

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ



**Μοντέλο Ακέραιου Προγραμματισμού για Βέλτιστη  
Ανάθεση Αδειών Ιπταμένων Αεροπορικών Εταιρειών με  
Βάση Κανόνες Αυστηρής Αρχαιότητας και Προτεραιότητας**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΣΟΥΛΑ ΣΕΙΕ**

**Βόλος 2022**

**Μοντέλο Ακέραιου Προγραμματισμού για Βέλτιστη  
Ανάθεση Αδειών Ιπταμένων Αεροπορικών Εταιρειών  
με Βάση Κανόνες Αυστηρής Αρχαιότητας και  
Προτεραιότητας**

Διπλωματική Εργασία της Σούλα Σέιε

Επιβλέπων Καθηγητής: Κοζανίδης Γεώργιος

Τριμελής εξεταστική επιτροπή:

1<sup>ο</sup> Μέλος

Γεώργιος Κοζανίδης  
Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών  
Πανεπιστήμιο Θεσσαλίας

2<sup>ο</sup> Μέλος

Δημήτριος Παντελής  
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών  
Πανεπιστήμιο Θεσσαλίας

3<sup>ο</sup> Μέλος

Γεώργιος Λυμπερόπουλος  
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών  
Πανεπιστήμιο Θεσσαλίας



## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου κ. Γεώργιο Κοζανίδη που ήταν πάντα διαθέσιμος όταν χρειαζόμουν τη γνώμη του και την καθοδήγηση του όσον αφορά την εκπόνηση της διπλωματικής εργασίας. Η υποστήριξη, η υπομονή του και οι τεχνικές συμβουλές του ήταν πολύτιμες για την ολοκλήρωση αυτής της εργασίας.

Επιπλέον, ένα ευχαριστώ στους καθηγητές Παντελή Δημήτριο και Λυμπερόπουλο Γεώργιο, όντας μέλη της εξεταστικής επιτροπής αφιέρωσαν τον χρόνο στην μελέτη της εργασίας μου.

Στους γονείς μου, σας ευχαριστώ για τη συνεχή υποστήριξη με κάθε δυνατό τρόπο. Θα ήθελα να ευχαριστήσω τη μητέρα μου, που αποτελεί το πρότυπο μου και συνεχίζει να μου δείχνει πως όλα είναι δυνατά. Στα αδέρφια μου, Μανταλένα και Μιχάλη, που συνεχίζουν να πιστεύουν σε εμένα και να με ενθαρρύνουν, ευχαριστώ για την ατελείωτη αγάπη σας.

Τέλος, θα ήθελα να εκφράσω την εκτίμησή μου στους φίλους Ναταλία, Γιώτα, Μιχαέλα, Φίλιπο, Ζωή και Μαρία και σε όλους τους φίλους μου, που ήταν δίπλα μου και με εμπύχωναν κατά την διάρκεια της εργασίας και των σπουδών μου. Θα κρατήσω τις προηγούμενες εμπειρίες μας στην καρδιά μου και ανυπομονώ για νέες.

Σούλα Σέιε

## **Abstract**

Modern air transport systems are complex networks that are critical to global travel and trade. As the demand for air transport has increased, so have flight delays, the associated environmental impacts as well as the need for better internal organization of flight crew members. With the expected further increase in demand, we need new control techniques, and perhaps even redesign of some parts of the already existing systems, to avoid delays, excessive pollution and achieve a smooth operation of airlines.

In this thesis, we solve a major problem that concerns most airlines, that of the optimal and fair assignment of leave to flight personnel (Vacation Management System-VMS). Having considered the personal preferences of each crew staff member, we are asked to respect some rules. Rules will be met independently for each crew member before we can assign him or her a day off.

We achieve the solution of the above problem by creating a mixed integer optimization algorithm as well as using the C programming language.

## Περίληψη

Τα σύγχρονα συστήματα αεροπορικών μεταφορών είναι πολύπλοκα δίκτυα που είναι κρίσιμα για τα παγκόσμια ταξίδια και το εμπόριο. Καθώς η ζήτηση για αεροπορικές μεταφορές έχει αυξηθεί, παράλληλα αυξάνονται οι καθυστερήσεις πτήσεων, οι συνακόλουθες περιβαλλοντικές επιπτώσεις καθώς και η ανάγκη για καλύτερη εσωτερική οργάνωση των μελών του στόλου. Με την αναμενόμενη περαιτέρω αύξηση της ζήτησης, χρειαζόμαστε νέες τεχνικές ελέγχου, και ίσως ακόμη και επανασχεδιασμό ορισμένων τμημάτων των ήδη υπάρχων συστημάτων, προκειμένου να αποφευχθούν οι καθυστερήσεις, η υπερβολική ρύπανση και να επιτύχουμε ομαλή λειτουργία των αεροπορικών εταιριών.

Στην παρούσα διπλωματική εργασία, επιλύουμε ένα μείζον πρόβλημα που απασχολεί τις περισσότερες αεροπορικές εταιρίες, αυτό της βέλτιστης και δίκαιης ανάθεσης αδειών σε μέλη πληρώματος (Vacation Management System-VMS). Έχοντας λάβει υπόψιν τις προσωπικές προτιμήσεις του κάθε μέλους του πληρώματος, καλούμαστε να υπακούσουμε σε κάποιους περιορισμούς. Οι περιορισμοί θα πρέπει να τηρηθούν ανεξαρτήτως για τον καθένα για να μπορέσουμε να του αναθέσουμε κάποια μέρα σαν άδεια.

Επιτυγχάνουμε την επίλυση του παραπάνω προβλήματος με τη δημιουργία ενός αλγορίθμου βελτιστοποίησης, μικτού ακέрайου προγραμματισμού, καθώς και με τη χρήση της γλώσσας προγραμματισμού C.

# ΠΕΡΙΕΧΟΜΕΝΑ

1.1 Κίνητρο	9
1.2 Σκοπός	10
1.3 Βιβλιογραφική Ανασκόπηση	11
1.4 Δομή	12
2.1 Η Εναέρια Μεταφορά στο πέρασμα των χρόνων	13
2.2 Η εξέλιξη των Αερομεταφορών με τη βοήθεια της Επιχειρησιακής Έρευνας	16
2.3 Αδειοδότηση στα Μέλη του Πληρώματος (Vacation Management System)	19
3.1 Μικτός Ακέραιος Προγραμματισμός	21
3.2 Γλώσσα Προγραμματισμού «C»	22
4.1 Ανάλυση του προβλήματος	23
4.2 Περιγραφή του Λογισμικού και Κώδικα που χρησιμοποιήθηκε	24
4.2.1 Περιγραφή Λογισμικού	24
4.2.2 Κατασκευή του Κώδικα	24
5.1 Περιγραφή Μεταβλητών και Παραμέτρων	29
5.2 Θεωρητική Ανάλυση της Αντικειμενικής Εξίσωσης και των Περιορισμών	30
6.1 Παρουσίαση προβλήματος μικρού σκέλους με εισαγωγή αρχείου	33
6.2 Λύση του προβλήματος μικρού σκέλους με εισαγωγή αρχείου	40
6.3 Παρουσίαση προβλήματος μικρού σκέλους με τυχαία γεννήτρια	41
6.4 Δοκιμές και πειράματα	44
7.1 Συμπεράσματα	46
7.2 Μελλοντικοί Στόχοι	47
Βιβλιογραφία	48
ΠΑΡΑΡΤΗΜΑ	50
ΥΠΟΛΟΓΙΣΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ	50

## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Αριθμός πτήσεων από το 2004-2022 .....	9
Εικόνα 2: Σχέδιο Da Vinci 2 .....	13
Εικόνα 3: Σχέδιο Da Vinci 1 .....	13
Εικόνα 4: Πρώτη επιτυχημένη πτήση των αδερφών Wright.....	14
Εικόνα 5: Πρώτη εμπορική πτήση του F 13 Junkers.....	15
Εικόνα 6: HUB & SPOKE Approach.....	17
Εικόνα 7: Από <<Σημείο σε Σημείο >> .....	17
Εικόνα 8 Παράδειγμα φόρμας στην οποία το προσωπικό δηλώνει τις προτιμήσεις του [16] .....	23
Εικόνα 9: Header Files του κώδικα .....	25
Εικόνα 10: Διάγραμμα ροής για τον έλεγχο Strict Priority.....	27
Εικόνα 11: Διάγραμμα ροής Ελέγχου Αδειών .....	28
Εικόνα 12: Παράδειγμα περιγραφής ιεραρχίας προσωπικού και επιθυμιών .....	31
Εικόνα 13: Παράδειγμα περιορισμών limit line .....	32
Εικόνα 14: Περιπτώσεις που οι μέρες θεωρούνται overlapped .....	32
Εικόνα 15: Πίνακας Δεδομένων.....	34
Εικόνα 16: Αρχείο δεδομένων που εισάγεται από τον χρήστη.....	35
Εικόνα 17: Δηλώσεις μεταβλητών στον κώδικα.....	35
Εικόνα 18: Αρχείο που έχουν εκτυπωθεί οι περιορισμοί .....	37
Εικόνα 19: Περιορισμοί Max Number of Bids .....	38
Εικόνα 20: Περιορισμοί Entitlement .....	38
Εικόνα 21: Περιορισμοί Overlapping.....	38
Εικόνα 22: Περιορισμός Separation .....	38
Εικόνα 23: Ψευδοκώδικας λογικής Separation .....	39
Εικόνα 24: Περιορισμοί Limit Line .....	39
Εικόνα 25: Περιορισμοί που συνδυάζουν Xij, Yij.....	40
Εικόνα 26: Λύση του προβλήματος .....	40
Εικόνα 27: Τυχαία δεδομένα από την γεννήτρια .....	42
Εικόνα 28: Λόγοι αποδοχής των Bids .....	44
Εικόνα 29: Λόγοι απόρριψης των Bids .....	45

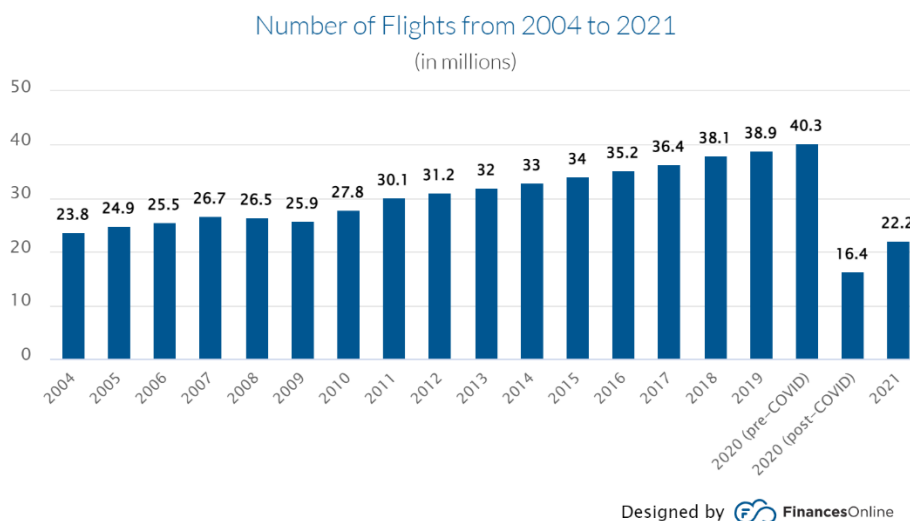
## Κεφάλαιο 1<sup>ο</sup>: Εισαγωγή



## 1.1 Κίνητρο

Η δημιουργία αποτελεσματικών προγραμμάτων ανάθεσης αδειών για το πλήρωμα είναι μεγάλης σημασίας για τις αεροπορικές εταιρείες. Είναι γνωστό πως ότι «Δίπλα στο κόστος των καυσίμων, το κόστος του πληρώματος αντιπροσωπεύει τον μεγαλύτερο ενιαίο παράγοντα κόστους για τις αεροπορικές εταιρείες». Φυσικά, η αποτελεσματική χρήση των πόρων, η μείωση των λειτουργικών εξόδων και η υψηλότερη ευρωστία είναι στόχοι που κάθε κερδοφόρα αεροπορική εταιρεία θέτει.

Ωστόσο, δυστυχώς η αύξηση της ζήτησης των πτήσεων τα τελευταία χρόνια έχει οδηγήσει σε υψηλότερα λειτουργικά κόστη και μικρότερη ευρωστία [1,12]. Σύμφωνα με το IATA (2021) ο αριθμός των εσωτερικών και παγκόσμιων πτήσεων ανήλθε στα 22,2 εκατομμύρια το 2021. Μετά από μια σταθερή ανοδική τάση του 3.2% κάθε χρόνο, εμφανίστηκε ο COVID-19 και προκάλεσε απότομη πτώση 46,77%, μέχρι και τον Φεβρουάριο του 2021 (OAG, 2021). Αυτό είχε ως αποτέλεσμα καθαρή ζημιά ύψους 119,5 δισεκατομμυρίων ευρώ το 2020 και συνεχίστηκε το 2021, η οποία έχει προβλεπόμενη καθαρή ζημιά 16,8 δισεκατομμυρίων ευρώ μέχρι το τέλος του έτους (IATA, 2020).



Εικόνα 1: Αριθμός πτήσεων από το 2004-2022

Δεδομένου ότι η βασική υπηρεσία που παρέχουν οι αεροπορικές εταιρείες είναι αυτή της μεταφοράς, είτε επιβατών είτε φορτίων, η επιτυχής λειτουργία τους εξαρτάται από τη συνδρομή των παραγόντων της συνέπειας, της συχνότητας των πτήσεων, της υψηλής ποιότητας της παρεχόμενης υπηρεσίας καθώς και της τιμολογιακής πολιτικής που ακολουθούν.

Έτσι λοιπόν, η τήρηση του προγράμματος αναχωρήσεων και αφίξεων, η πρόταξη της ασφάλειας, οι ικανοποιητικές υπηρεσίες κατά τη διάρκεια των πτήσεων από τα μέλη του πληρώματος και η προσφορά εισιτηρίων σε προσιτές τιμές καθιστούν μια αεροπορική εταιρία πετυχημένη και ελκυστική για το καταναλωτικό κοινό.

Γι' αυτό το λόγο οι αεροπορικές εταιρίες στρέφονται σε προηγμένες μεθόδους βελτιστοποίησης, προκειμένου να αναπτύξουν κατάλληλα συστήματα υποστήριξης αποφάσεων που αφορούν στον πληρέστερο έλεγχο και στη διαχείριση διαδικασιών ούτως ώστε να αποκτήσουν σημαντικό ανταγωνιστικό πλεονέκτημα στην αγορά. Η ανάπτυξη ενός συστήματος συντήρησης αεροσκαφών είναι μια περίπλοκη εργασία που περιλαμβάνει τη σύνθεση μιας σειράς παραγόντων, όπως οικονομικών, πολιτικών, νομικών και τεχνικών. Προκειμένου λοιπόν, να διασφαλίσουν υπηρεσίες υψηλής ποιότητας αλλά με το ελάχιστο δυνατό κόστος, τόσο για τις ίδιες όσο και για τους πελάτες τους, οι αεροπορικές εταιρίες έχουν δαπανήσει σημαντικά κεφάλαια στον κλάδο της επιχειρησιακής έρευνας για την εύρεση αποδοτικών τρόπων οργάνωσης με σκοπό την πετυχημένη και κερδοφόρα παρουσία και λειτουργία τους στον ανταγωνιστικό χώρο των εναέριων μεταφορών.

## 1.2 Σκοπός

Στα πλαίσια της παρούσας διπλωματικής εργασίας, καταβάλλεται προσπάθεια επίλυσης ενός σημαντικού λειτουργικού ζητήματος που αντιμετωπίζουν οι αεροπορικές εταιρίες σήμερα, το ζήτημα ανάθεσης αδειών στα μέλη του πληρώματος τους. Σήμερα πολλές εταιρίες χρησιμοποιούν ειδικές φόρμες, στις οποίες το προσωπικό μπορεί να δηλώσει τις προτιμήσεις τους προτιμήσεις για τις ημέρες αδειών που δικαιούνται. Έπειτα, εξετάζουμε τον τρόπο με τον οποίο ελέγχονται τα αιτήματα αυτά, βάση της πολιτικής που εφαρμόζουν οι αεροπορικές εταιρίες και την ανάθεση αυτών που θεωρούνται εφικτά.

Είναι σημαντικό να αναφερθεί, πως οι αεροπορικές εταιρίες επιβαρύνονται έξτρα κόστη σε περίπτωση που κάποιο μέλος του στόλου δεν εξαντλήσει το συνολικό αριθμό των ημερών που δικαιούται. Αναλαμβάνουμε να κατασκευάσουμε κατάλληλο μοντέλο βελτιστοποίησης, το οποίο θα αναθέτει τις ημέρες που υπολείπονται σε κάθε εργαζόμενο με τέτοιο τρόπο, ώστε να μη παραβιάζονται και πάλι οι κανόνες που καθιστούν μία άδεια εφικτή.

Μέσω της μελέτης της σχετικής βιβλιογραφίας και των μοντέλων βελτιστοποίησης που έχουν αναπτυχθεί με την πάροδο του χρόνου από τον κλάδο της Επιχειρησιακής Έρευνας, η μέθοδος μικτού ακέραιου προγραμματισμού είναι η πιο ευρέως διαδεδομένη και με αυτόν τον τρόπο αναλύεται η διαδικασία επίλυσης του συγκεκριμένου προβλήματος.

