_{115} \leq 1$
sim1_cap2#19: $x_{66} + x_{74} + x_{78} + x_{116} \leq 1$
sim1_cap3#20: $x_{75} + x_{97} + x_{117} \leq 1$
sim1_cap4#21: $x_{98} + x_{118} \leq 1$
sim1_cap5#22: $x_{76} + x_{119} \leq 1$

ΓιατουςFO:

sim2_cap0#23: $x_{68} + x_{84} + x_{93} \leq 1$
sim2_cap1#24: $x_{69} + x_{85} \leq 1$
sim2_cap2#25: $x_{70} + x_{86} + x_{106} + x_{109} \leq 1$
sim2_cap3#26: $x_{107} + x_{110} \leq 1$
sim2_cap4#27: $x_{87} + x_{111} \leq 1$
sim2_cap5#28: $x_{112} \leq 1$

✓ Οι περιορισμοί για την χωρητικότητα των τμημάτων θεωρίας.

➤ Συνολική χωρητικότητα

tot_cap0#17: $x_{64} + x_{68} + x_{72} + x_{80} + x_{84} + x_{93} + x_{100} + x_{114} + x_{121} \leq 5$

tot_cap1#18: $x_{65} + x_{69} + x_{73} + x_{85} + x_{89} + x_{101} + x_{104} + x_{115} + x_{123} \leq 5$

tot_cap2#19: $x_{66} + x_{70} + x_{74} + x_{78} + x_{81} + x_{86} + x_{90} + x_{106} + x_{109} + x_{116} + x_{124} \leq 5$

tot_cap3#20: $x_{75} + x_{91} + x_{95} + x_{97} + x_{107} + x_{110} + x_{117} \leq 5$

tot_cap4#21: $x_{82} + x_{87} + x_{98} + x_{111} + x_{118} \leq 5$

tot_cap5#22: $x_{76} + x_{102} + x_{112} + x_{119} \leq 5$

➤ Χωρητικότητα καμπίνας

cabin0#29: $x_{80} + x_{100} \leq 4$

cabin1#30: $x_{89} + x_{101} + x_{104} + x_{123} \leq 4$

cabin2#31: $x_{81} + x_{90} + x_{124} \leq 4$

cabin3#32: $x_{91} + x_{95} \leq 4$

cabin4#33: $x_{82} \leq 4$

cabin5#34: $x_{102} \leq 4$

➤ Χωρητικότητα πιλοτήριου

Cockpit0#23: $x_{64} + x_{68} + x_{72} + x_{84} + x_{93} + x_{114} + x_{121} \leq 4$

Cockpit1#24: $x_{65} + x_{69} + x_{73} + x_{85} + x_{115} \leq 4$

Cockpit2#25: $x_{66} + x_{70} + x_{74} + x_{78} + x_{86} + x_{106} + x_{109} + x_{116} \leq 4$

Cockpit3#26: $x_{75} + x_{97} + x_{107} + x_{110} + x_{117} \leq 4$

Cockpit4#27: $x_{87} + x_{98} + x_{111} + x_{118} \leq 4$

Cockpit5#28: $x_{76} + x_{112} + x_{119} \leq 4$

➤ Χωρητικότητα κυβερνητών

CP_cap0#35: $x_{64} + x_{72} + x_{114} + x_{121} \leq 2$

CP_cap1#36: $x_{65} + x_{73} + x_{115} \leq 2$

CP_cap2#37: $x_{66} + x_{74} + x_{78} + x_{116} \leq 2$

CP_cap3#38: $x_{75} + x_{97} + x_{117} \leq 2$

CP_cap4#39: $x_{98} + x_{118} \leq 2$

CP_cap5#40: $x_{76} + x_{119} \leq 2$

➤ Χωρητικότητα συγκυβερνητών

FO_cap0#41: $x_{68} + x_{84} + x_{93} \leq 2$

FO_cap1#42: $x_{69} + x_{85} \leq 2$

FO_cap2#43: $x_{70} + x_{86} + x_{106} + x_{109} \leq 2$

FO_cap3#44: $x_{107} + x_{110} \leq 2$

FO_cap4#45: $x_{87} + x_{111} \leq 2$

FO_cap5#46: $x_{112} \leq 2$

➤ Χωρητικότητα αρχιφροντιστών

purser0#47: $x_{80} \leq 2$

purser1#48: $x_{104} + x_{123} \leq 2$

purser2#49: $x_{81} + x_{124} \leq 2$

purser3#50: $x_{95} \leq 2$

purser4#51: $x_{82} \leq 2$

➤ Χωρητικότητα αεροσυνοδών

fl_att0#52: $x_{100} \leq 2$

fl_att1#53: $x_{89} + x_{101} \leq 2$

fl_att2#54: $x_{90} \leq 2$

fl_att3#55: $x_{91} \leq 2$

fl_att5#56: $x_{102} \leq 2$

fl_att5#56: $x_{102} \leq 2$

fl_att5#61: $x_{126} + x_{191} \leq 2$

✓ Περιορισμοί για την λήξη της πτητικής ικανότητας

➤ Expiration = -1

fix_expm1#57: $-x_{63} - x_{71} - x_{79} - x_{83} - x_{88} - x_{99} - x_{105} - x_{120} - x_{122} + x_{125} = 0$

fix_solm1#58: $x_{125} = 0$

➤ Expiration = 0

fix_exp0#59: $-x_{67} - x_{92} - x_{94} - x_{108} - x_{113} + x_{126} = 0$

fix_sol0#60: $x_{126} = 0$

➤ Expiration = 1

$$\text{fix_exp1\#61: } x_{77} - x_{96} - x_{103} + x_{127} = 0$$

$$\text{fix_sol1\#62: } x_{127} = 0$$

✓ Περιορισμοί που δημιουργούνται ανάλογα με τις προτιμήσεις των υπτάμενων.