### 1.3 Βιβλιογραφική Ανασκόπηση

Οι αεροπορικές εταιρείες αντιμετωπίζουν σειρά προκλήσεων στην συνεχή προσπάθεια τους να εξασφαλίσουν υψηλή απόδοση στις λειτουργίες τους. Τα προβλήματα των αεροπορικών εταιριών είναι πολύπλοκα, όπως αυτή της βέλτιστης ανάθεσης πτήσεων ή πληρώματος, ελαχιστοποίηση κόστους και φυσικά η βέλτιστη ανάθεση αδειών στους υπαλλήλους. Γι' αυτό το λόγο, πολλοί ερευνητές έχουν προσπαθήσει να δημιουργήσουν μοντέλα, που μπορούν να δώσουν λύση στα περισσότερα προβλήματα σε εύλογο χρονικό διάστημα. Συγκεκριμένα, οι Gürkan, Gürel, Aktürk [8] πρότειναν ένα μη γραμμικό μικτού ακέραιου προγραμματισμού μοντέλο για δημιουργία πρόγραμμα πτήσεων, ανάθεση στόλου, όπου για πρώτη φορά συνυπολογίζεται ο χρόνος πτήσεως, οι M. Ben Ahmeda, F. Zeghal Mansourb και M. Haouaric [2] προσπάθησαν να λύσουν την ανάθεση στόλου και την βέλτιστη δημιουργία προγράμματος πτήσεων σε ένα μοντέλο μη γραμμικού μικτού ακέραιου προγραμματισμού, πολυωνυμικού μεγέθους. Ένα σημαντικό μειονέκτημα της λύσης αυτής είναι ότι αγνοεί τις περισσότερες αλληλεξαρτήσεις των δύο προβλημάτων. Πολύ σημαντική αποτελεί και η έρευνα των İ. Zeki Akyurt, Y. Kuvvetli, M. Deveci, H. Garg και M. Yuzsever [9], οι οποίοι πρότειναν ένα μοντέλο μικτού ακέραιου προγραμματισμού, συμπεριλαμβάνοντας πληθώρα περιορισμών, όπως τις προσωπικές άδειες των ιπταμένων. Παρόμοιο μοντέλο πρότειναν και οι J. Soriano, E. Rex Jalao, I. A. Martinez [10], αν και το επίκεντρο έρευνας ήταν αποκλειστικά οι εργαζόμενοι και η βέλτιστη κατανομή εργασίας, αδειών και διαλλειμάτων. Όπως παρατηρούμαι, πολλοί ερευνητές προσπάθησαν να δημιουργήσουν ένα μοντέλο το οποίο λύνει συλλογικά όλα τα προβλήματα που μπορεί να αντιμετωπίζει μια εταιρία. Στη παρούσα διπλωματική εργασία καθώς και στις εργασίες προηγούμενων συναδέλφων, όπως της Α. Μαυριδοπούλου [13] και των Π. Ρίζου, Π. Τσιατά [15], γίνεται προσπάθεια δημιουργίας μοντέλου το οποίο εστιάζει αποκλειστικά στη βέλτιστη ανάθεση αδειών στους ιπταμένους, μέσω τεχνικές ακέραιου προγραμματισμού.

## 1.4 Δομή

Αρχικά στο Κεφάλαιο 2 γίνεται μια ιστορική ανασκόπηση στην επιθυμία του ανθρώπου για πτήση, καθώς τις σημαντικές προσπάθειες που γίνανε ανά τα χρόνια για την επίτευξη τους. Έπειτα, γίνεται αναφορά στην εξέλιξη του κλάδου σχετικά με τα χαρακτηριστικά γνωρίσματα των εναέριων μεταφορών και των σημαντικότερων προβλημάτων που αντιμετωπίζει. Στο Κεφάλαιο 3 γίνεται μια σύντομη αναφορά στην μέθοδο του προβλήματος καθώς και στην γλώσσα προγραμματισμού που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία.

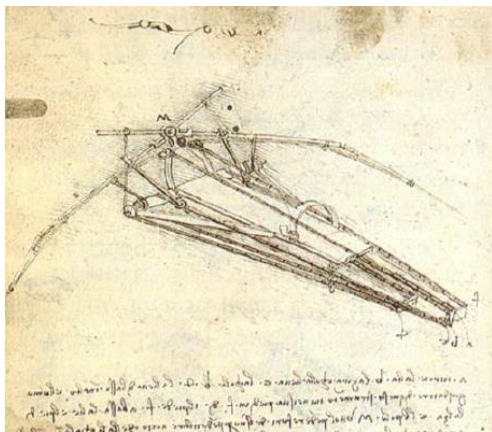
Ακολούθως, στο Κεφάλαιο 4 γίνεται αναλυτική περιγραφή της διαδικασίας που ακολουθήθηκε για την κατασκευή κατάλληλου κώδικα για την ορθή ανάθεση των αδειών που ζητούνται από τα μέλη του στόλου. Στο Κεφάλαιο 5, παρουσιάζεται η μαθηματική μοντελοποίηση του προβλήματος θεωρητικά, αναλύοντας διεξοδικά τις παραμέτρους και τους περιορισμούς. Έχοντας κατανόηση πλήρως το πρόβλημα, περνάμε στο Κεφάλαιο 6 όπου παρουσιάζεται ένα πρόβλημα μικρού σκέλους λυμένο είτε με εισαγωγή δεδομένων από τον χρήστη, είτε μέσω τυχαίας γεννήτριας αριθμών.

Τέλος, στο Κεφάλαιο 7 παρατίθενται τα συνολικά συμπεράσματα τα οποία εξήχθησαν από την εκπόνηση της παρούσας διπλωματικής εργασίας και οι μελλοντικοί στόχοι που θέτονται και τη βελτίωσή της.

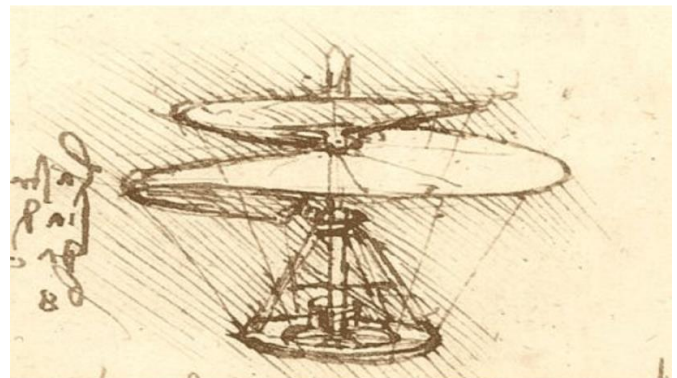
## Κεφάλαιο 2<sup>ο</sup> : Ιστορική Αναδρομή

### 2.1 Η Εναέρια Μεταφορά στο πέρασμα των χρόνων

Η ιδέα ότι ο άνθρωπος μπορεί να πετάει και να βρίσκεται στον αέρα, είναι όσο παλιά είναι και ο ανθρώπινος πολιτισμός μας. Ανά τους αιώνες αποπειράθηκαν πολλοί να κάνουν αυτή την ιδέα πραγματικότητα. Ένας από αυτούς ήταν και ο Leonardo Da Vinci[3]. Είχε κάνει την υπόθεση πως ο άνθρωπος θα μπορούσε να βρίσκεται στον αέρα με την βοήθεια μηχανικών φτερών, τα οποία θα ήταν σχεδιασμένα να αφηφούν την αντίσταση του αέρα. Υπάρχουν πολλά σχέδια του που δείχνουν διαφορετικές μηχανές. Μερικά δείχνουν έναν άνθρωπο, ξαπλωμένο οριζόντια στη συσκευή και τα χέρια του και τα πόδια είναι σταθερά ή ελέγχουν ορισμένα καλώδια/ράβδους και άλλα σχέδια δείχνουν έναν άνθρωπο να ελέγχει μηχανισμούς πτερυγίων που μετακινεί τα φτερά πάνω-κάτω (βλ. σχήμα 2 και 3). Επίσης, ανάμεσα στα πολλά σχέδια του έχει βρεθεί μέχρι και ένα αλεξίπτωτο σύστημα.



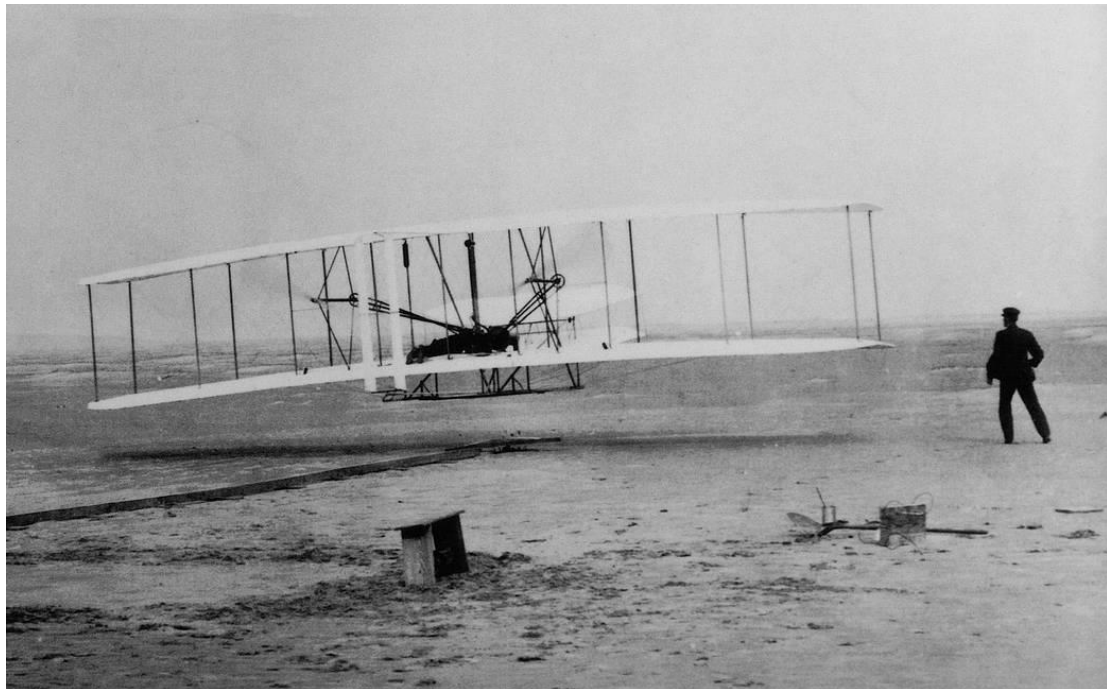
Εικόνα 3: Σχέδιο Da Vinci 1



Εικόνα 2: Σχέδιο Da Vinci 2

Προφανώς, οι προσπάθειες συνεχίστηκαν μέχρι και τις αρχές του 19ου αιώνα, χωρίς κάποια επιτυχία. Θα έπρεπε να περιμένουμε μέχρι της 17 Δεκεμβρίου του 1903, τα αδέρφια Ράιτ, Όρβιλ και Γουίλμπουρ. Ήταν δύο Αμερικανοί που εφηύραν και κατασκεύασαν το πρώτο επιτυχημένο αεροπλάνο στον κόσμο. Μπορεί η πτήση να διήρκεσε μόλις 12 δευτερόλεπτα, αλλά ήταν ικανή για να αλλάξει τη λογική των μεταφορών για πάντα. Το αεροπλάνο που κατασκευάστηκε ονομαζόταν Flyer 1 (Αεροπόρος 1), ήταν φτιαγμένο από ξύλο και ζύγιζε 341 κιλά μαζί με τον πιλότο. Το μήκος του ήταν 6.5 μέτρα και το άνοιγμα των φτερών του 12.3 μέτρα. Η πρώτη απόπειρα πτήσης έγινε στις 14 Δεκεμβρίου από τον Γουίλμπερ Ράιτ, η οποία όμως ήταν αποτυχημένη. Φυσικά, 3 ημέρες μετά, ο άλλος αδερφός, ο Όρβιλ Ράιτ

απογείωσε το αεροπλάνο αυτοδύναμα και μετά από 12 δευτερόλεπτα το προσγείωσε κανονικά με αποτέλεσμα να ολοκληρώσει επιτυχημένα την πτήση (βλ. εικόνα 4).



Εικόνα 4: Πρώτη επιτυχημένη πτήση των αδερφών Wright

Μέσα σε μόλις 10 χρόνια, αναπτύχθηκαν πολλές νέες ιπτάμενες μηχανές, με διαφορετικές διαμορφώσεις ουράς, πολλαπλά πτερύγια, διαφορετικές προωθητικές μηχανές κινητήρες και ενσωματώσεις κινητήρων.

Παρόλα αυτά, τα αεροσκάφη θα χρησιμοποιούνταν για εμπορικές πτήσεις αρκετά χρόνια αργότερα. Μετά το τέλος του Α' Παγκοσμίου Πολέμου, ο Χιούγκο Τζάνκερς, δημιουργός του Jankers F 13, το πρώτο μεταφορικό αεροσκάφος από μέταλλο στον κόσμο, ήταν ένα προηγμένο μονοπλάνο με πρόβολο, το οποίο μπορούσε να φιλοξενήσει τέσσερις επιβάτες, είχε το όραμα ότι τα αεροσκάφη θα μπορούσαν να χρησιμοποιηθούν σαν μέσο μεταφοράς.

Το F13 είχε όλα τα τυπικά χαρακτηριστικά ενός σημερινού αεροσκάφους (βλ εικόνα 5). Είχε ένα μόνο φτερό πρόβολου, μια κλασική ουρά, δύο κινητήρες με έλικες, και μια καμπίνα ατράκτου. Έτσι μόνο 16 χρόνια μετά την πρώτη πτήση από τους αδελφούς Ράιτ, μια σχεδόν τέλεια δημιουργία για τις αεροπορικές μεταφορές είχε ήδη αναπτυχθεί με όλα τα τυπικά χαρακτηριστικά ενός μεταφορικού αεροσκάφους, όπως γνωρίζουμε σήμερα:

- μια ενιαία καμπίνα,
- μπροστινό πιλοτήριο,
- μια άτρακτος για την υποδοχή του ωφέλιμου φορτίου
- μια κλασική ουρά για έλεγχο και σταθερότητα
- ένας κινητήρας τοποθετημένος μπροστά από την άτρακτο

Δυστυχώς, το F13 δεν αποτέλεσε επιτυχία καθώς η αγορά δεν ήταν έτοιμη για να υποδεχθεί ένα τέτοιο μηχάνημα.



Εικόνα 5: Πρώτη εμπορική πτήση του F 13 Junkers

Μια μεγάλη ώθηση για τις αεροπορικές μεταφορές ξεκίνησε το 1925 στις ΗΠΑ όπου η κυβέρνηση αποφάσισε να εντάξει ιδιωτικούς οργανισμούς αερομεταφορών για τη μεταφορά δεμάτων, με κινητήριο μοχλό την μείωση του κόστους αλλά και του χρόνου. Καθώς η εξέλιξη του κλάδου δε σημείωνε κάποια σημαντική πρόοδο, κατά τον Β' Παγκόσμιο Πόλεμο, μέχρι και μετά το τέλος του πολέμου, στο τέλος της δεκαετίας του 1950, άρχισε η παροχή υπηρεσιών από αεροσκάφη χωρίς έλικες. Σημειώνεται ότι, τα αεροσκάφη αυτά παρείχαν μεγαλύτερη άνεση, αυτονομία και ταχύτητα στις πτήσεις τους. Ακολούθως, το 1960, οι Eastern Airlines των ΗΠΑ, ξεκίνησαν τα πρώτα τακτικά δρομολόγια στο βορειοανατολικό τμήμα των ΗΠΑ. Ο κόσμος μπορούσε πλέον να πραγματοποιήσει κανονικά πτήση με αεροπλάνο. Με την πάροδο των χρόνων ο κλάδος συνέχισε να αναπτύσσεται και όλο ένα και περισσότερες εταιρίες εντάσσονταν σε αυτόν. Μεταξύ του 1945 και του 2000 το ποσοστό των ανθρώπων που χρησιμοποιούν το αεροπλάνο αυξήθηκε περίπου 12%, ενώ μεταξύ του 1960 και του 2000 περίπου 9%. Φυσικά, η εξέλιξη και η ανάπτυξη του κλάδου δεν ήταν ίδια σε όλο τον κόσμο. Η ανάπτυξη του κλάδου στην Αμερική και στην Ευρώπη ήταν εμφανώς μεγαλύτερη σε σχέση με αυτή της Αφρικής και της Μέσης Ανατολής.

Ερχόμενοι στο σήμερα, η πραγματικότητα μοιάζει πολύ διαφορετική. Εκτιμάτε ότι καθημερινά πραγματοποιούνται 151.435 πτήσεις, ενώ παγκοσμίως ο αριθμός των ατόμων που πραγματοποίησε έστω και μια πτήση ήταν 4.5 δισεκατομμύρια ετησίως. Τέλος αξίζει να αναφερθεί πως τα αεροπλάνα που μεταφέρουν δέματα και γενικότερα φορτία, ανέρχονται στα 25.368 και αναμένεται 3.4% ετήσια αύξηση τους.

## 2.2 Η εξέλιξη των Αερομεταφορών με τη βοήθεια της Επιχειρησιακής Έρευνας

Με την εξέλιξη του κλάδου όπως είδαμε, αυξήθηκε η ζήτηση σε πτήσεις, προσωπικό, αεροσκάφη και φυσικά δημιουργήθηκε η ανάγκη για καλύτερη οργάνωση των εταιριών. Οι αεροπορικές εταιρίες ανά τον κόσμο αντιμετωπίζουν παρόμοια προβλήματα, και γι' αυτό έγινε εφικτό να δοθεί μια κοινή λύση που θα βοηθούσε οικονομικά. Έτσι, το 1961 ιδρύθηκε ένας οργανισμός που στόχευε στη μελέτη και λύση τέτοιων προβλημάτων μέσω της επιχειρησιακής έρευνας, η «AGIFORS». Η «AGIFORS» (Airline Group of International Federation of Operational Research Societies) είναι ένας οργανισμός που αποτελείται από επαγγελματίες της επιχειρησιακής έρευνας που απασχολούνται με την βελτίωση της οργάνωσης των αεροπορικών εταιριών, και έχει ως στόχο της, την πρόοδο της επιστήμης στις αεροπορικές μεταφορές.

Ο κλάδος της επιχειρησιακής έρευνας, μέσω κατάλληλων αλγορίθμων, τεχνικών βελτιστοποίησης και συστημάτων υποστήριξης αποφάσεων λύνει μερικά από τα ζητήματα που προκύπτουν κατά τη λειτουργία μιας αεροπορικής εταιρίας και τα μοντέλα βελτιστοποίησης χρησιμοποιούνται ως σημαντικό εργαλείο για την επίλυσή τους [14].

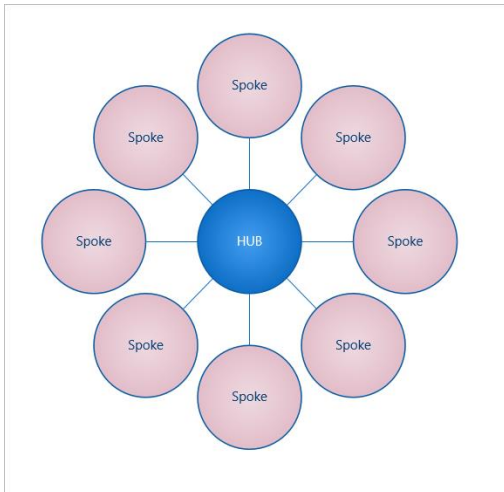
Κάποια από αυτά είναι :

- Σχεδιασμός δικτύου και κατασκευή προγράμματος πτήσεων

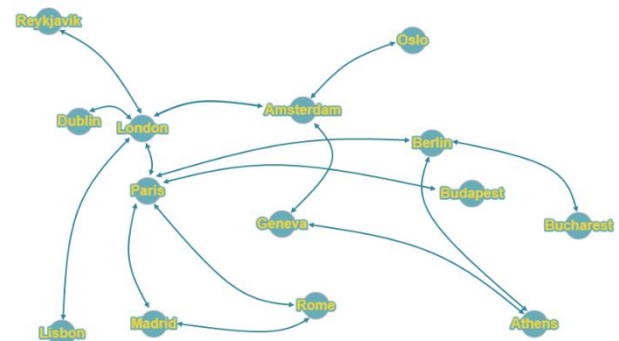
Ένα από τα σημαντικότερα προβλήματα που καλείται να λύσει μια αεροπορική εταιρία είναι αυτή του σχεδιασμού δικτύου, των επίγειων δηλαδή σταθμών ανάμεσα στους οποίους θα κινείται. Τα τελευταία χρόνια παρατηρούμαι μια αλλαγή στο τρόπο σχεδιασμού των δικτύων λόγω των απαιτήσεων της αγοράς. Οι εταιρείες είναι πιο ελεύθερες στην επιλογή προορισμών, και το μοντέλο σχεδιασμού που επικρατούσε παλιότερα <<από σημείο σε σημείο>>, έχει αντικατασταθεί από πρότυπο ενός κεντρικού σταθμού και διαφόρων περιφερειακών σταθμών που βρίσκονται σε μια ακτίνα γύρω από τον κεντρικό (πρότυπο Hub and Spoke) ανάλογα με την εμβέλεια των δυνατοτήτων της αεροπορικής εταιρίας.

Αφού κάθε εταιρία σχεδιάσει το δίκτυο της, η κατασκευή του προγράμματος πτήσεων θα λάβει χώρα. Το σύνολο των προορισμών που θα εξυπηρετεί κάθε αεροπορική είναι πρωταρχικής σημασίας για την ομαλή και κερδοφόρα λειτουργία τους. Το πρόγραμμα των πτήσεων προσδιορίζει ένα σύνολο από ανταποκρινόμενους σταθμούς αναχώρησης και άφιξης τους οποίους εξυπηρετεί η κάθε αεροπορική εταιρία καθώς και τις ακριβείς ώρες αναχώρησης και άφιξης για κάθε συνδυασμό αυτών. Η απόφαση της αεροπορικής για το ποιους σταθμούς θα εξυπηρετεί, εξαρτάται σε μεγάλο βαθμό από τις απαιτήσεις της εκάστοτε αγοράς.





Εικόνα 6: HUB & SPOKE Approach



Εικόνα 7: Από <<Σημείο σε Σημείο >>

- Ανάθεση αεροσκαφών σε πτήσεις

Ο προγραμματισμός ανάθεσης των αεροσκαφών γίνεται στην πραγματικότητα αρκετό καιρό, ίσως και μήνες πριν την πρακτική εφαρμογή του. Έτσι, δίνεται το περιθώριο ανακατανομής και βελτίωσης του αποτελέσματος λαμβάνοντας κάθε φορά τις ισχύουσες πραγματικές συνθήκες. Το πρόγραμμα των πτήσεων παραμένει συνήθως σταθερό και επαναλαμβάνεται περιοδικά.

Για την βελτιστοποίηση του συγκεκριμένου προβλήματος χρησιμοποιούμε μεθόδους που αναπαριστούν το πρόγραμμα των πτήσεων ως χωροχρονικό δίκτυο ροής, με τόξα ανάμεσα στους διάφορους σταθμούς και μια λίστα πιθανών πτήσεων (τόξων) σε κάθε σταθμό. Οι διάφορες αναχωρήσεις και αφίξεις σε κάθε σταθμό αναπαρίστανται ως κόμβοι. Δεύτερος τρόπος ανάθεσης αεροσκαφών σε πτήσεις αποτελεί δημιουργία ενός δικτύου συνδέσεων, όπου οι κόμβοι είναι οι πτήσεις και τα τόξα η σύνδεση μεταξύ των πτήσεων. Οι πιθανοί τρόποι σύνδεσης των πτήσεων μεταξύ τους είναι πολλοί και συχνά οδηγούμαστε σε μεγαλύτερης κλίμακας δίκτυα από ότι στην περίπτωση του χωροχρονικού δικτύου.

- Ανάθεση πτήσεων σε αεροσκάφη

Η βέλτιστη ανάθεση των πτήσεων στα αεροσκάφη αποτελεί σημαντικό βήμα για την ομαλή εξέλιξη μιας πτήσης και φυσικά ολόκληρης της εταιρίας. Η ανάθεση πρέπει να γίνεται με τέτοιο τρόπο ώστε τα δρομολόγια να υπακούν σε κανόνες όπως, τα αεροπλάνα να περνάνε από την τακτική συντήρηση τους και η διανομή των πτήσεων να είναι εξισορροπημένη σε διάφορα αεροσκάφη.

Η σωστή διανομή των πτήσεων, εξασφαλίζει ότι όλα τα αεροσκάφη θα χρησιμοποιηθούν εξίσου κατά την λειτουργία του προγράμματος της αεροπορικής εταιρίας, ώστε να μην επιφορτίζεται μόνο μέρος του στόλου, πράγμα που οδηγεί

σε αυξανόμενα κόστη συντήρησης καθώς και αυξανόμενα διαφυγόντα κέρδη εκμετάλλευσης του τεχνικού εξοπλισμού της εταιρίας.

- **Ανάθεση πληρωμάτων σε πτήσεις**

Ο τρόπος με τον οποίαν γίνεται κατανομή του πληρώματος στις διαθέσιμες πτήσεις, εξαρτάται τόσο από νομικούς ή συμβατικούς περιορισμούς, που έχουν να κάνουν με τον επιτρεπόμενο αριθμό συνεχούς πτήσης, όσο και από φυσικούς, όπως οι διαθέσιμες πτήσεις αλλά και ο αριθμός διαθέσιμος πληρώματος. Οι μέθοδοι που έχουν χρησιμοποιηθεί για την επίλυση και βελτιστοποίηση του προβλήματος βασίζονται σε ένα καθορισμένο πρόβλημα καθορισμού, το οποίο βασίζεται ότι όλες οι πτήσεις παρουσιάζονται σε στήλες, ενώ το διαθέσιμο πλήρωμα σε γραμμές, δημιουργώντας όλους τους πιθανούς συνδυασμούς, με βάση κάποιους περιορισμούς.

Στη συνέχεια, για την οριστικοποίηση αυτών των ταιριασμάτων και χρησιμοποιώντας μαθηματικούς αλγορίθμους, μέθοδος δημιουργίας στηλών (column generation ) καταλήγουμε στις βέλτιστες λύσεις.

- **Ανάθεση Αδειών στο Πλήρωμα**

Η χρήση των δικαιωμάτων άδειας από τους εργαζόμενους έχει θετική και σημαντική επίδραση στην οργάνωση των εταιριών καθώς αποτελεί βασικό στοιχείο στη διαμόρφωση παραγωγικής κουλτούρας του οργανισμού[11]. Έρευνες έχουν δείξει πως ένα καλό σύστημα ανάθεσης αδειών μπορεί να βελτιώσει και να αυξήσει την παραγωγικότητα των υπάλληλων, ειδικά σε αεροπορικές εταιρίες όπου το πρόγραμμα εργασίας μπορεί κάποιες φορές να υπερφορτώνεται.

- **Διαχείριση εισροών**

Η κύρια ροή εισροών για τις αεροπορικές αποτελεί φυσικά η πώληση των εισιτηρίων πτήσεων. Τα εισιτήρια προφανώς, ανάλογα το χρόνο πτήσης, προορισμό, θέση του επιβάτη μέσα στο χώρο του αεροπλάνου, κυμαίνονται σε διαφορετικές τιμές. Σημαντικό εργαλείο για τη διαχείριση του εισοδήματος αποτελεί η διαχείριση του προβλήματος ελέγχου διαθέσιμων θέσεων (seat inventory control problem). Στόχος είναι να βρεθεί ο σωστός συνδυασμός τύπου επιβατών στις πτήσεις έτσι ώστε τα εισοδήματα να μεγιστοποιούνται. Σε αυτό συμβάλλει και η σωστή πρόβλεψη ζήτησης και η αναπροσαρμογή των εταιριών στα νέα δεδομένα.

- **Διαχείριση μη προγραμματισμένων ενεργειών**

Όπως είδαμε παραπάνω, οι εταιρίες σχεδιάζουν και εκδίδουν το πρόγραμμα τους στο κοινό μήνες πριν την πραγματική εφαρμογή του. Πολλές φορές όμως αναπάντεχα περιστατικά μπορεί να συμβούν όπως: προβλήματα έλλειψης δυναμικού, τεχνικές βλάβες ή κάποια ραγδαία επιδείνωση του καιρού. Όλα τα παραπάνω μπορεί να αποτρέψουν τις εταιρίες να ακολουθήσουν το κανονικό τους

πρόγραμμα με αποτέλεσμα να υπάρχουν μεγάλες οικονομικές ζημιές. Όταν τέτοιες καταστάσεις συμβούν, απαιτείται άμεση ανταπόκριση ώστε η αεροπορική να επανέλθει στο κατάλληλο πρόγραμμα της.

Έτσι λοιπόν γίνεται αντιληπτό πως η διαχείριση μη προγραμματισμένων ενεργειών, έχει να κάνει, κατά την διαδικασία επίλυσης της, με τους εξής τομείς μιας αεροπορικής: δρομολόγηση αεροσκαφών, προγραμματισμός πληρώματος και αποκατάσταση των πελατών.

Τα παραπάνω προβλήματα συνήθως επιλύονται με τη σειρά που περιεγράφηκαν καθώς τα ζητούμενα του ενός θα αποτελέσουν δεδομένα του άλλου επιτυγχάνοντας τελικά την ομαλή λειτουργία των εταιριών.

## 2.3 Αδειοδότηση στα Μέλη του Πληρώματος (Vacation Management System)

Για την εύρυθμη λειτουργία των αεροπορικών εταιριών, είναι φυσικά σημαντικό να διατηρείται η ισορροπία μεταξύ της προσωπικής και επαγγελματικής ζωής. Έτσι, μια ακόμη παράμετρος που πρέπει οι εταιρίες να λάβουν υπόψιν είναι ο τρόπος με τον οποίο οι άδειες κατανέμονται στο στόλο.

Η αδειοδότηση ή αλλιώς Vacation Management System (VMS) είναι ουσιαστικά η διαχείριση των διακοπών των εργαζομένων [4]. Οι εταιρίες έχουν να αντιμετωπίσουν πολλές προκλήσεις: αιτήματα της τελευταίας στιγμής, πολλή γραφειοκρατία, ανάγκη συμμόρφωσης με τις πολιτικές της εταιρείας. Όλα αυτά μπορούν να κάνουν την διαδικασία πολύ πολύπλοκη.

Πολλές αεροπορικές έχουν υιοθετήσει ένα σύστημα διαχείρισης αδειών, στο οποίο οι εργαζόμενοι μπορούν να εισέρχονται και να δηλώνουν τις προτιμήσεις τους. Έπειτα, τα αιτήματα τους επεξεργάζονται από αυτοματοποιημένες ροές εργασίας που κατευθύνουν αυτά τα αιτήματα σε αρμόδιους διευθυντές και ειδικούς ανθρώπινου δυναμικού για έγκριση, και ειδοποιεί τους υπαλλήλους για τις αποφάσεις.

Στην εκπόνηση της παρούσας διπλωματικής λάβαμε υπόψιν μας τους παρακάτω περιορισμούς :

- **Maximum Number of Bids**

Ο περιορισμός αυτός επιτρέπει στον κάθε εργαζόμενο να δηλώσει συγκεκριμένο αριθμό προτιμήσεων, ανεξάρτητα φυσικά από το αν και ποιες από αυτές θα εγκριθούν τελικά.

- Entitlement

Κάθε εργαζόμενος έχει συγκεκριμένο αριθμό ημερών που μπορεί να του ανατεθεί και ο οποίος δεν πρέπει να ξεπεραστεί.

- Overlapping

Ο κανόνας αυτός αφορά τις άδειες του ίδιου εργαζομένου. Ένα αίτημα θεωρείται εφικτό εφόσον οι ημερομηνίες λήξης και έναρξης της άδειας δεν επικαλύπτεται με κάποια άλλη δηλωμένη προτίμηση του.

Δηλαδή εάν έχουμε 2 επιθυμίες A,B με startA, startB η έναρξη των αδειών αντίστοιχα και endA, endB η λήξη τους, θα πρέπει ( $StartA > EndB$ ) και ( $EndA < StartB$ ).

- Limit Lines

Είναι σημαντικό να αναφέρουμε πως για την σωστή λειτουργία μιας εταιρίας δεν γίνεται να λείπει μεγάλο μέρος του εργατικού δυναμικού ταυτόχρονα. Έτσι, ανάλογα τις απαιτήσεις και τις δραστηριότητες ορίζεται μέγιστος αριθμός εργαζομένων που μπορούν να απουσιάζουν ταυτόχρονα. Ο περιορισμός αυτός μας βοηθάει στο να διατηρήσουμε τον κανονισμό αυτόν.

- Separation Days

Κάθε άδεια που έχει ζητηθεί από τον εργαζόμενο πρέπει να απέχει έναν συγκεκριμένο αριθμό ημερών από την προηγούμενη και την επόμενη άδεια που του έχει ανατεθεί προκειμένου να θεωρηθεί εφικτή.

- Περιορισμός που εμπεριέχει τις 2 δυαδικές μεταβλητές απόφασης

Ο συγκεκριμένος περιορισμός εξασφαλίζει ότι οι μέρες που τελικά ανατίθενται στους εργαζομένους δεν ξεπερνάνε την διάρκεια της επιθυμίας που δηλώνουν αρχικά.

- Starting Day of the Bid

Οι εργαζόμενοι καλούνται να δηλώσουν στο σύστημα την ακριβή ημερομηνία έναρξης των αδειών τους, καθώς είναι μια σημαντική πληροφορία για τον έλεγχο και όλων των παραπάνω περιορισμών.

- Bid Length

Τέλος, είναι σημαντικό για τον ίδιο λόγο που αναφέρθηκε από πάνω να γνωρίζουμε και τη διάρκεια της άδειας που επιθυμεί ο εργαζόμενος να λάβει.

Η παρούσα διπλωματική εργασία αναλύει τις προτιμήσεις των μελών του στόλου και αναθέτει όσες από αυτές τηρούν τους κανόνες και περιορισμούς. Έπειτα συνεχίζεται και η ανάθεση των ημερών άδειας που υπολείπονται.

## Κεφάλαιο 3<sup>ο</sup> : Θεωρητική Περιγραφή των μεθόδων που χρησιμοποιήθηκαν

### 3.1 Μικτός Ακέραιος Προγραμματισμός

Ο γραμμικός προγραμματισμός μεγιστοποιεί (ή ελαχιστοποιεί) μια γραμμική αντικειμενική συνάρτηση που υπόκειται σε έναν ή περισσότερους περιορισμούς. Προσθέτει μια επιπλέον συνθήκη που τουλάχιστον μία από τις μεταβλητές μπορεί να αναλάβει μόνο ακέραιες τιμές. Στο παρόν πρόβλημα όλες οι μεταβλητές μας είναι ακέραιες.

Η τεχνική βρίσκει ευρεία χρήση στην επιχειρησιακή έρευνα. Η ικανότητα επίλυσης μεγάλων προβλημάτων που περιλαμβάνουν πληθώρα μεταβλητών και παραμέτρων είναι ιδιαίτερα σημαντική για τον σχεδιασμό πολύπλοκων συστημάτων [7]. Οι μέθοδοι μικτού ακέραιου προγραμματισμού, είναι σχεδιασμένες με τέτοιο τρόπο που μπορούν να διαχειριστούν μεγάλο αριθμό μεταβλητών. Δεδομένου ότι το πλήθος των μεταβλητών δυσχεραίνει την υπολογιστική διαδικασία επίλυσης του προβλήματος και οι περισσότερες από αυτές έχουν μηδενική τιμή στη βέλτιστη λύση αυτού, η μέθοδος αυτή στοχεύει στην επίλυση του αρχικού προβλήματος συμπεριλαμβάνοντας όμως μόνο μια υποστοιχία των μεταβλητών αυτών.

Η απλοποιημένη αναπαράσταση ενός προβλήματος μικτού ακέραιου προγραμματισμού είναι:

$$z = C$$

Υπό τους περιορισμούς τύπου:  $AX \leq b$ ,  $X \geq 0$ , μερικά  $x_i$  περιορίζονται σε ακέραιες τιμές.