$$\text{fix_bids1\#63: } x_{78} = 1$$

$$\text{fix_bids2\#64: } x_{104} = 1$$

$$\text{fix_bids3\#65: } x_{90} = 1$$

$$\text{fix_bids4\#66: } x_{123} = 1$$

$$\text{fix_bids5\#67: } x_{93} = 1$$

$$\text{fix_bids6\#68: } x_{64} = 1$$

$$\text{fix_bids7\#69: } x_{70} = 1$$

$$\text{fix_bids8\#70: } x_{110} = 1$$

$$\text{fix_bids9\#71: } x_{121} = 1$$

$$\text{fix_bids10\#72: } x_{84} = 1$$

$$\text{fix_bids11\#73: } x_{95} = 1$$

$$\text{fix_bids12\#74: } x_{97} = 1$$

$$\text{fix_bids13\#75: } x_{101} = 1$$

$$\text{fix_bids14\#76: } x_{76} = 1$$

$$\text{fix_bids15\#77: } x_{81} = 1$$

$$\text{fix_bids16\#78: } x_{119} = 1$$

✓ Περιορισμοί για την ισορροπία των τμημάτων θεωρίας.

$$\text{up_bal0\#79: } -x_{64} - x_{68} - x_{72} - x_{80} - x_{84} - x_{93} - x_{100} - x_{114} - x_{121} + x_{128} \geq -3$$

$$\text{low_bal0\#80: } x_{64} + x_{68} + x_{72} + x_{80} + x_{84} + x_{93} + x_{100} + x_{114} + x_{121} + x_{128} \geq 1$$

$$\text{up_bal1\#81: } -x_{65} - x_{69} - x_{73} - x_{85} - x_{89} - x_{101} - x_{104} - x_{115} - x_{123} + x_{129} \geq -3$$

$$\text{low_bal1\#82: } x_{65} + x_{69} + x_{73} + x_{85} + x_{89} + x_{101} + x_{104} + x_{115} + x_{123} + x_{129} \geq 1$$

$$\text{up_bal2\#83: } -x_{66} - x_{70} - x_{74} - x_{78} - x_{81} - x_{86} - x_{90} - x_{106} - x_{109} - x_{116} - x_{124} + x_{130} \geq -3$$

$$\text{low_bal2\#84: } x_{66} + x_{70} + x_{74} + x_{78} + x_{81} + x_{86} + x_{90} + x_{106} + x_{109} + x_{116} + x_{124} + x_{130} \geq 1$$

$$\text{up_bal3\#85: } -x_{75} - x_{91} - x_{95} - x_{97} - x_{107} - x_{110} - x_{117} + x_{131} \geq -3$$

$$\text{low_bal3\#86: } x_{75} + x_{91} + x_{95} + x_{97} + x_{107} + x_{110} + x_{117} + x_{131} \geq 1$$

$$\text{up_bal4\#87: } -x_{82} - x_{87} - x_{98} - x_{111} - x_{118} + x_{132} \geq -3$$

$$\text{low_bal4\#88: } x_{82} + x_{87} + x_{98} + x_{111} + x_{118} + x_{132} \geq 1$$

$$\text{up_bal5\#89: } -x_{76} - x_{102} - x_{112} - x_{119} + x_{133} \geq -3$$

$$\text{low_bal5\#90: } x_{76} + x_{102} + x_{112} + x_{119} + x_{133} \geq 1$$

Οι μεταβλητές ανάθεσης που μας έδωσε ο κώδικας βρίσκονται σε αντιστοιχία με τις μεταβλητές απόφασης του παραπάνω μαθηματικού μοντέλου. Αυτό φαίνεται στον παρακάτω πίνακα.

Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος	Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος
X111	X64	X36	X76
X12	X65	X43	X78
X13	X66	X51	X80
X21	X68	X53	X81
X22	X69	X55	X82
X23	X70	X61	X84
X31	X72	X62	X85
X32	X73	X63	X86
X33	X74	X65	X87
X34	X75	X72	X89
X73	X90	X15,6	X119
X74	X91	X16,1	X121
X81	X93	X17,2	X123
X94	X95	X17,3	X124
X10,4	X97	Y1	X63
X10,5	X98	Y2	X67
X11,1	X100	Y3	X71
X11,2	X101	Y4	X77
X11,5	X102	Y5	X79
X12,2	X104	Y6	X83
X13,3	X106	Y7	X88
X13,4	X107	Y8	X92
X14,3	X109	Y9	X94
X14,4	X110	Y10	X96
X14,5	X111	Y11	X99
X14,6	X112	Y12	X103
X15,1	X114	Y13	X105
X15,2	X115	Y14	X108
X15,3	X116	Y15	X113
X15,4	X117	Y16	X120
X15,5	X118	Y17	X122

Πίνακας 7 4: Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 1

Τρέχουμε τον κώδικα μόνο για τα τμήματα θεωρίας (classrooms) και βρίσκουμε ότι ο χρόνος επίλυσης του προβλήματος είναι **2,715sec** και τα αποτελέσματα που προέκυψαν φαίνονται στον παρακάτω πίνακα.

ΙΠΤΑΜΕΝΟΣ	ΣΥΝΕΔΡΙΑ ΠΟΥ ΑΝΑΤΙΘΕΝΤΑΙ
1	1
2	3
3	6
4	3
5	3
6	1
7	3
8	1
9	4
10	4
11	2
12	2
13	3
14	4
15	6
16	1
17	2

Πίνακας 7 5: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 1

Στην περίπτωση που τρέξουμε το πρόγραμμα για προσομοιωτή (**simulator**), εμφανίζονται διαφορές τόσο στον χρόνο όσο και στην ανάθεση κάποιων ιπτάμενων. Έτσι τα αποτελέσματα που βρήκαμε για τον προσομοιωτή είναι:

- Χρόνος επίλυσης **1,961sec**
- Ο ιπτάμενος 1 ανατίθεται στην συνεδρία 2 αντί στην 1
- Ο ιπτάμενος 6 ανατίθεται στην συνεδρία 2 αντί στην 1
- Ο ιπτάμενος 13 ανατίθεται στην συνεδρία 4 αντί στην 3
- Ο ιπτάμενος 14 ανατίθεται στην συνεδρία 5 αντί στην 4
- Ο ιπτάμενος 14 ανατίθεται στην συνεδρία 5 αντί στην 6

Όσον αφορά την αντικειμενική συνάρτηση, θα παραμείνει ίδια με τη διαφορά ότι οι συντελεστές κόστους που αφορούν τις μετβλητές απόφασης (x128 έως x133), και σχετίζονται με την ισορροπία, θα είναι ίσοι με το 0.