όπου

$$X = (x_1, x_2, \dots, x_n)'$$

$$C = (c_1, c_2, \dots, c_n)$$

$$b = (b_1, b_2, \dots, b_m)'$$

$$A = \begin{bmatrix} a_{11} & a_{1n} \\ a_{m1} & a_{mn} \end{bmatrix}$$

Τα  $x_i$  είναι οι μεταβλητές απόφασης ενώ η πρώτη εξίσωση ( $z = C$ ) ονομάζεται αντικειμενική συνάρτηση.

Οι ανισότητες (και οι ισότητες) ονομάζονται περιορισμοί. Τα όρια περιορισμού, ονομάζονται συχνά δεξιά πλευρές (RHS).

### 3.2 Γλώσσα Προγραμματισμού «C»

Η C είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου και γενικής χρήσης που είναι ιδανική για την ανάπτυξη λογισμικού ή εφαρμογών. Αρχικά προοριζόταν για τη γραφή λογισμικού συστήματος, η C αναπτύχθηκε στα Bell Labs από τον Dennis Ritchie για το λειτουργικό σύστημα Unix στις αρχές της δεκαετίας του 1970.

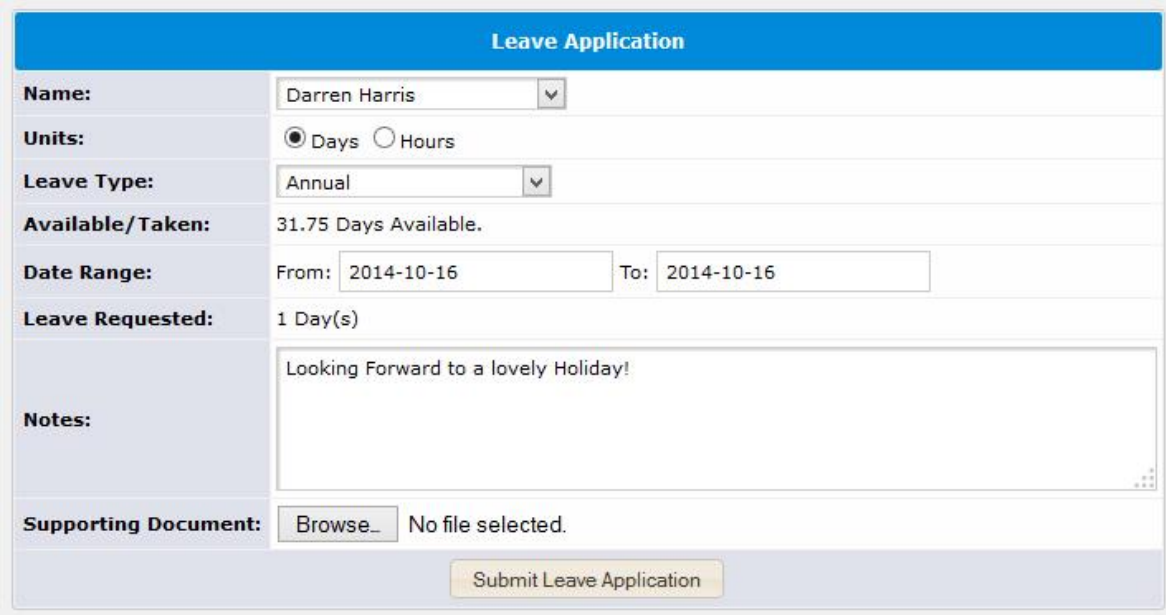
Στη παρούσα διπλωματική επιλέξαμε να χρησιμοποιήσουμε την «C» διότι αποτελεί μία από τις πιο γνωστές και ευρέως χρησιμοποιούμενες γλώσσες. Διαθέτει μεταγλωττιστή για τα περισσότερα συστήματα υπολογιστών, και έχει πολύ καλή υπολογιστική δύναμη. Επίσης, η «C» είναι μια δομημένη γλώσσα προγραμματισμού στην οποία το πρόγραμμα χωρίζεται σε διάφορες ενότητες ή συναρτήσεις όπως τις καλούμε. Κάθε συνάρτηση μπορεί να κωδικοποιηθεί ξεχωριστά και όλες μαζί αποτελούν ένα ενιαίο πρόγραμμα «C». Αυτή η δομή διευκολύνει τη δοκιμή, τη συντήρηση και τον εντοπισμό σφαλμάτων.

Ένα άλλο χαρακτηριστικό της γλώσσας είναι ότι περιέχει διάφορες λειτουργίες που αποτελούν μέρος μιας βιβλιοθήκης. Η CPLEX είναι μια από τις βιβλιοθήκες που θα χρησιμοποιηθεί σε μεγάλο βαθμό για την επίλυση του προβλήματος μας.

## Κεφάλαιο 4<sup>ο</sup> : Ανάθεση του προβλήματος σύμφωνα με τις προτιμήσεις του προσωπικού

### 4.1 Ανάλυση του προβλήματος

Όπως αναφέραμε και στο παραπάνω κεφάλαιο, ένα από τα ζητήματα που καλείται μια αεροπορική να λύσει είναι αυτό της κατανομής αδειών στα μέλη του πληρώματος αεροπορικών εταιριών. Αρχικά γίνεται η συλλογή όλων των αιτημάτων, μέσω μιας φόρμας σχεδιασμένη για αυτό το σκοπό. Οι εργαζόμενοι θα πρέπει να δηλώσουν έναρξη και διάρκεια των αδειών που επιθυμούν να πάρουν.



Leave Application	
Name:	Darren Harris
Units:	<input checked="" type="radio"/> Days <input type="radio"/> Hours
Leave Type:	Annual
Available/Taken:	31.75 Days Available.
Date Range:	From: 2014-10-16 To: 2014-10-16
Leave Requested:	1 Day(s)
Notes:	Looking Forward to a lovely Holiday!
Supporting Document:	Browse... No file selected.
<input type="button" value="Submit Leave Application"/>	

Εικόνα 8 Παράδειγμα φόρμας στην οποία το προσωπικό δηλώνει τις προτιμήσεις του [16]

Έπειτα, αφού οι εργαζόμενοι δηλώσουν τις προτιμήσεις τους, αυτές επεξεργάζονται, ελέγχονται και κρίνονται εφικτές με βάση τους περιορισμούς που αναφέραμε παραπάνω. Οι εφικτές άδειες κατανέμονται στο προσωπικό, συνήθως με προτεραιότητα στα μέλη της εταιρίας με τα περισσότερα χρόνια υπηρεσίας (Seniority Rule). Πολλές αεροπορικές εταιρίες, ακολουθούν τον κανόνα αυστηρής προτεραιότητας (Strict Priority). Η ανάθεση εδώ ξεκινά με τα μέλη του προσωπικού που έχουν τα περισσότερα χρόνια προϋπηρεσίας στην εταιρία. Ελέγχονται δηλαδή, όλες οι δηλωμένες προτιμήσεις του αρχαιότερου μέλους και όσες κριθούν εφικτές του ανατίθενται. Αφού τελειώνει η ανάθεση για το πρώτο μέλος η διαδικασία συνεχίζεται στο επόμενο μέλος με τα αμέσως περισσότερα χρόνια προϋπηρεσίας.

Είναι σημαντικό να αναφέρουμε, ότι πέρα από τον παραπάνω τρόπο ανάθεσης πολλές εταιρίες χρησιμοποιούν και τον κανόνα της δίκαιης προτεραιότητας (Fair Priority). Η

ανάθεση εδώ ξεκινά πάλι με τον εργαζόμενο που έχει τα περισσότερα χρόνια προϋπηρεσίας, αλλά εφόσον του ανατεθεί η πρώτη εφικτή άδεια η διαδικασία θα περάσει στον αμέσως επόμενο με τα περισσότερα χρόνια στην εταιρία. Η διαδικασία συνεχίζεται έως ότου όλες οι πιθανές και εφικτές άδειες έχουν ανατεθεί στους εργαζομένους.

Για την εκπόνηση της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε ο κανόνας της αυστηρής προτεραιότητας και αυστηρής αρχαιότητας του προσωπικού (Strict Seniority - Strict Priority).

## **4.2 Περιγραφή του Λογισμικού και Κώδικα που χρησιμοποιήθηκε**

### **4.2.1 Περιγραφή Λογισμικού**

Το σύστημα που περιγράφουμε κατασκευάστηκε με τη βοήθεια της γλώσσας προγραμματισμού C. Θεωρήθηκε η κατάλληλη γλώσσα καθώς είναι μια γλώσσα γενικής χρήσης που μπορεί να χρησιμοποιηθεί για το σχεδιασμό γραφικών, ηλεκτρονικών παιχνιδιών και φυσικά εφαρμογών και συστημάτων όπως αυτού που περιγράφουμε.

Το λογισμικό που επιλέχθηκε να χρησιμοποιηθεί είναι Visual Studio 2022 C/C++. Η τελευταία έκδοση αποτελεί το πρώτο τους IDE 64-bit, το οποίο διευκολύνει την εργασία με μεγαλύτερα και περισσότερο πολύπλοκα συστήματα.

Ακόμη, για καλύτερη βελτιστοποίηση χρησιμοποιήσαμε τις βιβλιοθήκες CLEX οι οποίες προέρχονται από το λογισμικό IBM ILOG CPLEX 12.5.1

### **4.2.2 Κατασκευή του Κώδικα**

Όπως αναφέραμε παραπάνω ο κώδικάς μας χτίζεται για να κάνει ανάθεση αδειών με την μέθοδο Strict Seniority – Strict Priority. Αρχικά, επιλέγουμε να εργαστούμε με δύο δυαδικές μεταβλητές, μία για τις ημέρες άδειας του κάθε εργαζομένου και μία ακόμη που συμβολίζει τις επιθυμίες του κάθε εργαζομένου, δηλαδή τα <<πακέτα>> ημερών που έχει δηλώσει. Η επιτυχία ανάθεσης θα δηλώνετε εάν πάρουν την τιμή 1, και 0 εάν δεν τους ανατεθεί η άδεια. Για να είναι βέλτιστα γραμμένος, αλλά και εύκολος προς την ανάγνωση ο κώδικας, αποφασίζουμε κάθε μείζον μέρος να διατυπωθεί σε μορφή συνάρτησης.



Χωρίζουμε την κατασκευή του κώδικα σε 5 βασικά κομμάτια:

1. Δημιουργία σωστού περιβάλλοντος κατάλληλο να διαχειριστεί τη γλώσσα C και το πρόβλημα CPLEX.
2. Δημιουργία των περιορισμών εφικτότητας αδειών.
3. Δημιουργία της Αντικειμενικής Συνάρτησης
4. Δίνουμε την δυνατότητα στο χρήστη να κάνει εισαγωγή δεδομένων από έτοιμο αρχείο ή μέσω τυχαίας επιλογής αριθμών.
5. Έλεγχος των περιορισμών και τελική ανάθεση και λύση του προβλήματος.

➤ **Δημιουργία σωστού περιβάλλοντος κατάλληλο να διαχειριστεί τη γλώσσα C και το πρόβλημα CPLEX :**

Κατά την έναρξη του κώδικα, εισάγουμε τα απαραίτητα header files με την εντολή `#include`. Τα αρχεία αυτά περιέχουν έτοιμο κώδικα, ο οποίος βρίσκεται αποθηκευμένος σε αρχεία εκτός του πηγαίου κώδικα που γράφουμε και είναι απαραίτητος για την ομαλή εκτέλεση του. Μερικά από τα header files που χρησιμοποιήσαμε είναι:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <ilcplex/cplex.h>
#include <stdbool.h>
```

Εικόνα 9: Header Files του κώδικα

Έπειτα ακολουθεί η δήλωση των δεδομένων με την εντολή `#define`, η οποία επιτρέπει τη δήλωση σταθερών τιμών για χρήση σε ολόκληρο των κώδικα. Οι μεταβλητές αυτές ονομάζονται *global variables*. Κάποιες από αυτές τις σταθερές μεταβλητές είναι ο αριθμός των εργαζομένων, ο αριθμός των ημερών που εξετάζονται στο πρόβλημα και μπορούν να ανατεθούν και τελικά ο μέγιστος αριθμός ημερών άδειας που μπορούν να τους ανατεθούν.

Στη συνέχεια, καλούμε την υπορουτίνα `CPXopenCPLEX`, η οποία αρχικοποιεί το πρόβλημα μας και την `CPXcreateprob` που θα δημιουργήσει το πρόβλημα μας. Το πρόβλημα που δημιουργείται είναι αρχικά ένα πρόβλημα ελαχιστοποίησης. Παρόλα αυτά το πρόβλημα μας αποτελεί ένα πρόβλημα μεγιστοποίησης. Έτσι, με την χρήση της `CPXchgobjsen` το μετατρέπουμε σε ένα πρόβλημα μεγιστοποίησης με μηδενικούς περιορισμούς, μηδέν μεταβλητές και έναν κενό πίνακα περιορισμών, που φυσικά θα

προστεθούν αργότερα στο κώδικα. Σημαντικό είναι να αναφέρουμε πως στο τέλος του κώδικα και το περιβάλλον CPLEX αλλά και το ίδιο το πρόβλημα πρέπει να ελευθερωθούν. Αυτό επιτυγχάνεται μέσω των υπορουτινών CPXcloseCPLEX και CPXfreeprob αντίστοιχα.

Τέλος θέλοντας να είμαστε ικανοί να δούμε εάν το πρόβλημα μας εκτυπώνεται σωστά μπορούμε ακριβώς μετά την δήλωση των μεταβλητών με την #define να καλέσουμε την ρουτίνα CPXwriteprob. Ανοίγουμε ένα αρχείο στο οποίο θα εκτυπωθεί, η αντικειμενική συνάρτηση, όλοι οι περιορισμοί και μεταβλητές που θα εισαχθούν, καθώς και η λύση.

#### ➤ Δημιουργία των περιορισμών εφικτότητας αδειών:

Στο συγκεκριμένο σημείο του κώδικα μας όπως αναφέραμε παραπάνω το πρόβλημα είναι άδειο. Κατά την έναρξη του κώδικα θα πρέπει να προηγηθεί αρχικοποίηση των πινάκων μεταβλητών απόφασης, ώστε να επιλυθεί το πρόβλημα. Τις αρχικοποιούμε δίνοντας τους την τιμή - 1, εννοώντας ότι δεν έχουν προστεθεί ακόμη στο πρόβλημα, αφού όπως είδαμε παραπάνω οι τιμές 0 και 1 εκφράζουν την επιτυχία ή αποτυχία ανάθεσης της άδειας.

Έπειτα λοιπόν, δημιουργούμε τις δύο πρώτες συναρτήσεις με τις οποίες προσθέτουμε όλες τις μεταβλητές απόφασης. Οι υπορουτίνες CPXnewcols και CPXnewrows χρησιμοποιούνται εδώ γι' αυτό το σκοπό.

Στη συνέχεια, σε ξεχωριστές συναρτήσεις προσθέτουμε τους περιορισμούς μας με σειρά σημαντικότητας :

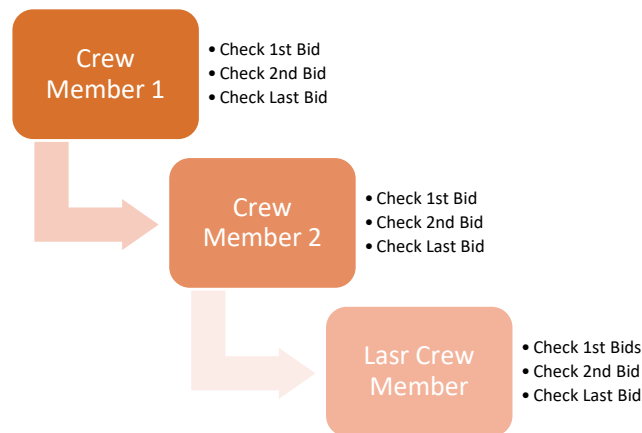
- Maximum Number of Bids
- Entitlement
- Limit Lines
- Overlapping
- Separation
- Connection between decision dual variables

Οι ρουτίνες CPXnewrows , μάς είναι εξαιρετικά χρήσιμες στη προσθήκη των νέων σειρών στους περιορισμούς, καθώς και οι ρουτίνες CPXgetnumrows/CPXgetnumcols είναι σημαντικές διότι, δίνουν πρόσβαση στον αριθμό των γραμμών στον πίνακα περιορισμών.

#### ➤ Δημιουργία της Αντικειμενικής Συνάρτησης:

Αναφέραμε παραπάνω πως το πρόβλημα μας αποτελεί πρόβλημα μεγιστοποίησης και επιτυγχάνουμε αυτή την αλλαγή με τη βοήθεια της συνάρτησής CPLEX. Για την σωστή ανάθεση αδειών με την μέθοδο Strict Priority, ελέγχουμε όλες τις επιθυμίες του πρώτου μέλους του στόλου, το οποίο έχει και τα περισσότερα χρόνια προϋπηρεσίας και όσες περάσουν τον έλεγχο ανατίθενται. Παρ' όλα αυτά θα πρέπει

να θυμόμαστε πως για τον κάθε εργαζόμενο μπορεί να ικανοποιηθεί συγκεκριμένος αριθμός επιθυμιών.



Εικόνα 10: Διάγραμμα ροής για τον έλεγχο Strict Priority

Για τον σκοπό αυτό δημιουργούμε έναν συντελεστή με αρχική τιμή το 1. Για κάθε επιθυμία, διπλασιάζουμε την τιμή του και στη συνέχεια ο συντελεστής πολλαπλασιάζεται με τις μεταβλητές απόφασης  $Y_{ij}$ . Διαβάζουμε τους πίνακες που αφορούν τον στόλο με σειρά αρχαιότητας και τις επιθυμίες ξεκινώντας από το τέλος. Εργαζόμαστε κατά αυτόν τον τρόπο έτσι ώστε ο μικρότερος συντελεστής να ανατίθεται στην επιθυμία με την μικρότερη προτεραιότητα για κάθε μέλος του πληρώματος.

Αναλυτικότερα, σε κάθε επανάληψη ελέγχουμε τις επιθυμίες του αρχαιότερου μέλους του πληρώματος, λύνουμε το πρόβλημα και αναθέτουμε τις επιθυμίες που πραγματοποιήθηκαν, με την ίδια λογική περνά στον αμέσως επόμενο εργαζόμενο. Όσο το πρόβλημα λύνετε για τον κάθε μέλος του προσωπικού ξεχωριστά, στην αντικειμενική συνάρτηση θετικό συντελεστή έχουν οι επιθυμίες που ελέγχονται την δεδομένη στιγμή. Οι άδειες των υπολοίπων θεωρούνται <<ανενεργές>>, μέχρι την στιγμή που θα περάσουν από τους ελέγχους εφικτότητας.

Είναι σημαντικό να αναφέρουμε πως για κάθε εργαζόμενο ο συντελεστής αυτός θα αρχικοποιείται στο 1. Κατά αυτόν τον τρόπο αποφεύγετε η δημιουργία πολύ μεγάλων συντελεστών και το πρόγραμμα μας δεν <<υπερχειλίζει>>.

- **Δίνουμε την δυνατότητα στο χρήστη να κάνει εισαγωγή δεδομένων από έτοιμο αρχείο ή μέσω τυχαίας επιλογής αριθμών:**

Πριν λάβουν μέρος όσα προαναφέρθηκαν, δίνουμε την δυνατότητα στο χρήστη του συστήματος είτε να λάβει τα δεδομένα από ένα ήδη υπάρχον αρχείο ή από γεννήτρια τυχαίων αριθμών. Στην πρώτη περίπτωση, δημιουργούμε μία συνάρτηση στην οποία εισάγουμε και διαβάζουμε τα δεδομένα μας.

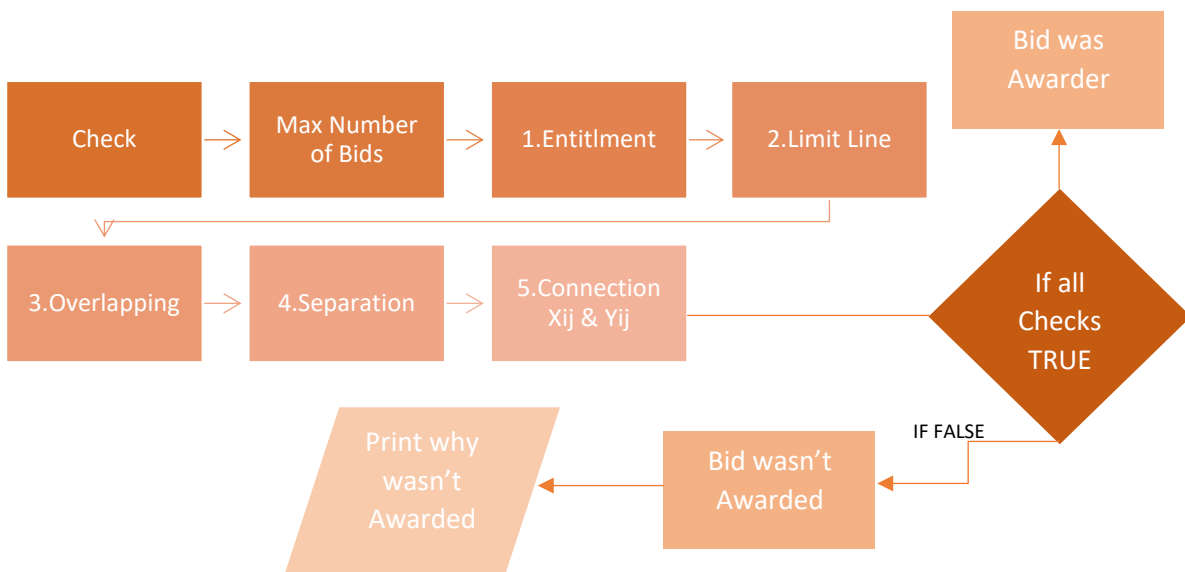
Στη δεύτερη περίπτωση δημιουργούμε `void()` type συναρτήσεις όπου δημιουργούμε τυχαίους αριθμούς με την εντολή `rand()` , για όλους τους περιορισμούς.

Για χάριν ευκολίας, αποφασίζουμε τον αριθμό των ιπταμένων, τις ημέρες που εξετάζουμε καθώς και τον μέγιστο αριθμό ημερών να μείνουν σταθερά ανεξαρτήτως επιλογής του χρήστη.

➤ **Έλεγχος των περιορισμών και τελική ανάθεση και λύση του προβλήματος:**

Έχοντας λοιπόν αρχικοποιήσει το πρόβλημα και τις τιμές και έχοντας προσθέσει όλους τους απαραίτητους περιορισμούς, θα πρέπει τα δεδομένα που παράγονται να ελέγχονται για την εφικτότητα τους. Κατασκευάζουμε λοιπόν τις συναρτήσεις `Check`, μία για κάθε περιορισμό.

Πραγματοποιούμε κατ' επανάληψη για όλα τα μέλη του προσωπικού και όλες τις επιθυμίες τους απαραίτητους ελέγχους και έτσι αποφασίζεται η ανάθεση των ημερών. Η σειρά ελέγχου των περιορισμών έχει μεγάλη σημασία για την σωστή ανάθεση. Στη παρούσα εργασία ο έλεγχος γίνεται με την σειρά με την οποία προστέθηκαν στο πρόβλημα μας. Παρακάτω βλέπου και το αντίστοιχο διάγραμμα ροής.



Εικόνα 11: Διάγραμμα ροής Ελέγχου Αδειών

Πιο συγκεκριμένα, ελέγχονται αρχικά όλες οι επιθυμίες του μέλους του πληρώματος με την μεγαλύτερη προϋπηρεσία και του ανατίθενται όσες έχουν περάσει τους ελέγχους. Αφού ολοκληρωθεί ο πρώτος κύκλος ανάθεσης, περνάει στον αμέσως επόμενο και ξεκινάει τους ελέγχους από την αρχή. Αυτό συμβαίνει μέχρι να εκτιμηθούν και να γίνει ανάθεση όλων των αδειών που μπορεί να γίνει.

Για τα δύο τελευταία βήματα δεν χρησιμοποιήσαμε κάποια από τις ρουτίνες CPLEX.

Τέλος, συμπεριλαμβάνουμε στην `main()` όλες τις συναρτήσεις που πρέπει να <<διαβάσει>> ο κώδικας για να λυθεί το πρόβλημα καθώς και ένα μετρητή χρόνου που θα μας βοηθήσει να δούμε την ταχύτητα του προγράμματος.

## *Κεφάλαιο 5<sup>ο</sup> : Μαθηματική Μοντελοποίηση του Προβλήματος*

### **5.1 Περιγραφή Μεταβλητών και Παραμέτρων**

Προκειμένου να κατανοηθεί το πρόβλημα που μελετάμε ας δούμε την μαθηματική μοντελοποίηση του αναλυτικά παρακάτω.

#### **Μεταβλητές Απόφασης:**

**i:** δείκτης που συμβολίζει το κάθε μέλος του στόλου

**j:** δείκτης που συμβολίζει την κάθε επιθυμία του αντίστοιχου μέλους του στόλου

**$Y_{ij} = 1$**  αν ικανοποιηθεί η επιθυμία  $j$  του εργαζομένου  $i$  και  **$0$**  αν όχι

**$X_{ij} = 1$**  αν ο εργαζόμενος  $i$  πάρει άδεια την ημέρα  $j$  και  **$0$**  αν όχι

#### **Σύνολα:**

**N :** Αριθμός μελών του στόλου στους οποίους ανατίθενται οι άδειες

**T:** Αριθμός ημερών που μπορούν να ανατεθούν σαν μέρες άδειας

#### **Παράμετροι:**

**a:** Συντελεστής προτεραιότητας

**b:** Μέγιστος αριθμός αδειών που μπορεί να πάρει τελικά ο κάθε εργαζόμενος

**c:** Διάρκεια της κάθε άδειας του κάθε εργαζομένου

**d:** Μέγιστος αριθμός ημερών που μπορούν να ανατεθούν

## 5.2 Θεωρητική Ανάλυση της Αντικειμενικής Εξίσωσης και των Περιορισμών

Χρησιμοποιώντας την παραπάνω σημειογραφία μορφοποιείται το παρακάτω πρόβλημα βελτιστοποίησης:

$$\text{Max } \sum_{i,j} a * Y_{ij} \quad (5.1)$$

$$\text{s. t. } \sum_{i,j} Y_{ij} \leq b \quad (5.2)$$

$$\sum_{i,j} c * Y_{ij} \leq d_i \quad (5.3)$$

$$\sum_{i,j} X_{i,j} + \dots + X_{i+n,j} \leq 2 \quad (5.4)$$

$$\sum_{i,j} Y_{i,j} \leq 1 \quad (5.5)$$

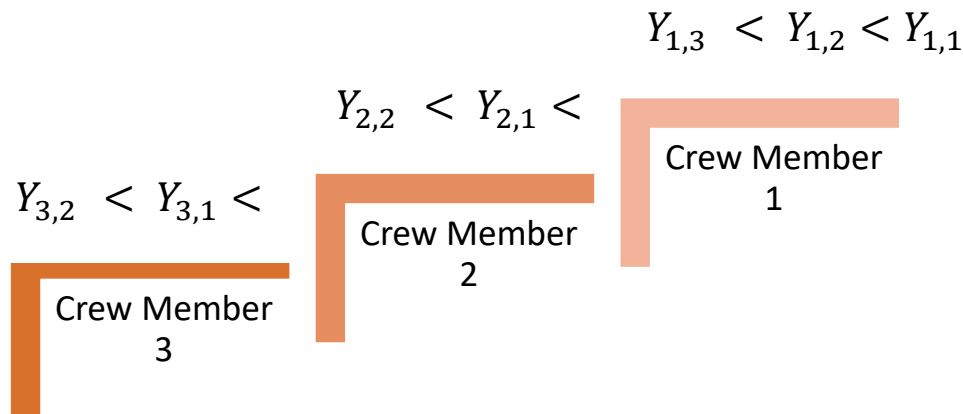
$$\sum_{i,j} c * Y_{ij} \leq \sum_{i,j} X_{ij} \quad (5.6)$$

$$X_{ij}, Y_{ij} \text{ binary, } \in Z$$

Η εξίσωση (5.1) αποτελεί την αντικειμενική συνάρτηση του προβλήματος. Είναι ένα πρόβλημα μεγιστοποίησης και στόχος μας είναι η μέγιστη ικανοποίηση των επιθυμιών των εργαζομένων. Τα μέλη του στόλου και οι επιθυμίες τους καταχωρούνται στο σύστημα με βάση την μέθοδο που μελετάμε Strict Seniority – Strict Priority. Πιο συγκεκριμένα, ο πρώτος εργαζόμενος είναι και ο πιο παλιός, ενώ η πρώτη επιθυμία του καταχωρημένη στο σύστημα έχει την μεγαλύτερη προτεραιότητα.

Κατά την λύση του προβλήματος, δημιουργούμαι έναν συντελεστή **a** όπως αναφέρθηκε παραπάνω, ο οποίος σε κάθε επανάληψη πολλαπλασιάζεται με κάθε επιθυμία  $Y_{ij}$ . Αυτός ο αριθμός μας βοηθάει να εξασφαλίσουμε ότι η επιθυμία του αρχαιότερου μέλους του προσωπικού, με την μεγαλύτερη προτεραιότητα θα <<κερδίσει>> σε περίπτωση που 2 επιθυμίες του ίδιου εργαζομένου πάρουν την τιμή 1 κατά την διαδικασία ελέγχου και ανάθεσης, δεδομένου ότι μπορούν να πάρουν συγκεκριμένο αριθμό αδειών.

Για παράδειγμα, εάν έχουμε 3 μέλη του προσωπικού με καταχωρημένες επιθυμίες 3,2,2 αντίστοιχα η προτεραιότητα δίνεται ως εξής:



Εικόνα 12: Παράδειγμα περιγραφής ιεραρχίας προσωπικού και επιθυμιών

Η εξίσωση (5.2) αφορά τον περιορισμό των μέγιστων επιθυμιών που μπορεί να ανατεθεί στο κάθε μέλος του στόλου. Το σύστημα ορίζει συγκεκριμένο αριθμό επιθυμιών που μπορεί να ικανοποιηθεί για όλους, δηλαδή ανεξαρτήτως τον αριθμό των επιθυμιών που θα δηλωθούν από κάθε εργαζόμενο στο τέλος για όλους θα ικανοποιηθεί  $b$  αριθμός επιθυμιών.

Το σύνολο των εξισώσεων (5.3) αφορά την πλήρη ανάθεση όλων των ημερών που δικαιούνται οι εργαζόμενοι. Πολλαπλασιάζουμε την κάθε επιθυμία με την διάρκεια τής, αυτό εκφράζεται από την παράμετρο  $c$ , ενώ η παράμετρος  $d$  δηλώνει τις μέγιστες ημέρες άδειας του καθενός.

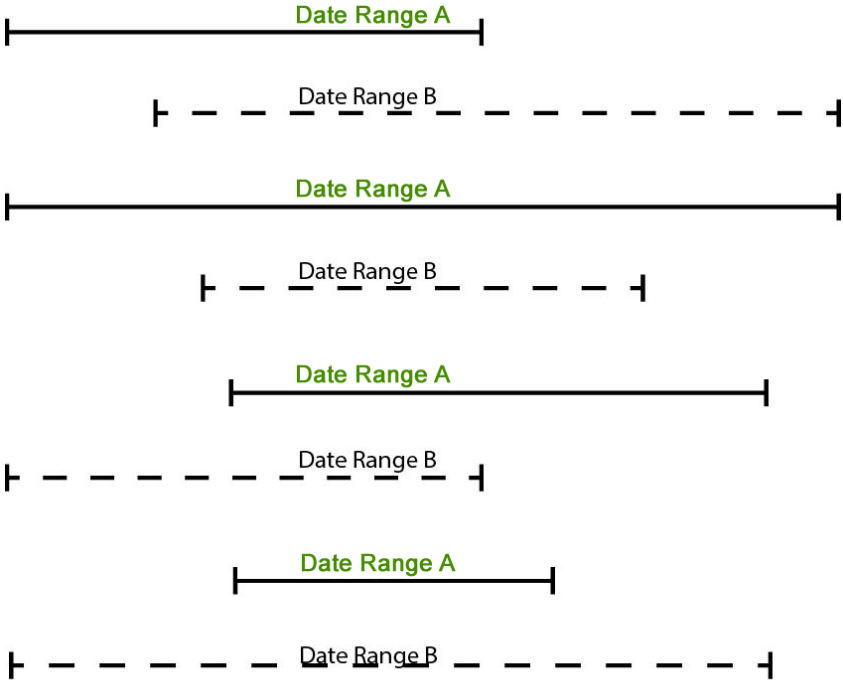
Πολύ σημαντικές αποτελούν το σύνολο των εξισώσεων (5.4), που αφορούν τα limit lines. Όπως αναφέραμε για κάθε ημέρα του χρόνου υπάρχει ένας συγκεκριμένος αριθμός εργαζομένων που μπορούν να λείπουν σε άδεια, χωρίς να επηρεάζεται η πραγματοποίηση των δραστηριοτήτων της αεροπορικής εταιρίας. Κατά την δημιουργία των εξισώσεων αυτών υπάρχει περίπτωση κάποιες να είναι περιττές. Ας υποθέσουμε πάλι 3 εργαζομένους, οι οποίοι έχουν δηλώσει κάποιες κοινές μέρες άδειας με limit line 2, οι περιορισμοί θα παρουσιάζονταν ως εξής:

```
X13 <= 2;  
X35 <= 2;  
X26 + X36 <= 2;  
X17 + X27 <= 2;  
X110 + X310 <= 2;  
X111 + X311 <= 2;  
X112 + X212 + X312 <= 2;  
X113 + X213 + X313 <= 2;
```

Εικόνα 13: Παράδειγμα περιορισμών limit line

Αφού οι τιμές μας θα παίρνουν πάντα τις τιμές 0 ή 1, οι δύο πρώτες εξισώσεις θεωρούνται περιττές σε αυτή την περίπτωση, και καθώς χτίζουμε τον κώδικα μας πρέπει να το λάβουμε υπόψιν και να μην τις συμπεριλάβουμε. Με αυτόν, τον τρόπο ο κώδικας μας βελτιστοποιείται αλλά και ο χρόνος ελέγχου αδειών μειώνεται, ειδικά για πολύ μεγάλα προβλήματα.

Η εξίσωση (5.5) παριστάνει 2 περιορισμούς, separation και overlap. Οι περιορισμοί αυτοί αφορούν τον ίδιο εργαζόμενο. Ο περιορισμός separation αναφέρει πως οι άδειες που θα ανατεθούν στον ίδιο εργαζόμενο, πρέπει να απέχουν έναν συγκεκριμένο αριθμό που καθορίζεται κάθε φορά από την εταιρία. Οι overlap περιορισμοί ελέγχουν ότι για τις ίδιες ημερομηνίες έναρξης και λήξης της άδειας δεν έχει ανατεθεί κάποια άλλη άδεια στον εργαζόμενο.



Εικόνα 14: Περιπτώσεις που οι μέρες θεωρούνται overlapped



Τέλος, οι εξισώσεις (5.6) καθιερώνουν την σχέση μεταξύ των μεταβλητών απόφασης  $Y_{ij}$  και  $X_{ij}$ . Η διάρκεια κάθε επιθυμίας πολλαπλασιασμένη με την μεταβλητή κάθε επιθυμίας πρέπει να είναι μικρότερη ή ίση με το άθροισμα των ημερών που αντιστοιχούν σε κάθε άδεια.

$$c * Y_{i,j} \leq X_{i,j} + X_{i,j+1} \dots + X_{i,j+n} \quad (5.8)$$

## Κεφάλαιο 6<sup>ο</sup> : Παρουσίαση προβλήματος μικρού σκέλους

Για την καλύτερη κατανόηση του προβλήματος και της διαδικασίας ανάθεσης, σε αυτό το κεφάλαιο θα παρουσιάσουμε την αναλυτική λύση ενός ολοκληρωμένου προβλήματος. Στη πρώτη περίπτωση θα διαβάσουμε ένα έτοιμο αρχείο και στο δεύτερο σκέλος θα καλέσουμε την γεννήτρια τυχαίων αριθμών.

### 6.1 Παρουσίαση προβλήματος μικρού σκέλους με εισαγωγή αρχείου

Η επιλογή των δεδομένων εισόδου στο παράδειγμα που τίθεται στο παρόν κεφάλαιο έγινε με σκοπό η μεθοδολογία επίλυσης του προβλήματος να γίνει όσο το δυνατόν πιο σαφής στον αναγνώστη.

Για αυτό το λόγο, επιλέγουμε να αναλύσουμε την απλούστερη εκδοχή του προβλήματος με δεδομένα εισόδου τα εξής:

**N = 3** αριθμός μελών του στόλου

**T = 31** ημέρες άδειας (θεωρούμε τον μήνα Ιανουάριο για χάριν κατανόησης του προβλήματος)

**MaxBids = 20** μέγιστος αριθμός ημερών άδειας για κάθε εργαζόμενο

**Seperdays = 3** αριθμός ημερών που πρέπει να απέχουν οι άδειες

**MaxNumberOfBids = 2** μέγιστος αριθμός επιθυμιών που μπορεί να ικανοποιηθεί για κάθε εργαζόμενο

**Limit Line = 2** αριθμός εργαζομένων που μπορούν να λείπουν σε άδεια

Ποιο αναλυτικά τα δεδομένα του προβλήματος παρουσιάζονται στον παρακάτω πίνακα :

<b>Δείκτης ιπταμένου</b>	<b>Ημέρες άδειας</b>	<b># επιθυμιών</b>
<b>1</b>	20	4
<b>2</b>	15	3
<b>3</b>	22	5

*Εικόνα 15:Πίνακας Δεδομένων*

Οι επιθυμίες των 3 μελών του πληρώματος παρουσιάζονται κατά σειρά προτεραιότητας και οι εργαζόμενοι παρατίθενται κατά σειρά αρχαιότητας:

#### **Crew Member 1**

**1η επιθυμία :** 7 ημέρες ξεκινώντας στις **7** Ιανουαρίου

**2η επιθυμία :** 15 ημέρες ξεκινώντας στις **13** Ιανουαρίου

**3η επιθυμία:** 10 ημέρες ξεκινώντας στις **10** Ιανουαρίου

**4<sup>η</sup> επιθυμία:** 2 ημέρες ξεκινώντας στις **2** Ιανουαρίου

#### **Crew Member 2**

**1η επιθυμία :** 3 ημέρες ξεκινώντας στις **6** Ιανουαρίου

**2η επιθυμία :** 9 ημέρες ξεκινώντας στις **12** Ιανουαρίου

**3η επιθυμία:** 5 ημέρες ξεκινώντας στις **18** Ιανουαρίου

#### **Crew Member 3**

**1η επιθυμία :** 7 ημέρες ξεκινώντας στις **17** Ιανουαρίου

**2η επιθυμία :** 5 ημέρες ξεκινώντας στις **23** Ιανουαρίου

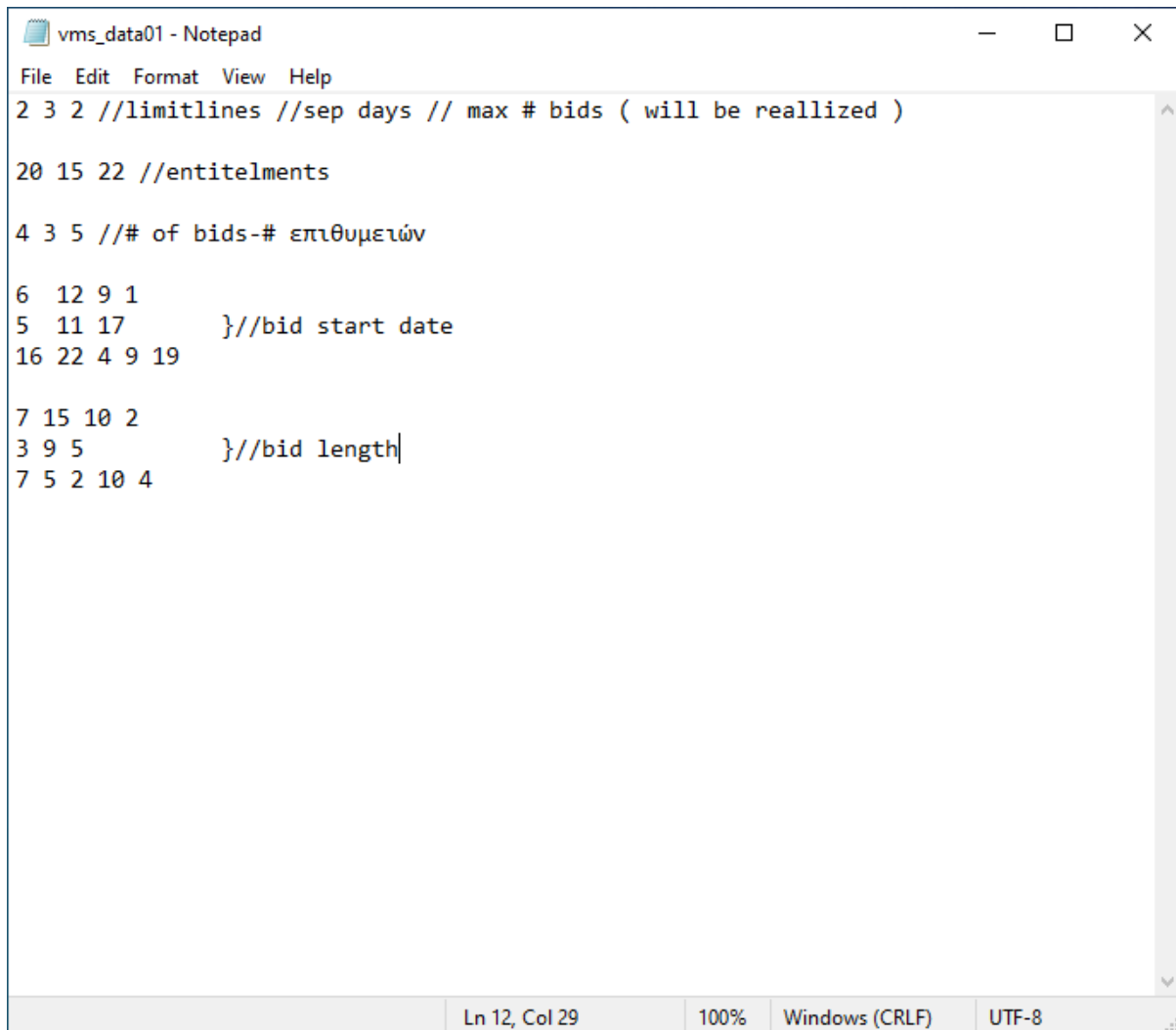
**3η επιθυμία:** 2 ημέρες ξεκινώντας στις **5** Ιανουαρίου

**4<sup>η</sup> επιθυμία:** 10 ημέρες ξεκινώντας στις **10** Ιανουαρίου

**5η επιθυμία :** 4 ημέρες ξεκινώντας στις **20** Ιανουαρίου

Για να μπορεί ο κώδικας μας να διαβάσει τα παραπάνω δεδομένα, δημιουργούμε ένα .txt αρχείο. Πρέπει το αρχείο μας να είναι κατάλληλο γραμμένο έτσι ώστε ο κώδικας να μπορεί να το διαβάσει.

Στη συγκεκριμένη περίπτωση το αρχείο τροφοδοτείται με την εξής μορφή:



```
vms_data01 - Notepad
File Edit Format View Help
2 3 2 //limitlines //sep days // max # bids ( will be realized )

20 15 22 //entitelments

4 3 5 //# of bids-# επιθυμιών

6 12 9 1
5 11 17 }//bid start date
16 22 4 9 19

7 15 10 2
3 9 5 }//bid length|
7 5 2 10 4
```

Εικόνα 16: Αρχείο δεδομένων που εισάγεται από τον χρήστη

Πριν το σύστημα διαβάσει το αρχείο, έχει γίνει κατάλληλη δήλωση των σταθερών δεδομένων. Να αναφέρουμε πως για την επίλυση του προβλήματος χρειαζόμαστε την δήλωση και όλων των απαραίτητων πινάκων.

```
#define N 3 //αριθμός μελών του στόλου
#define T 31 //αριθμός ημερών που θα ανατεθούν για άδεια
#define MaxBids 20 //μέγιστος αριθμός ημερών άδειας για τον καθένα

int separdays = 3;
int MaxNumberOfBids = 2; //μέγιστος αριθμός επιθυμιών που μπορεί να ικανοποιηθεί για τον καθένα
```

Εικόνα 17: Δηλώσεις μεταβλητών στον κώδικα

Όπως αναφέραμε και παραπάνω, αφού αρχικοποιήσουμε το πρόβλημα και το περιβάλλον CPLEX, οι πρώτες συναρτήσεις που δημιουργούνται έχουν σκοπό να εισάγουν τις μεταβλητές στο πρόβλημα. Όλες είναι κάτω και άνω φραγμένες μεταξύ του 0 και 1.

```

thesis - Notepad
File Edit Format View Help
\ENCODING=ISO-8859-1
\Problem name: myprob

Maximize
obj1: 0 Y1_1 + 0 Y1_2 + 0 Y1_3 + 0 Y1_4 + 0 Y2_1 + 0 Y2_2 + 0 Y2_3 + 0 Y3_1
      + 0 Y3_2 + 0 Y3_3 + 0 Y3_4 + 0 Y3_5 + 0 X1_7 + 0 X1_8 + 0 X1_9 + 0 X1_10
      + 0 X1_11 + 0 X1_12 + 0 X1_13 + 0 X1_14 + 0 X1_15 + 0 X1_16 + 0 X1_17
      + 0 X1_18 + 0 X1_19 + 0 X1_20 + 0 X1_21 + 0 X1_22 + 0 X1_23 + 0 X1_24
      + 0 X1_25 + 0 X1_26 + 0 X1_27 + 0 X1_2 + 0 X1_3 + 0 X2_6 + 0 X2_7
      + 0 X2_8 + 0 X2_12 + 0 X2_13 + 0 X2_14 + 0 X2_15 + 0 X2_16 + 0 X2_17
      + 0 X2_18 + 0 X2_19 + 0 X2_20 + 0 X2_21 + 0 X2_22 + 0 X3_17 + 0 X3_18
      + 0 X3_19 + 0 X3_20 + 0 X3_21 + 0 X3_22 + 0 X3_23 + 0 X3_24 + 0 X3_25
      + 0 X3_26 + 0 X3_27 + 0 X3_5 + 0 X3_6 + 0 X3_10 + 0 X3_11 + 0 X3_12
      + 0 X3_13 + 0 X3_14 + 0 X3_15 + 0 X3_16

Bounds
0 <= Y1_1 <= 1
0 <= Y1_2 <= 1
0 <= Y1_3 <= 1
0 <= Y1_4 <= 1
0 <= Y2_1 <= 1
0 <= Y2_2 <= 1
0 <= Y2_3 <= 1
0 <= Y3_1 <= 1
0 <= Y3_2 <= 1
0 <= Y3_3 <= 1
0 <= Y3_4 <= 1
0 <= Y3_5 <= 1
0 <= X1_7 <= 1
0 <= X1_8 <= 1
0 <= X1_9 <= 1
0 <= X1_10 <= 1
0 <= X1_11 <= 1
0 <= X1_12 <= 1
0 <= X1_13 <= 1
0 <= X1_14 <= 1
0 <= X1_15 <= 1
0 <= X1_16 <= 1
0 <= X1_17 <= 1
0 <= X1_18 <= 1
0 <= X1_19 <= 1
0 <= X1_20 <= 1
0 <= X1_21 <= 1
0 <= X1_22 <= 1
0 <= X1_23 <= 1
0 <= X1_24 <= 1
0 <= X1_25 <= 1

```

```

0 <= X1_26 <= 1
0 <= X1_27 <= 1
0 <= X1_2 <= 1
0 <= X1_3 <= 1
0 <= X2_6 <= 1
0 <= X2_7 <= 1
0 <= X2_8 <= 1
0 <= X2_12 <= 1
0 <= X2_13 <= 1
0 <= X2_14 <= 1
0 <= X2_15 <= 1
0 <= X2_16 <= 1
0 <= X2_17 <= 1
0 <= X2_18 <= 1
0 <= X2_19 <= 1
0 <= X2_20 <= 1
0 <= X2_21 <= 1
0 <= X2_22 <= 1
0 <= X3_17 <= 1
0 <= X3_18 <= 1
0 <= X3_19 <= 1
0 <= X3_20 <= 1
0 <= X3_21 <= 1
0 <= X3_22 <= 1
0 <= X3_23 <= 1
0 <= X3_24 <= 1
0 <= X3_25 <= 1
0 <= X3_26 <= 1
0 <= X3_27 <= 1
0 <= X3_5 <= 1
0 <= X3_6 <= 1
0 <= X3_10 <= 1
0 <= X3_11 <= 1
0 <= X3_12 <= 1
0 <= X3_13 <= 1
0 <= X3_14 <= 1
0 <= X3_15 <= 1
0 <= X3_16 <= 1
Binaries
Y1_1 Y1_2 Y1_3 Y1_4 Y2_1 Y2_2 Y2_3 Y3_1 Y3_2 Y3_3 Y3_4 Y3_5 X1_7
X1_8 X1_9 X1_10 X1_11 X1_12 X1_13 X1_14 X1_15 X1_16 X1_17 X1_18
X1_19 X1_20 X1_21 X1_22 X1_23 X1_24 X1_25 X1_26 X1_27 X1_2 X1_3
X2_6 X2_7 X2_8 X2_12 X2_13 X2_14 X2_15 X2_16 X2_17 X2_18 X2_19
X2_20 X2_21 X2_22 X3_17 X3_18 X3_19 X3_20 X3_21 X3_22 X3_23 X3_24
X3_25 X3_26 X3_27 X3_5 X3_6 X3_10 X3_11 X3_12 X3_13 X3_14 X3_15
X3_16
End

```

Εικόνα 18:Αρχείο που έχουν εκτυπωθεί οι περιορισμοί

Παρατηρούμαι ότι στην αντικειμενική συνάρτηση έχουν συμπεριληφθεί και οι μεταβλητές  $X_{ij}$ , καθώς και πως ο συντελεστής είναι 0. Αυτό συμβαίνει γιατί σε αυτό το βήμα κάνουμε απλή εισαγωγή και δημιουργία μεταβλητών. Αργότερα στο πρόβλημα δημιουργούμε μια συνάρτηση στην οποία διορθώνουμε την αντικειμενική εξίσωση και την διαμορφώνουμε σύμφωνα με όσα αναλύσαμε στην αρχή του Κεφαλαίου 5.

Στη συνέχεια εισάγουμε τους περιορισμούς με την σειρά προτεραιότητας που αναφέρθηκε στο προηγούμενο κεφάλαιο. Παρακάτω βλέπουμε πώς στο αρχείο δημιουργείται το **Subject to** κομμάτι, κάτω από το οποίο θα βρίσκονται όλοι οι περιορισμοί.

Για το συγκεκριμένο πρόβλημα οι περιορισμοί διαμορφώνονται ως εξής.

### Maximum Number Of Bids

Στο παρόν πρόβλημα, μόνο 2 επιθυμίες από κάθε μέλος του πληρώματος μπορούν να πραγματοποιηθούν. Έτσι, το άθροισμα των επιθυμιών για το κάθε εργαζόμενο διαμορφώνεται ως εξής.

Subject To
c1: $Y1\_1 + Y1\_2 + Y1\_3 + Y1\_4 \leq 2$
c2: $Y2\_1 + Y2\_2 + Y2\_3 \leq 2$
c3: $Y3\_1 + Y3\_2 + Y3\_3 + Y3\_4 + Y3\_5 \leq 2$

Εικόνα 19: Περιορισμοί Max Number of Bids

### Entitlement

c4: $7 Y1\_1 + 15 Y1\_2 + 10 Y1\_3 + 2 Y1\_4 \leq 20$
c5: $3 Y2\_1 + 9 Y2\_2 + 5 Y2\_3 \leq 15$
c6: $7 Y3\_1 + 5 Y3\_2 + 2 Y3\_3 + 10 Y3\_4 + 4 Y3\_5 \leq 22$

Εικόνα 20: Περιορισμοί Entitlement

### Overlapping

c7: $Y1\_1 + Y1\_2 \leq 1$
c8: $Y1\_1 + Y1\_3 \leq 1$
c9: $Y1\_2 + Y1\_3 \leq 1$
c10: $Y2\_2 + Y2\_3 \leq 1$
c11: $Y3\_1 + Y3\_2 \leq 1$
c12: $Y3\_1 + Y3\_4 \leq 1$
c13: $Y3\_1 + Y3\_5 \leq 1$
c14: $Y3\_2 + Y3\_5 \leq 1$

Εικόνα 21: Περιορισμοί Overlapping

### Separation

Στο παρόν παράδειγμα υπάρχουν μόνο δύο επιθυμίες του εργαζομένου 3 που παραβιάζουν τον κανόνα του separation.

c15: $Y3\_4 + Y3\_5 \leq 1$
-----------------------------

Εικόνα 22: Περιορισμός Separation

Έχουμε θεωρήσει πως ο κανόνας overlap υπερέχει του separation. Έτσι, εάν ο overlap παραβιάζεται και ο separation όχι ο κώδικας θα θεωρήσει αυτό τον συνδυασμό αδειών μη εφικτό. Επομένως, στον κώδικα μας πρέπει να υπάρχει ο παρακάτω έλεγχος:

```
For κάθε εργαζόμενο i
  For κάθε επιθυμία j
    For κάθε επιθυμία j+1
      If Overlap Rule παραβιάζεται συνέχισε στην επόμενη δυάδα επιθυμιών
      If Separation Rule παραβιάζεται
        (take necessary actions)
```

Εικόνα 23:Ψευδοκώδικας λογικής Separation

## Limit Line

Όπως διατυπώθηκε στην περιγραφή της εξίσωσης (5.4) στο Κεφάλαιο 5, κατά την εισαγωγή των περιορισμών limit line υπάρχουν και περιττές εξισώσεις. Η εξαγωγή τους είναι σημαντική διότι κατά τη διάρκεια των ελέγχων εφικτότητας δε θα χρειαστεί να τις ελέγξει. Μειώνουμε τον υπολογιστικό χρόνο του συστήματος.