7.2 Δεύτερο πείραμα

Το δεύτερο παράδειγμα αφορά την ανάθεση δεκαπέντε (**N=15**) υπτάμενων σε πέντε (**S=5**) εκπαιδευτικές συνεδρίες. Στον πρώτο πίνακα παρουσιάζονται τα χαρακτηριστικά που έχει κάθε υπάμενος αλλά και η χωρητικότητα της κάθε συνεδρίας.

N	EXPIRATION	SENIORITY	POSITION
1	-1	5	1(CP)
2	-1	14	1(CP)
3	0	6	3(PURSER)
4	-1	8	4(FA)
5	1	10	2(FO)
6	-1	1	1(CP)
7	-1	7	2(FO)
8	-1	11	4(FA)
9	0	13	2(FO)
10	1	9	3(PURSER)
11	0	4	1(CP)
12	1	12	3(PURSER)
13	0	2	4(FA)
14	-1	3	4(FA)
15	-1	15	2(FO)

Πίνακας 7 6:Χαρακτηριστικά υπτάμενων παράδειγμα 2

S \ N	1	2	3	4	5	ΣΥΝΟΛΟ ΕΦΙΚΤΩΝ ΣΥΝΕΔΡΙΩΝ
1	0	0	0	1	2	4
2	5	4	3	2	1	2
3	1	0	0	0	2	2
4	0	1	0	2	0	2

5	5	3	1	2	4	5
6	1	2	3	4	0	3
7	0	0	2	1	3	2
8	1	2	3	4	5	2
9	0	1	0	0	0	1
10	5	3	1	2	4	5
11	1	0	2	3	4	1
12	0	1	0	2	0	4
13	2	3	1	5	4	3
14	1	0	2	3	0	2
15	1	2	3	4	5	5

Πίνακας 7 7:Λίστα με τις προτιμήσεις των υπάμενων και το σύνολο των εφικτών συνεδριών παράδειγμα 2

	ΓΙΑ ΚΑΘΕ ΣΥΝΕΔΡΙΑ
ΣΥΝΟΛΙΚΗ ΧΩΡΗΤΙΚΟΤΗΤΑ	5
ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΑΜΠΙΝΑΣ	4
ΧΩΡΗΤΙΚΟΤΗΤΑ ΠΙΛΟΤΗΡΙΟΥ	4
ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΥΒΕΡΝΗΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΣΥΓΚΥΒΕΡΝΗΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΡΧΙΦΡΟΝΤΙΣΤΩΝ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΕΡΟΣΥΝΟΔΩΝ	2

Πίνακας 7 8:Η χωρητικότητα για κάθε συνεδρία παράδειγμα 2

Αφού τρέξαμε τον κώδικα για το παράδειγμα 2 πήραμε την παρακάτω αντικειμενική συνάρτηση και τους αντίστοιχους περιορισμούς:

✓ Αντικειμενική συνάρτηση

Minimize obj: $0 x_1 + 0 x_2 + 0 x_3 + 0 x_4 + 0 x_5 + 0 x_6 + 0 x_7 + 0 x_8 + 0 x_9 + 0 x_{10} + 0 x_{11} + 0 x_{12} + 0 x_{13} + 0 x_{14} + 0 x_{15} + 0 x_{16} + 0 x_{17} + 0 x_{18} + 0 x_{19} + 0 x_{20} + 0 x_{21} + 0 x_{22} + 0 x_{23} + 0 x_{24} + 0 x_{25} + 0 x_{26} + 0 x_{27} + 0 x_{28} + 0 x_{29} + 0 x_{30} + 0 x_{31} + 0 x_{32} + 0 x_{33} + 0 x_{34} + 0 x_{35} + 0 x_{36} + 0 x_{37} + 0 x_{38} + 0 x_{39} + 0 x_{40} + 0 x_{41} + 0 x_{42} + 0 x_{43} + 0 x_{44} + 0 x_{45} + 0 x_{46} + 0 x_{47} + 0 x_{48} + 0 x_{49} + 0 x_{50} + 0 x_{51} + 0 x_{52} + 0 x_{53} + 0 x_{54} + 1000 x_{112} + 1000 x_{113} + 1000 x_{114} + 1000 x_{115} + 1000 x_{116}$

✓ Οι περιορισμοί ανάθεσης και μη ανάθεσης των υπτάμενων. Η πρώτη μεταβλητή απόφασης αναφέρεται στην μη ανάθεση.

Assign0#0: $x_{55} + x_{56} + x_{57} + x_{58} + x_{59} = 1$

Assign1#1: $x_{60} + x_{61} + x_{62} = 1$

Assign2#2: $x_{63} + x_{64} + x_{65} = 1$

Assign3#3: $x_{66} + x_{67} = 1$

Assign4#4: $x_{68} + x_{69} + x_{70} + x_{71} + x_{72} + x_{73} = 1$

Assign5#5: $x_{74} + x_{75} + x_{76} + x_{77} = 1$

Assign6#6: $x_{78} + x_{79} + x_{80} = 1$

Assign7#7: $x_{81} + x_{82} + x_{83} = 1$

Assign8#8: $x_{84} + x_{85} = 1$

Assign9#9: $x_{86} + x_{87} + x_{88} + x_{89} + x_{90} + x_{91} = 1$

Assign10#10: $x_{92} + x_{93} = 1$

Assign11#11: $x_{94} + x_{95} = 1$

Assign12#12: $x_{96} + x_{97} + x_{98} + x_{99} = 1$

Assign13#13: $x_{100} + x_{101} + x_{102} = 1$

Assign14#14: $x_{103} + x_{104} + x_{105} + x_{106} + x_{107} + x_{108} = 1$

- ✓ Οι περιορισμοί για την χωρητικότητα του προσομοιωτή

Για τους CP:

$$\text{sim1_cap0\#15: } x_{56} + x_{93} \leq 1$$

$$\text{sim1_cap1\#16: } x_{57} + x_{61} + x_{75} \leq 1$$

$$\text{sim1_cap2\#17: } x_{58} + x_{76} \leq 1$$

$$\text{sim1_cap3\#18: } x_{59} + x_{62} + x_{77} \leq 1$$

$$\text{sim1_cap4\#19: } \leq 1$$

Για τους FO:

$$\text{sim2_cap0\#20: } x_{69} + x_{79} + x_{104} \leq 1$$

$$\text{sim2_cap1\#21: } x_{70} + x_{80} + x_{105} \leq 1$$

$$\text{sim2_cap2\#22: } x_{71} + x_{106} \leq 1$$

$$\text{sim2_cap3\#23: } x_{72} + x_{107} \leq 1$$

$$\text{sim2_cap4\#24: } x_{73} + x_{85} + x_{108} \leq 1$$

- ✓ Οι περιορισμοί χωρητικότητα για τα τμήματα θεωρίας

- Συνολική χωρητικότητα

$$\text{tot_cap0\#15: } x_{56} + x_{69} + x_{79} + x_{82} + x_{87} + x_{93} + x_{101} + x_{104} \leq 5$$

$$\text{tot_cap1\#16: } x_{57} + x_{61} + x_{70} + x_{75} + x_{80} + x_{88} + x_{95} + x_{102} + x_{105} \leq 5$$

$$\text{tot_cap2\#17: } x_{58} + x_{64} + x_{67} + x_{71} + x_{76} + x_{89} + x_{97} + x_{106} \leq 5$$

$$\text{tot_cap3\#18: } x_{59} + x_{62} + x_{65} + x_{72} + x_{77} + x_{83} + x_{90} + x_{98} + x_{107} \leq 5$$

$$\text{tot_cap4\#19: } x_{73} + x_{85} + x_{91} + x_{99} + x_{108} \leq 5$$

- Χωρητικότητα καμπίνας

$$\text{cabin0\#25: } x_{82} + x_{87} + x_{101} \leq 4$$

$$\text{cabin1\#26: } x_{88} + x_{95} + x_{102} \leq 4$$

$$\text{cabin2\#27: } x_{64} + x_{67} + x_{89} + x_{97} \leq 4$$

$$\text{cabin3\#28: } x_{65} + x_{83} + x_{90} + x_{98} \leq 4$$

$$\text{cabin4\#29: } x_{91} + x_{99} \leq 4$$

- Χωρητικότητα πιλοτηρίου

$$\text{Cockpit0\#20: } x_{56} + x_{69} + x_{79} + x_{93} + x_{104} \leq 4$$

$$\text{Cockpit1\#21: } x_{57} + x_{61} + x_{70} + x_{75} + x_{80} + x_{105} \leq 4$$

$$\text{Cockpit2\#22: } x_{58} + x_{71} + x_{76} + x_{106} \leq 4$$

$$\text{Cockpit3\#23: } x_{59} + x_{62} + x_{72} + x_{77} + x_{107} \leq 4$$

$$\text{Cockpit4\#24: } x_{73} + x_{85} + x_{108} \leq 4$$

➤ Χωρητικότητα κυβερνητών

$$\text{CP_cap0\#30: } x_{56} + x_{93} \leq 2$$

$$\text{CP_cap1\#31: } x_{57} + x_{61} + x_{75} \leq 2$$

$$\text{CP_cap2\#32: } x_{58} + x_{76} \leq 2$$

$$\text{CP_cap3\#33: } x_{59} + x_{62} + x_{77} \leq 2$$

➤ Χωρητικότητα συγκυβερνητών

$$\text{FO_cap0\#34: } x_{69} + x_{79} + x_{104} \leq 2$$

$$\text{FO_cap1\#35: } x_{70} + x_{80} + x_{105} \leq 2$$

$$\text{FO_cap2\#36: } x_{71} + x_{106} \leq 2$$

$$\text{FO_cap3\#37: } x_{72} + x_{107} \leq 2$$

$$\text{FO_cap4\#38: } x_{73} + x_{85} + x_{108} \leq 2$$

➤ Χωρητικότητα αρχιφροντιστών

$$\text{purser0\#39: } x_{87} \leq 2$$

$$\text{purser1\#40: } x_{88} + x_{95} \leq 2$$

$$\text{purser2\#41: } x_{64} + x_{89} \leq 2$$

$$\text{purser3\#42: } x_{65} + x_{90} \leq 2$$

$$\text{purser4\#43: } x_{91} \leq 2$$

➤ Χωρητικότητα αεροσυνοδών

$$\text{fl_att0\#44: } x_{82} + x_{101} \leq 2$$

$$\text{fl_att1\#45: } x_{102} \leq 2$$

$$\text{fl_att2\#46: } x_{67} + x_{97} \leq 2$$

$$\text{fl_att3\#47: } x_{83} + x_{98} \leq 2$$

$$\text{fl_att4\#48: } x_{99} \leq 2$$

✓ Οι περιορισμοί για την λήξη της πτητικής ικανότητας

➤ Expiration = -1

$$\text{fix_expm1\#49: } -x_{55} - x_{60} - x_{66} - x_{74} - x_{78} - x_{81} - x_{100} - x_{103} + x_{109} = 0$$

$$\text{fix_solm1\#50: } x_{109} = 0$$

➤ Expiration = 0

$$\text{fix_exp0\#51: } -x63 - x84 - x92 - x96 + x110 = 0$$

$$\text{fix_sol0\#52: } x110 = 0$$

➤ Expiration = 1

$$\text{fix_exp1\#53: } -x68 - x86 - x94 + x111 = 0$$

$$\text{fix_sol1\#54: } x111 = 0$$

✓ Περιορισμοί που δημιουργούνται ανάλογα με τις προτιμήσεις των υπτάμενων.