```
c16: <= 1
c17: X1_2 <= 1
c18: X1_3 <= 1
c19: <= 1
c20: X3_5 <= 1
c21: X2_6 + X3_6 <= 1
c22: X1_7 + X2_7 <= 1
c23: X1_8 + X2_8 <= 1
c24: X1_9 <= 1
c25: X1_10 + X3_10 <= 1
c26: X1_11 + X3_11 <= 1
c27: X1_12 + X2_12 + X3_12 <= 2
c28: X1_13 + X2_13 + X3_13 <= 2
c29: X1_14 + X2_14 + X3_14 <= 2
c30: X1_15 + X2_15 + X3_15 <= 2
c31: X1_16 + X2_16 + X3_16 <= 2
c32: X1_17 + X2_17 + X3_17 <= 2
c33: X1_18 + X2_18 + X3_18 <= 2
c34: X1_19 + X2_19 + X3_19 <= 2
c35: X1_20 + X2_20 + X3_20 <= 2
c36: X1_21 + X2_21 + X3_21 <= 2
c37: X1_22 + X2_22 + X3_22 <= 2
c38: X1_23 + X3_23 <= 1
c39: X1_24 + X3_24 <= 1
c40: X1_25 + X3_25 <= 1
c41: X1_26 + X3_26 <= 1
c42: X1_27 + X3_27 <= 1
c43: <= 1
c44: <= 1
c45: <= 1
c46: <= 1
```

Εικόνα 24: Περιορισμοί Limit Line

## Connection between decision variables

```
c47: 7 Y1_1 - X1_7 - X1_8 - X1_9 - X1_10 - X1_11 - X1_12 - X1_13 <= 0
c48: 15 Y1_2 - X1_13 - X1_14 - X1_15 - X1_16 - X1_17 - X1_18 - X1_19 - X1_20 | X1_21 - X1_22 - X1_23 - X1_24 - X1_25 - X1_26 - X1_27 <= 0
c49: 10 Y1_3 - X1_10 - X1_11 - X1_12 - X1_13 - X1_14 - X1_15 - X1_16 - X1_17 - X1_18 - X1_19 <= 0
c50: 2 Y1_4 - X1_2 - X1_3 <= 0
c51: 3 Y2_1 - X2_6 - X2_7 - X2_8 <= 0
c52: 9 Y2_2 - X2_12 - X2_13 - X2_14 - X2_15 - X2_16 - X2_17 - X2_18 - X2_19 - X2_20 <= 0
c53: 5 Y2_3 - X2_18 - X2_19 - X2_20 - X2_21 - X2_22 <= 0
c54: 7 Y3_1 - X3_17 - X3_18 - X3_19 - X3_20 - X3_21 - X3_22 - X3_23 <= 0
c55: 5 Y3_2 - X3_23 - X3_24 - X3_25 - X3_26 - X3_27 <= 0
c56: 2 Y3_3 - X3_5 - X3_6 <= 0
c57: 10 Y3_4 - X3_17 - X3_18 - X3_19 - X3_10 - X3_11 - X3_12 - X3_13 - X3_14 - X3_15 - X3_16 <= 0
c58: 4 Y3_5 - X3_20 - X3_21 - X3_22 - X3_23 <= 0
```

Εικόνα 25: Περιορισμοί που συνδυάζουν  $X_{ij}$ ,  $Y_{ij}$

## 6.2 Λύση του προβλήματος μικρού σκέλους με εισαγωγή αρχείου

Σε αυτό το στάδιο του προβλήματος «διορθώνουμε» την αντικειμενική συνάρτηση καθώς και κάνουμε όλους τους ελέγχους και εξάγουμε την τελική λύση του προβλήματος.

Τελικά οι άδειες που ανατέθηκαν στον κάθε εργαζόμενο είναι οι εξής:

Crew Member 1: Άδειες 1 και 4

Crew Member 2: Άδειες 1 και 2

Crew Member 3: Άδειες 1 και 3

Βλέπουμε, πως η λύση φαίνεται στο αρχείο. Οι άδειες που εν τέλη ανατέθηκαν έχουν πάρει την τιμή 1.

```
Y1_1 = 1
0 <= Y1_2 <= 1
0 <= Y1_3 <= 1
Y1_4 = 1
Y2_1 = 1
Y2_2 = 1
0 <= Y2_3 <= 1
Y3_1 = 1
0 <= Y3_2 <= 1
Y3_3 = 1
0 <= Y3_4 <= 1
0 <= Y3_5 <= 1
```

Εικόνα 26: Λύση του προβλήματος



Ανάμεσα στις άδειες που ανατέθηκαν μπορούμε να δούμε και ποιοι περιορισμοί παραβιάστηκαν, το οποίο οδήγησε σε απόρριψη της επιθυμίας αυτής.

```
Crew Member 1 was Assigned bid 1 - No rules were Violated
Crew member 1 wasn't Assigned bid 2 - Violating Overlap
Crew member 1 wasn't Assigned bid 3 - Violating Overlap
Crew Member 1 was Assigned bid 4 - No rules were Violated
Crew Member 2 was Assigned bid 1 - No rules were Violated
Crew Member 2 was Assigned bid 2 - No rules were Violated
Crew member 2 wasn't Assigned bid 3 - Violating Overlap
Crew Member 3 was Assigned bid 1 - No rules were Violated
Crew member 3 wasn't Assigned bid 2 - Violating Overlap
Crew Member 3 was Assigned bid 3 - No rules were Violated
Crew member 3 wasn't Assigned bid 4 - Violating LimitLines
Crew member 3 wasn't Assigned bid 5 - Violating LimitLines
```

### 6.3 Παρουσίαση προβλήματος μικρού σκέλους με τυχαία γεννήτρια

Κατά την έναρξη του κώδικα ο χρήστης παρουσιάζεται με δύο επιλογές, είτε να εισάγει ένα αρχείο με τα απαραίτητα δεδομένα, είτε να καλέσει τυχαία γεννήτρια η οποία θα δημιουργήσει ένα τυχαίο πρόβλημα.

Είναι σημαντικό η γεννήτρια που θα δημιουργήσουμε να αποφεύγει την ανάθεση μηδενικών τιμών στα δεδομένα μας. Ακόμη, θα πρέπει να προσέξουμε το άθροισμα ανάμεσα της ημέρας έναρξης και των ημερών διάρκειας της άδειας, να μην υπερβαίνει τον διαθέσιμο αριθμό ημερών προς ανάθεση.

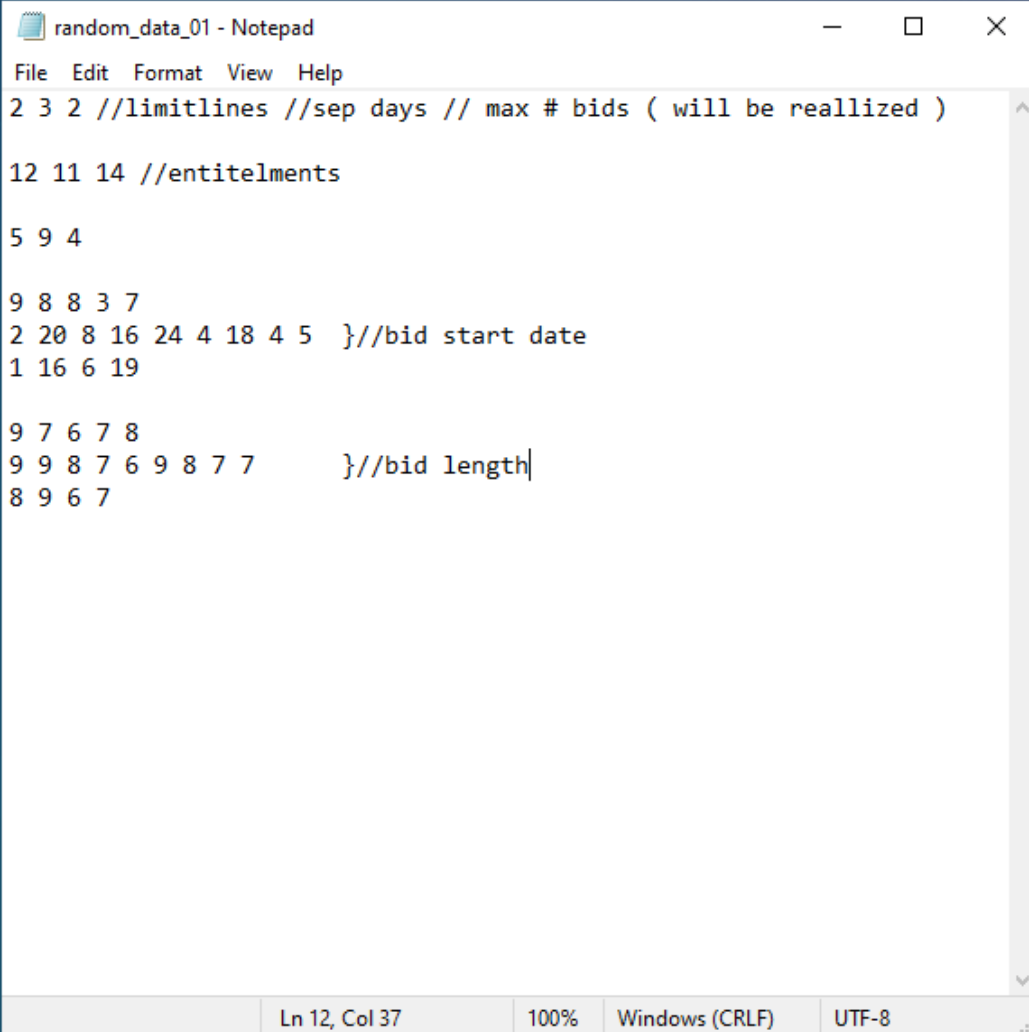
Για το παρόν παράδειγμα:  $\text{BidStart} + \text{BidLength} < T$

Για ευκολία παρουσίασης του τυχαίου προβλήματος επιλέγουμε οι παγκόσμιες μεταβλητές να παραμείνουν ίδιες:

```
#define N 3 //αριθμός προσωπικού
#define T 31 //αριθμός ημερών μου θα ανατεθούν για άδεια
#define MaxBids 20 //μέγιστος αριθμός ημερών άδειας
```

Σε αυτή την περίπτωση, πέρα από την δημιουργία όλων των απαραίτητων μεταβλητών που θα χρειαστούν, πρέπει να <<ανοίξουμε>> ένα αρχείο στο οποίο θα γράφουμε τα δεδομένα με την δομή που δείξαμε παραπάνω.

Το αρχείο αυτό στη συνέχεια θα διαβάζεται από τον κώδικα και θα προσχωρήσει στην επίλυση του.



```
random_data_01 - Notepad
File Edit Format View Help
2 3 2 //limitlines //sep days // max # bids ( will be realized )
12 11 14 //entitelments
5 9 4
9 8 8 3 7
2 20 8 16 24 4 18 4 5 }//bid start date
1 16 6 19
9 7 6 7 8
9 9 8 7 6 9 8 7 7 }//bid length|
8 9 6 7
Ln 12, Col 37 100% Windows (CRLF) UTF-8
```

Εικόνα 27: Τυχαία δεδομένα από την γεννήτρια

Αφού το πρόγραμμά διαβάσει το αρχείο, η επίλυση του προβλήματος παραμένει ίδια. Εισάγονται όλοι οι περιορισμοί με την σειρά βαρύτητας και ελέγχονται. Τέλος, γίνεται η επίλυση και η ανάθεση των αδειών.

Η λύση αυτού του προβλήματος παρουσιάζεται παρακάτω.