$$\text{fix_bids0\#55: } x75 = 1$$

$$\text{fix_bids1\#56: } x97 = 1$$

$$\text{fix_bids2\#57: } x101 = 1$$

$$\text{fix_bids3\#58: } x93 = 1$$

$$\text{fix_bids4\#59: } x59 = 1$$

$$\text{fix_bids8\#60: } x89 = 1$$

$$\text{fix_bids9\#61: } x71 = 1$$

$$\text{fix_bids10\#62: } x82 = 1$$

$$\text{fix_bids11\#63: } x95 = 1$$

$$\text{fix_bids13\#64: } x62 = 1$$

$$\text{fix_bids14\#65: } x104 = 1$$

✓ Περιορισμοί για την ισορροπία των τμημάτων θεωρίας

$$\text{up_bal0\#66: } -x56 - x69 - x79 - x82 - x87 - x93 - x101 - x104 + x112 \geq -4$$

$$\text{low_bal0\#67: } x56 + x69 + x79 + x82 + x87 + x93 + x101 + x104 + x112 \geq 2$$

$$\text{up_bal1\#68: } -x57 - x61 - x70 - x75 - x80 - x88 - x95 - x102 - x105 + x113 \geq -4$$

$$\text{low_bal1\#69: } x57 + x61 + x70 + x75 + x80 + x88 + x95 + x102 + x105 + x113 \geq 2$$

$$\text{up_bal2\#70: } -x58 - x64 - x67 - x71 - x76 - x89 - x97 - x106 + x114 \geq -4$$

$$\text{low_bal2\#71: } x58 + x64 + x67 + x71 + x76 + x89 + x97 + x106 + x114 \geq 2$$

$$\text{up_bal3\#72: } -x59 - x62 - x65 - x72 - x77 - x83 - x90 - x98 - x107 + x115 \geq -4$$

$$\text{low_bal3\#73: } x59 + x62 + x65 + x72 + x77 + x83 + x90 + x98 + x107 + x115 \geq 2$$

$$\text{up_bal4\#74: } -x73 - x85 - x91 - x99 - x108 + x116 \geq -4$$

$$\text{low_bal4\#75: } x73 + x85 + x91 + x99 + x108 + x116 \geq 2$$

Οι μεταβλητές ανάθεσης που μας έδωσε ο κώδικας βρίσκονται σε αντιστοιχία με τις μεταβλητές απόφασης του παραπάνω μαθηματικού μοντέλου. Αυτό φαίνεται στον παρακάτω πίνακα.

Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος	Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος
X11	X56	X64	X77
X12	X57	X71	X79
X13	X58	X72	X80
X14	X59	X81	X82
X22	X61	X84	X83
X24	X62	X95	X85
X33	X64	X10,1	X87
X34	X65	X10,2	X88
X42	X67	X10,3	X89
X51	X69	X10,4	X90
X52	X70	X10,5	X91
X53	X71	X11,1	X93
X54	X72	X12,2	X95
X55	X73	X13,3	X97
X62	X75	X13,4	X98
X63	x76	X13,5	X99
X14,1	X101	Y5	X68
X14,2	X102	Y6	X74
X15,1	X104	Y7	X78
X15,2	X105	Y8	X81
X15,3	X106	Y9	X84
X15,4	X107	Y10	X86
X15,5	X108	Y11	X92
Y1	X55	Y12	X94
Y2	X60	Y13	X96
Y3	X63	Y14	X100
Y4	X66	Y15	X103

Πίνακας 7 9: Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 2

Τρέχουμε τον κώδικα μόνο για τα τμήματα θεωρίας (**classrooms**) και βρήκαμε ότι ο χρόνος επίλυσης του προβλήματος είναι **2,624 sec** και τα αποτελέσματα που προέκυψαν φαίνονται στον παρακάτω πίνακα

ΙΠΤΑΜΕΝΟΣ	ΣΥΝΕΔΡΙΑ ΠΟΥ ΑΝΑΤΙΘΕΝΤΑΙ
1	4
2	4
3	4
4	3
5	3
6	2
7	2
8	1
9	5
10	3
11	1
12	2
13	3
14	1
15	1

Πίνακας 7 10: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 2

Στην περίπτωση που τρέξουμε το πρόγραμμα για προσομοιωτή (**simulator**), εμφανίζονται διαφορές τόσο στον χρόνο όσο και στην ανάθεση κάποιον ιπτάμενων. Έτσι τα αποτελέσματα που βρήκαμε για τον προσομοιωτή είναι:

- Χρόνος επίλυσης **2,076 sec**
- Ο ιπτάμενος 1 ανατίθεται στην συνεδρία 3 αντί στην 4
- Ο ιπτάμενος 3 ανατίθεται στην συνεδρία 3 αντί στην 4

Όσον αφορά την αντικειμενική συνάρτηση, θα παραμείνει ίδια με τη διαφορά ότι οι συντελεστές κόστους που αφορούν τις μετβλητές απόφασης (x112 έως x116), και σχετίζονται με την ισορροπία, θα είναι ίσοι με το 0

7.3 Τρίτο πείραμα

Το τρίτο παράδειγμα αφορά την ανάθεση πέντε (**N=5**) ιπτάμενων σε τρεις (**S=3**) εκπαιδευτικές συνεδρίες. Τώρα όμως υπάρχει η ιδιαιτερότητα ότι όλοι οι ιπτάμενοι έχουν ίδιες προτιμήσεις αλλά και ίδιο μήνα λήξης της πτητικής τους ικανότητα. Στην

αρχή παρουσιάζονται τα χαρακτηριστικά που έχει κάθε ιπτάμενος αλλά και η χωρητικότητα της κάθε συνεδρίας.

N	EXPIRATION	SENIORITY	POSITION
1	-1	5	1(CP)
2	-1	2	1(CP)
3	-1	1	3(PURSER)
4	-1	4	4(FA)
5	-1	3	2(FO)

Πίνακας 7 11: Χαρακτηριστικά ιπτάμενων παράδειγμα 3

S \ N	0	1	3	ΣΥΝΟΛΟ ΕΦΙΚΤΩΝ ΣΥΝΕΔΡΙΩΝ
1	1	2	3	1
2	1	2	3	2
3	1	2	3	1
4	1	2	3	1
5	1	2	3	2

Πίνακας 7 12:Λίστα προτιμήσεων και σύνολο εφικτών συνεδριών παράδειγμα 3

	ΓΙΑ ΚΑΘΕ ΣΥΝΕΔΡΙΑ
ΣΥΝΟΛΙΚΗ ΧΩΡΗΤΙΚΟΤΗΤΑ	3
ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΑΜΠΙΝΑΣ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΠΙΛΟΤΗΡΙΟΥ	2
ΧΩΡΗΤΙΚΟΤΗΤΑ ΚΥΒΕΡΝΗΤΩΝ	1
ΧΩΡΗΤΙΚΟΤΗΤΑ ΣΥΓΚΥΒΕΡΝΗΤΩΝ	1
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΡΧΙΦΡΟΝΤΙΣΤΩΝ	1
ΧΩΡΗΤΙΚΟΤΗΤΑ ΑΕΡΟΣΥΝΟΔΩΝ	1

Πίνακας 7 13:Χωρητικότητα για κάθε συνεδρία παράδειγμα 3

Αφού τρέξαμε τον κώδικα για το παράδειγμα3 πήραμε την παρακάτω αντικειμενική συνάρτηση και τους αντίστοιχους περιορισμούς.

- ✓ Η αντικειμενική συνάρτηση είναι

$$\text{Minimizeobj: } 0 x_1 + 0 x_2 + 0 x_3 + 0 x_4 + 0 x_5 + 0 x_6 + 0 x_7 + 0 x_8 + 0 x_9 + 0 x_{10} + 0x_{11} + 0 x_{12} + 1000 x_{26} + 1000 x_{27} + 1000 x_{28}$$

- ✓ Οι περιορισμοί για την ανάθεση και την μη ανάθεση των ιπτάμενων. Η πρώτη μεταβλητή απόφασης αναφέρεται στην μη ανάθεση

$$\text{Assign0\#0: } x_{13} + x_{14} = 1$$

$$\text{Assign1\#1: } x_{15} + x_{16} + x_{17} = 1$$

$$\text{Assign2\#2: } x_{18} + x_{19} = 1$$

$$\text{Assign3\#3: } x_{20} + x_{21} = 1$$

$$\text{Assign4\#4: } x_{22} + x_{23} + x_{24} = 1$$

- ✓ Οι περιορισμοί για την χωρητικότητα του προσομοιωτή

ΓιατουςCP:

$$\text{sim1_cap0\#5: } x_{14} \leq 1$$

$$\text{sim1_cap1\#6: } x_{16} \leq 1$$

$$\text{sim1_cap2\#7: } x_{17} \leq 1$$

ΓιατουςFO:

$$\text{sim2_cap0\#8: } x_{23} \leq 1$$

$$\text{sim2_cap2\#9: } x_{24} \leq 1$$

- ✓ Οι περιορισμοί για την χωρητικότητα των τμημάτων θεωρίας
 - Συνολική χωρητικότητα

$$\text{tot_cap0\#5: } x_{14} + x_{23} \leq 3$$

$$\text{tot_cap1\#6: } x_{16} + x_{21} \leq 3$$

$$\text{tot_cap2\#7: } x_{17} + x_{19} + x_{24} \leq 3$$

- Χωρητικότητα καμπίνας

cabin1#11: $x_{21} \leq 2$

cabin2#12: $x_{19} \leq 2$

- Χωρητικότητα πιλοτήριου

Cockpit0#8: $x_{14} + x_{23} \leq 2$

Cockpit1#9: $x_{16} \leq 2$

Cockpit2#10: $x_{17} + x_{24} \leq 2$

- Χωρητικότητα κυβερνητών

CP_cap0#13: $x_{14} \leq 1$

CP_cap1#14: $x_{16} \leq 1$

CP_cap2#15: $x_{17} \leq 1$

- Χωρητικότητα συγκυβερνητών

FO_cap0#16: $x_{23} \leq 1$

FO_cap2#17: $x_{24} \leq 1$

- Χωρητικότητα αρχιφροντιστών

purser2#18: $x_{19} \leq 1$

- Χωρητικότητα αεροσυνοδών

fl_att1#19: $x_{21} \leq 1$

- ✓ Περιορισμοί για την λήξη της πτητικής ικανότητας

- Expiration = -1

fix_expm1#20: $-x_{13} - x_{15} - x_{18} - x_{20} - x_{22} + x_{25} = 0$

fix_solm1#21: $x_{25} = 0$

- ✓ Περιορισμοί που δημιουργούνται ανάλογα με τις προτιμήσεις των ιπτάμενων.

$$\text{fix_bids0\#22: } x_{19} = 1$$

$$\text{fix_bids1\#23: } x_{16} = 1$$

$$\text{fix_bids2\#24: } x_{23} = 1$$

$$\text{fix_bids3\#25: } x_{21} = 1$$

$$\text{fix_bids4\#26: } x_{14} = 1$$

- ✓ Περιορισμοί για την ισορροπία των τμημάτων θεωρίας

$$\text{up_bal0\#27: } -x_{14} - x_{23} + x_{26} \geq -2$$

$$\text{low_bal0\#28: } x_{14} + x_{23} + x_{26} \geq 0$$

$$\text{up_bal1\#29: } -x_{16} - x_{21} + x_{27} \geq -2$$

$$\text{low_bal1\#30: } x_{16} + x_{21} + x_{27} \geq 0$$

$$\text{up_bal2\#31: } -x_{17} - x_{19} - x_{24} + x_{28} \geq -2$$

$$\text{low_bal2\#32: } x_{17} + x_{19} + x_{24} + x_{28} \geq 0$$

Οι μεταβλητές ανάθεσης που μας έδωσε ο κώδικας βρίσκονται σε αντιστοιχία με τις μεταβλητές απόφασης του παραπάνω μαθηματικού μοντέλου. Αυτό φαίνεται στον παρακάτω πίνακα.

Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος	Μεταβλητές απόφασης μαθηματικού μοντέλου	Μεταβλητές απόφασης του προγράμματος
X11	X14	Y1	X13
X22	X16	Y2	X15
X23	X17	Y3	X18
X31	X19	Y4	X20
X42	X21	Y5	X22
X51	X23		
X52	X24		

Πίνακας 7 14: Αντιστοιχία μεταβλητών απόφασης του μαθηματικού μοντέλου και του προγράμματος παράδειγμα 3

Τρέχουμε τον κώδικα μόνο για τα τμήματα θεωρίας (**classrooms**) και βρήκαμε ότι ο χρόνος επίλυσης του προβλήματος είναι **0,165sec** και τα αποτελέσματα που προέκυψαν φαίνονται στον παρακάτω πίνακα.

ΙΠΤΑΜΕΝΟΣ	ΣΥΝΕΔΡΙΑ ΠΟΥ ΑΝΑΤΙΘΕΝΤΑΙ
1	1
2	2
3	3
4	2
5	1

Πίνακας 7 15: Η ανάθεση των ιπτάμενων στις συνεδρίες παράδειγμα 3

Στην άλλη περίπτωση, αν δηλαδή τρέξουμε τον κώδικα για τον προσομοιωτή(**simulator**) δεν παρατηρούνται αλλαγές στην ανάθεση των ιπτάμενων και στον χρόνο. Η μόνη αλλαγή που παρατηρείται είναι στην αντικειμενική συνάρτηση, όπου οι συντελεστές των μεταβλητών απόφασης που αφορούν την ισορροπία θα είναι μηδενικοί(x26 έως x28)

8.ΕΠΙΛΟΓΟΣ

Σ' αυτό το κεφάλαιο κάνουμε μια ανασκόπηση στην εργασία που εκτελέσαμε και παρουσιάζουμε τα συμπεράσματα που προέκυψαν κατά την επίλυση του προβλήματος μας. Τέλος προτείνουμε και πιθανά θέματα με τα οποία θα μπορούσαμε να επεκτείνουμε την εργασία μας στο μέλλον.

8.1 Σύνοψη και αποτελέσματα

Στόχος αυτής της εργασίας ήταν να βρούμε την βέλτιστη ανάθεση των ιπτάμενων σε εκπαιδευτικές συνεδρίες ,για προσομοιωτή και τμήματα θεωρίας με τέτοιο τρόπο ώστε να τηρούνται οι περιορισμοί δυναμικότητας, εξισορρόπησης, προτιμήσεων και λήξη πτητικής ικανότητας.

Για την επίλυση του προβλήματος αρχικά κάναμε έρευνα σε διάφορες βιβλιογραφικές αναφορές που σχετίζονται με το θέμα της εργασίας μας και στη συνέχεια ποσοτικοποιήσαμε τα δεδομένα μας και προχωρήσαμε στην μαθηματική μοντελοποίηση του προβλήματος. Για την επίλυση του μαθηματικού μας μοντέλου καταστρώσαμε τον κατάλληλο κώδικα χρησιμοποιώντας την γλώσσα προγραμματισμού C++ και το λογισμικό βελτιστοποίησης προβλημάτων CPLEX. Στη συνέχεια εκτελέσαμε μια σειρά πειραμάτων είτε με πραγματικά δεδομένα είτε με δεδομένα από γεννήτριες τυχαίων αριθμών για να διαπιστώσουμε την ορθή λειτουργία του κώδικα μας.

Το αποτέλεσμα που πήραμε ήταν ένα πρόγραμμα ανάθεσης ιπτάμενων η οποία πραγματοποιείται εξασφαλίζοντας τις μέγιστες προτιμήσεις βάση αρχαιότητας των ιπτάμενων. Επίσης εξασφαλίσαμε την κάλυψη όσο την δυνατόν περισσότερων εφικτών θέσεων σε κάθε συνέδρια χωρίς να υπάρχει ιπτάμενος που δεν έχει ανατεθεί. Επιπλέον δόθηκε μεγάλη προσοχή στην λήξη της πτητικής ικανότητας των ιπτάμενων, δίνοντας προτεραιότητα σ' αυτούς όπου η λήξη τους είναι τον παρόντα μήνα αλλά και στην ισορροπία που πρέπει να έχουν τα τμήματα θεωρίας ώστε να μην υπάρχει συσσώρευση ιπτάμενων σε συγκεκριμένα τμήματα.

Συμφώνα με τα παραπάνω, συμπεραίνουμε ότι το μοντέλο που καταστρώσαμε με την μεγάλη πολυπλοκότητα που το διακρίνει μπορεί να μας δώσει βέλτιστες λύσεις σε σύντομα χρονικά διαστήματα. Επίσης έχοντας μια ιεραρχία στα στάδια της επίλυσης μπορούμε να επιλύσουμε μεγάλα προβλήματα πιο εύκολα εφόσον ο κώδικας μας με αυτήν την μεθοδολογία καταφέρνει να περιορίσει το μέγεθος των συντελεστών απόφασης που από μόνο τους είναι απίστευτα μεγάλο.

8.2Μελλοντικές επεκτάσεις

Στο τέλος αυτής της εργασίας αναφέρουμε ορισμένα θέματα τα οποία θα μπορούσαν να εξελίξουν αυτήν εργασία και να προβληματίσουν τις αεροπορικές εταιρείες στο μέλλον.

Μια επέκταση του μοντέλου που θα μπορούσε να ασχοληθεί κάποιος είναι βελτιστοποίηση της ανάθεσης όχι μόνο των εκπαιδευόμενων αλλά και των εκπαιδευτών. Θα εργαζόμασταν δηλαδή όχι μόνο για την ελαχιστοποίηση των ιπτάμενων που θα μείνουν εκτός αλλά και για την ελαχιστοποίηση των εκπαιδευτών που δεν θα αναλάβουν κάποιο τμήμα. Επίσης ένα άλλο θέμα για έρευνα θα είναι να δημιουργηθούν συνεδρίες που τα μαθήματα θα γίνονται σε άλλες γλώσσες. Για παράδειγμα ένας ιπτάμενος ο οποίος δεν κατανοεί την αγγλική γλώσσα θα μπορεί να έχει την ευκαιρία να παρακολουθήσει τα σεμινάρια σε οποία άλλη γλώσσα γνωρίζει καλύτερα. Τέλος θα μπορούσε να δημιουργηθεί ένα καινούριος περιορισμός στην χωρητικότητα του προσομοιωτή. Στο πρόβλημα μας έχουμε σαν δεδομένο ότι στον προσομοιωτή θα μπει ένας κυβερνήτης και ένα συγκυβερνήτης απαραίτητως. Όμως μπορεί να βρεθούν αεροπορικές εταιρείες οι οποίες θα επιθυμούν εάν δεν μπορεί να μπει ένας κυβερνήτης και ένας συγκυβερνήτης να μπουν 2 από την κάθε ειδικότητα ώστε να μην μείνουν οι θέσεις κενές , πράξη η οποία αλλάζει τον ήδη υπάρχοντα περιορισμό χωρητικότητας.

9. ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Καρλαύτης, Μ., Λαγαρός, Ν. *Επιχειρησιακή Έρευνα και Βελτιστοποίηση για Μηχανικούς*, . Αθήνα. : Εκδόσεις Συμμετρία, 2010.
2. Στογιάννης, Ι. Κολέτσος και Δ. *Εισαγωγή στην Επιχειρησιακή Έρευνα*. Αθήνα : Εκδόσεις Συμμετών, 2012.
3. from Wikipedia, the free Encyclopedia. *Βελτιστοποίηση*. 16 Δεκεμβρίου 2016.
4. ΤΑΗΑ, HAMDY Α. *ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΧΕΙΡΗΣΙΑΚΗ ΕΡΕΥΝΑ*. s.l. : ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2014. 9η Έκδοση.
5. ILOG CPLEX Optimization Studio 12.5.0. https://www.ibm.com/support/knowledgecenter/SSSA5P_12.5.0/ilog.odms.ide.help/OPL_Studio/opllanguser/topics/opl_languser_app_areas_MILP_what.html.
[Ηλεκτρονικό]
6. Shapiro, M. *"Scheduling Crewmen for Recurrent Training*. s.l. : Interfaces, 1981. 11 (3), pp. 1-8,.
7. Brown, R. P. *Optimizing Readiness and Equity in Marine Corps Aviation Training Schedules*. Naval Postgraduate School at Monterey, California, USA : s.n., 1995.
8. G. Yu, S. Dugan and M. Argüello. *"Moving Toward an Integrated Decision Support System for Manpower Planning at Continental Airlines: Optimization of Pilot Training Assignments,*. s.l. : Industrial Applications of Combinatorial Optimization, 1998. 16, pp. 1-24.
9. M. Gamache, F. Soumis, D. Villeneuve, J. Desrosiers and É. Gélinas. *The Preferential Bidding System at Air Canada*. s.l. : Transportation Science, 1998. 32 (3), pp. 246-255.
10. Davut Aslan, Captain, TUAF. *A DECISION SUPPORT SYSTEM FOR EFFECTIVE SCHEDULING IN AN F-16 PILOT TRAINING SQUADRON* . Wright-Patterson AFB, OH. School of Engineering and Management. : Air Force Inst. of Tech, March 26, 2003.
11. B. G. Thengvall and J. E. Pachon. *Integrated Decision Support System for Optimizing the Training and Transition of Airline Pilots*. United States Patent : s.n., 2008. 7,346,528 B2.
12. Habel, Reka. *Pilot Training Optimization for Airlines*. Budapest : s.n., 2013.
13. From Wikipedia, the free encyclopedia. *Flight simulator*. 24 September 2019.
14. Oketch, Christine A. *ALGORITHM OPTIMIZATION FOR SOLVING CREW SCHEDULING PROBLEMS*. Universitat Oberta de Catalunya : s.n., 2013.

15. Gang Yu, Julian Pachon, Benjamin Thengvall, Darryal Chandler, Al Wilson. *Optimizing Pilot Planning and Training for Continental Airlines*. s.l. : Interfaces, July–August 2004. Vol. 34, No. 4, pp. 253–264.
16. Holm, Asa. *Manpower Planning in Airlines- Modeling and Optimization*. Goteborg : s.n., August 2008.
17. McLaughlin, J. Ryan. *OPTIMIZING ADVERSARY TRAINING AND THE STRUCTURE OF THE NAVY ADVERSARY FLEET*. MONTEREY, CALIFORNIA : s.n., September 2013.
18. Gamache, Michel, et al. The Preferential Bidding System at Air Canada. *Transportation Science*, 32 (3). 1998, pp. 246-255.
19. Thengvall, Benjamin Glover and Pachon, Julian Enrique. *Integrated Decision Support System for Optimizing the Training and Transition of Airline Pilots*. 7,346,528 B2 United States, 2008.
20. Hábel, Reka. Pilot Training Optimization for Airlines. *Master thesis, Eötvös Loránd University, Budapest, Hungary*. 2013.
21. Θεοδώρα, Πανταζή. *ΜΕΙΚΤΟΣ ΑΚΕΡΑΙΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΓΙΑ ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΙΠΤΑΜΕΝΩΝ ΣΕ ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΣΥΝΕΔΡΙΕΣ ΠΡΟΣΟΜΟΙΩΤΗ*. Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, Βόλος : s.n., 2013.
22. ΔΕΣΠΟΙΝΑ, ΜΑΥΡΑΤΖΑ. *ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΑΝΑΘΕΣΗΣ ΙΠΤΑΜΕΝΩΝ ΣΕ ΕΚΠΑΙΔΕΥΤΙΚΕΣ ΣΥΝΕΔΡΙΕΣ ΜΕ ΧΡΗΣΗ ΜΕΙΚΤΟΥ ΑΚΕΡΑΙΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ*. Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, Βόλος : s.n., 2014.