```
Crew Member 1 was Assigned bid 1 - No rules were Violated
Crew member 1 wasn't Assigned bid 2 - Violating Overlap
Crew member 1 wasn't Assigned bid 3 - Violating Overlap
Crew member 1 wasn't Assigned bid 4 - Violating Separation
Crew member 1 wasn't Assigned bid 5 - Violating Overlap
Crew Member 2 was Assigned bid 1 - No rules were Violated
Crew Member 2 was Assigned bid 2 - No rules were Violated
Crew member 2 wasn't Assigned bid 3 - Violating LimitLines
Crew member 2 wasn't Assigned bid 4 - Violating Overlap
Crew member 2 wasn't Assigned bid 5 - Violating Overlap
Crew member 2 wasn't Assigned bid 6 - Violating LimitLines
Crew member 2 wasn't Assigned bid 7 - Violating Overlap
Crew member 2 wasn't Assigned bid 8 - Violating LimitLines
Crew member 2 wasn't Assigned bid 9 - Violating LimitLines
Crew Member 3 was Assigned bid 1 - No rules were Violated
Crew Member 3 was Assigned bid 2 - No rules were Violated
Crew member 3 wasn't Assigned bid 3 - Violating LimitLines
Crew member 3 wasn't Assigned bid 4 - Violating LimitLines
```

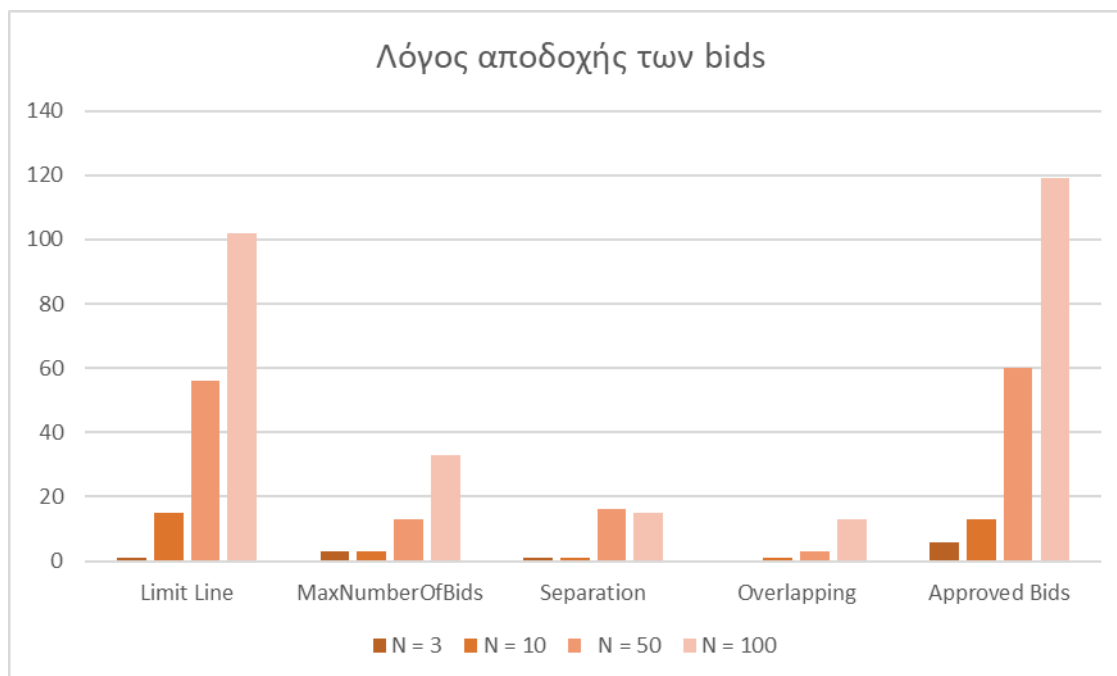
## 6.4 Δοκιμές και πειράματα

Παρακάτω παρουσιάζονται κάποιες δοκιμές που γίναν, αλλάζοντας κάθε φορά τις παραμέτρους και τα δεδομένα του προβλήματος. Η συλλογή των αποτελεσμάτων και η ανάλυσή τους οδηγεί στην εξαγωγή πολύτιμων συμπερασμάτων.

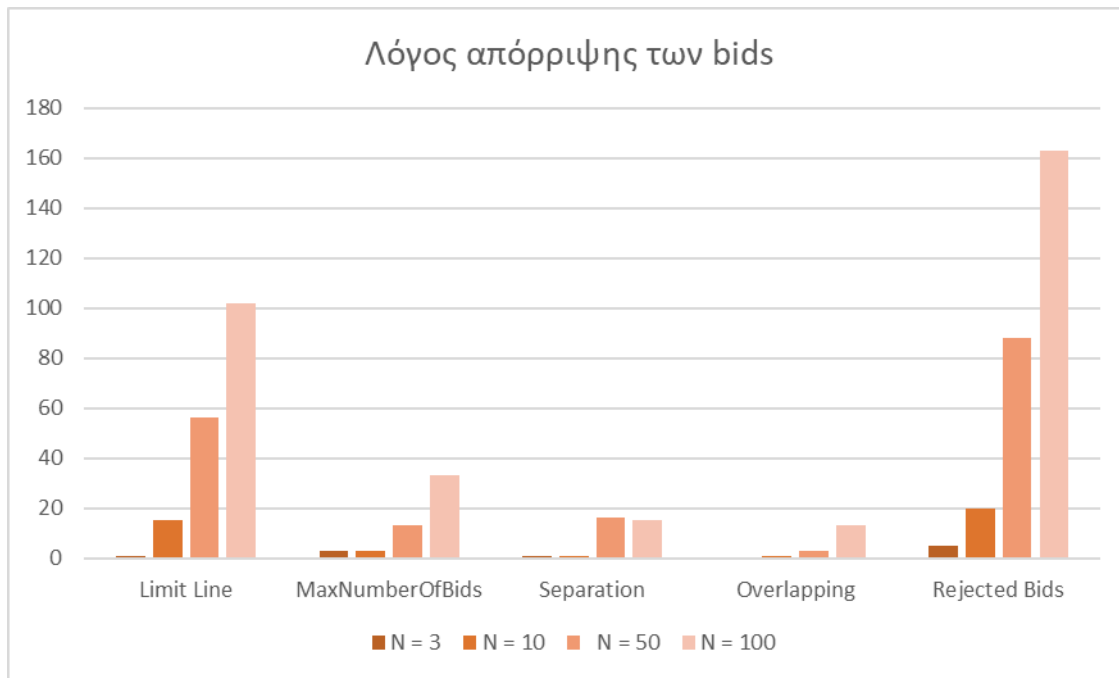
Έλαβαν χώρα πειράματα με αριθμό εργαζομένων 3, 10, 50, 100 με τυχαίο αριθμό προτιμήσεων από 1 μέχρι και 5 για τον καθένα τους. Υποτέθηκε πως ο αριθμός των ημερών που έχει το κάθε μέλος του πληρώματος στη διάθεσή είναι 20, η εταιρεία έχει θεσπίσει limit line με τυχαίους αριθμούς. Επίσης έγινε η υπόθεση ότι το separation είναι 10 και ότι η διάρκεια της άδειας για κάθε προτίμηση είναι τυχαίος αριθμός μεταξύ 6 και 10 ημερών. Τέλος, το πείραμα γίνεται για κύκλο 90 ημερών.

Εφαρμόζοντας την μέθοδο Strict Seniority – Strict Priority για διαφορετικό αριθμό εργαζομένων αλλά κρατώντας σταθερό τον αριθμό ημερών που έχει ο καθένας στη διάθεση του και ορίζοντας 2 τον μέγιστο αριθμό που μπορεί να ικανοποιηθεί, λαμβάνουμε πολύ σημαντικά αποτελέσματα τα οποία παρουσιάζουμε παρακάτω.

Για το πείραμα  $N = 3$  ο αριθμός Limit Line πήρε την τιμή 1, για  $N = 10$  την τιμή 2, για  $N = 50$  την τιμή 8 ενώ για  $N = 100$  την τιμή 15.



Εικόνα 28: Λόγοι αποδοχής των Bids



Εικόνα 29: Λόγοι απόρριψης των Bids

Εξετάζοντας τα αποτελέσματα του πειράματος οδηγούμαστε σε μια σειρά από συμπεράσματα. Βέβαια όσο μεγαλύτερο είναι το δείγμα, στην περίπτωση μας ο αριθμός των μελών του στόλου, τόσο πιο ασφαλή συμπεράσματα μπορούμε να βγάλουμε.

Γίνεται προφανές πως όσο αυξάνουμε τον αριθμό του  $N$  σε ένα πείραμα και τον αριθμό των προτιμήσεων τους, τόσο μειώνεται ο αριθμός των ικανοποιημένων επιθυμιών. Το μοντέλο αρχικά ικανοποιεί όσες επιθυμίες μπορεί για κάθε εργαζόμενο. Η ικανοποίηση μέχρι να φτάσει στον τελευταίο μέλος πληρώματος έχει μειωθεί σημαντικά.

Παρατηρείται επίσης ότι στη μικρή τιμή των limit lines, περισσότερα bids μένουν ανικανοποίητα. Αυτό συμβαίνει γιατί λόγω ιεραρχίας θα ικανοποιούνται σχεδόν πλήρως οι επιθυμίες του πρώτου, έτσι οι επόμενοι δε θα μπορούν να πάρουν κάποια άδεια στη συγκεκριμένη μέρα.

# *Κεφάλαιο 7<sup>ο</sup>: Συμπεράσματα και Μελλοντικοί Στόχοι*

## **7.1 Συμπεράσματα**

Στην παρούσα διπλωματική εργασία παρουσιάστηκε ένα μοντέλο μικτού ακέрайου προγραμματισμού για το πρόβλημα ανάθεσης ημερών άδειας σε μέλη πληρώματος με τη μέθοδο Strict Seniority – Strict Priority. Στην προτεινόμενη μεθοδολογία εισάγουμε αρκετούς περιορισμούς που βοηθάνε στην βέλτιστή ανάθεση αδειών, τηρώντας την αρχαιότητα κάθε μέλους προσωπικού καθώς και την προτεραιότητα κάθε επιθυμίας.

Διεξάγοντας αρκετά πειράματα, μεγάλου ή μικρού σκέλους εξήχθησαν ποικίλα συμπεράσματα, μετά το τέλος της διπλωματικής εργασίας.

Αρχικά, με όσα μελετήθηκαν είναι πλέον αδιαμφισβήτητο ότι η κατασκευή κώδικα για τον έλεγχο των αιτημάτων του πληρώματος, βάσει συγκεκριμένων κανόνων που ορίζονται από την πολιτική των αεροπορικών εταιριών, και η ανάθεση όσων από αυτά κρίνονται ορθά, είναι εφικτή. Αποδεικνύεται επίσης, πως τέτοια μοντέλα, όπως αυτό που κατασκευάστηκε στην παρούσα εργασία μπορεί να εφαρμοστεί για ποικίλες εταιρίες σε αρκετές βιομηχανίες. Η λογική που μελετήσαμε καθώς και ο ίδιος ο κώδικας μπορεί εύκολα να προσαρμοστεί στις ανάγκες και τις επιθυμίες κάθε εταιρίας.

Τέλος, μέσα από παραδείγματα και δοκιμές που έλαβαν χώρα μετά την ολοκλήρωση του κώδικα, διαπιστώθηκε πως η μεθοδολογία που επιλέχθηκε για την επίλυση του προβλήματος είναι ιδιαίτερα αποτελεσματική καθώς κατορθώνει στις περισσότερες περιπτώσεις να κάνει πλήρη ανάθεση των επιθυμιών και να εξάγει πλήρως επιθυμητά αποτελέσματα.

## 7.2 Μελλοντικοί Στόχοι

Στην παρούσα εργασία εξετάσαμε εκτενώς και με επιτυχία το ζήτημα ανάθεσης επιθυμιών με βάση τα χρόνια προϋπηρεσίας κάθε μέλους προσωπικού, και την σειρά προτεραιότητας των επιθυμιών τους. Επιτρέπουμε στο κάθε εργαζόμενο να δηλώσει ελεύθερα τις προτιμήσεις του, την ημέρα εκκίνησης, διάρκειας και φυσικά την σημαντικότητα της άδειας σε σχέση με τις υπόλοιπες. Κατασκευάσαμε ένα μοντέλο το οποίο είναι αρκετά γρήγορο και αποτελεσματικό και η ανάθεση αδειών γίνεται χωρίς να κοστίζει ιδιαίτερο υπολογιστικό χρόνο στον χρήστη.

Μελλοντικά, πρέπει να εστιάσουμε στην καλύτερη και βέλτιστη γραφή του κώδικα, ώστε να μην επιβαρύνεται το σύστημα ανεξαρτήτως αριθμό περιορισμών ή όγκο δεδομένων που μελετάμε.

Όπως αναφέρθηκε σε παραπάνω κεφάλαιο, πέρα από την λογική που μελετήσαμε, ευρέως διαδεδομένη είναι και η Fair Priority. Η ανάθεση ξεκινά με τον εργαζόμενο που έχει τα περισσότερα χρόνια προϋπηρεσίας, αλλά εφόσον του ανατεθεί η πρώτη εφικτή άδεια η διαδικασία περνά στον αμέσως επόμενο και συνεχίζεται έως ότου όλες οι πιθανές και εφικτές άδειες έχουν ανατεθεί.

Με τις προτεινόμενες βελτιώσεις θα ήμαστε ικανοί να καλύψουμε μεγαλύτερο μέρος της ζήτησης αγοράς για τέτοια είδους συστήματα, ικανοποιώντας μεγάλο μέρος της αγοράς. Ταυτόχρονα το σύστημά μας θα γίνει προσαρμόσιμο σε οποιοσδήποτε αλλαγές ηγεσίας αναφορικά με τον ορισμό ημερών άδειας, αφού έχουμε ενσωματώσει και τις δύο λογικές.

# Βιβλιογραφία

1. Anne Mercier, François Soumis. (2005, 11 15). An integrated aircraft routing, crew scheduling and flight retiming model. p. 15. Retrieved from <https://doi.org/10.1016/j.cor.2005.09.001>
2. Chellappan Sriram, Ali Haghani. (2001, 12 13). An optimization model for aircraft maintenance scheduling and re-assignment. Ανάκτηση από [https://doi.org/10.1016/S0965-8564\(02\)00004-6](https://doi.org/10.1016/S0965-8564(02)00004-6)
3. Dieter Schmitt, Volker Gollnick. (2015, 01 01). Historical Development of Air Transport. Ανάκτηση από [https://doi.org/10.1007/978-3-7091-1880-1\\_2](https://doi.org/10.1007/978-3-7091-1880-1_2)
4. ETSCHMAIER , ROTHSTEIN . (1973, 07). Operations Research in the Management of the Airlines . Ανάκτηση από <https://pdf.sciencedirectassets.com/271690/1-s2.0-S0305048300X01478/1-s2.0-0305048374900875/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEDQaCXVzLWVhc3QtMSJGMEQCIgI8c87SMQ%2FqM09p0zhHx0BewcjSOGdXkJRYXHu90k5GAiBFXNYSgPxm%2FUHzZ4%2F%2F8tF1czY%2BI6W%2FHoN95>
5. Hanif D.Sherali, Ebru K.Bish, Xiaomei Zhu. (2006, 07 1). Airline fleet assignment concepts, models, and algorithms. Ανάκτηση από <https://doi.org/10.1016/j.ejor.2005.01.056>
6. Hesham K. Alfares. (1999, 06 1). Aircraft maintenance workforce scheduling. Ανάκτηση από <https://doi.org/10.1108/13552519910271784>
7. Hüseyin Gürkan, Sinan Gürel, , M. Selim Aktürk. (2016, 07). An integrated approach for airline scheduling, aircraft fleet and routing with cruise speed control. Ανάκτηση από <https://doi.org/10.1016/j.trc.2016.03.002>
8. Hüseyin Gürkan, Sinan Gürel, M. Selim Aktürk. (2016, 05 1). Transportation Research Part C: Emerging Technologies. Ανάκτηση από <https://www.sciencedirect.com/journal/transportation-research-part-c-emerging-technologies>
9. İbrahim Zeki Akyurt, Yusuf Kuvvetli, Muhammet Devci, Harish Garg & Mert Yuzsever . (2021, 05 21). A new mathematical model for determining optimal workforce planning of pilots in an airline company. Ανάκτηση από <https://link.springer.com/article/10.1007/s40747-021-00386-x>
10. Jhunievieve Soriano, Eugene Rex Jalao, Iris Ann Martinez. (2020, 06). Integrated employee scheduling with known employee demand, including breaks, overtime, and employee preferences. Ανάκτηση από <http://www.jiem.org/index.php/jiem/article/view/3126>
11. Sentot Imam Wahjono, Anna Marina, Ismail Rasulong, Fam Soo Fen. (2020, 08 31). Leave management information system using InsideDPS software for efficiency of human resources management. Ανάκτηση από <https://kinetik.umm.ac.id/index.php/kinetik/article/view/1087>
12. Thomas A. Feo, Jonathan F. Bard. (1989, 12). Flight Scheduling and Maintenance Base Planning. Ανάκτηση από <http://www.jstor.org/stable/2632228>



13. Αλεξάνδρα Μαυριδοπούλου. (2019). Βέλτιστος Χρονοπρογραμματισμός Αδειών Ιπτάμενου Προσωπικού Μέσω της Μεθοδολογίας Δημιουργίας Στηλών. Ανάκτηση από <https://ir.lib.uth.gr/xmlui/bitstream/handle/11615/51045/19867.pdf?sequence=1>
14. Αρκουμάνης Ιωάννης. (2004). Βελτιστοποίηση του προβλήματος ανάθεσης και δρομολόγησης αεροσκαφών αεροπορικής εταιρίας, με την χρήση ευρετικού αλγορίθμου GRASP. Ανάκτηση από <https://doi.org/10.26233/heallink.tuc.10314>
15. Π. Ρίζος, Π. Τσιατάς. (2021). ΒΕΛΤΙΣΤΗ ΑΝΑΘΕΣΗ ΑΔΕΙΩΝ ΙΠΤΑΜΕΝΟΥ ΠΡΟΣΩΠΙΚΟΥ ΜΕ ΤΕΧΝΙΚΕΣ ΑΚΕΡΑΙΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.
16. Leave Application System [http://www.intoweb.com/hr/module\\_leave.php](http://www.intoweb.com/hr/module_leave.php)

# ΠΑΡΑΡΤΗΜΑ

## ΥΠΟΛΟΓΙΣΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ

Στο παρόν Παράρτημα επιλύουμε ένα πρόβλημα το, όπου τα δεδομένα προέρχονται από την τυχαία γεννήτρια αριθμών. Τα δεδομένα εισόδου είναι τα εξής:

**N = 8** αριθμός μελών πληρώματος

**T = 85** αριθμός ημερών που θα ανατεθούν για άδεια

**Separation Days = 10**

**Limit Line = 2**

**Max Number of Satisfied Bids = 2**

```
2 10 2 //limitlines //sep days // max # bids
```

```
11 12 14 14 15 15 11 13 //entitlements
```

```
4 2 4 5 3 2 4 5 //# of bids-# επιθυμιών
```

```
44 65 51 16
```

```
16 51
```

```
23 44 30 72
```

```
51 9 23 58 37 } //bid start date
```

```
9 58 44
```

```
79 9
```

```
16 44 30 16
```

```
65 65 51 9 16
```

```
7 10 7 7
```

```
7 7
```

```
6 10 8 6
```

```
7 10 7 8 10 } //bid length
```

```
9 7 7
```

```
6 6
```

```
7 8 6 6
```

```
8 10 10 10 6
```

Στη συνέχεια παρουσιάζονται οι περιορισμοί γι' αυτό το πρόβλημα.

Subject To

$$\text{Bids\#0: } Y1\_1 + Y1\_2 + Y1\_3 + Y1\_4 \leq 2$$

$$\text{Bids\#1: } Y2\_1 + Y2\_2 \leq 2$$

$$\text{Bids\#2: } Y3\_1 + Y3\_2 + Y3\_3 + Y3\_4 \leq 2$$

$$\text{Bids\#3: } Y4\_1 + Y4\_2 + Y4\_3 + Y4\_4 + Y4\_5 \leq 2$$

$$\text{Bids\#4: } Y5\_1 + Y5\_2 + Y5\_3 \leq 2$$

$$\text{Bids\#5: } Y6\_1 + Y6\_2 \leq 2$$

$$\text{Bids\#6: } Y7\_1 + Y7\_2 + Y7\_3 + Y7\_4 \leq 2$$

$$\text{Bids\#7: } Y8\_1 + Y8\_2 + Y8\_3 + Y8\_4 + Y8\_5 \leq 2$$

$$\text{Entit\#8: } 7 Y1\_1 + 10 Y1\_2 + 7 Y1\_3 + 7 Y1\_4 \leq 11$$

$$\text{Entit\#9: } 7 Y2\_1 + 7 Y2\_2 \leq 12$$

$$\text{Entit\#10: } 6 Y3\_1 + 10 Y3\_2 + 8 Y3\_3 + 6 Y3\_4 \leq 14$$

$$\text{Entit\#11: } 7 Y4\_1 + 10 Y4\_2 + 7 Y4\_3 + 8 Y4\_4 + 10 Y4\_5 \leq 14$$

$$\text{Entit\#12: } 9 Y5\_1 + 7 Y5\_2 + 7 Y5\_3 \leq 15$$

$$\text{Entit\#13: } 6 Y6\_1 + 6 Y6\_2 \leq 15$$

$$\text{Entit\#14: } 7 Y7\_1 + 8 Y7\_2 + 6 Y7\_3 + 6 Y7\_4 \leq 11$$

$$\text{Entit\#15: } 8 Y8\_1 + 10 Y8\_2 + 10 Y8\_3 + 10 Y8\_4 + 6 Y8\_5 \leq 13$$

Separ#16:  $Y7\_1 + Y7\_4 \leq 1$

Separ#17:  $Y8\_1 + Y8\_2 \leq 1$

Separ#18:  $Y8\_4 + Y8\_5 \leq 1$

Separ#19:  $Y1\_3 \leq 1$

Separ#20:  $Y3\_2 + Y3\_4 \leq 1$

Separ#21:  $Y3\_3 \leq 1$

Separ#22:  $Y3\_3 + Y3\_4 \leq 1$

Separ#23:  $Y4\_4 \leq 1$

Separ#24:  $Y4\_3 + Y4\_4 \leq 1$

Separ#25:  $Y4\_3 + Y4\_5 \leq 1$

Separ#26:  $Y4\_4 \leq 1$

Separ#27:  $Y7\_3 \leq 1$

LL#28:  $X4\_10 + X5\_10 + X6\_10 + X8\_10 \leq 1$

LL#29:  $X4\_11 + X5\_11 + X6\_11 + X8\_11 \leq 1$

LL#30:  $X4\_12 + X5\_12 + X6\_12 + X8\_12 \leq 1$

LL#31:  $X4\_13 + X5\_13 + X6\_13 + X8\_13 \leq 1$

LL#32:  $X4\_14 + X5\_14 + X6\_14 + X8\_14 \leq 1$

LL#33:  $X4\_15 + X5\_15 + X6\_15 + X8\_15 \leq 1$

LL#34:  $X4\_16 + X5\_16 + X8\_16 \leq 1$

LL#35:  $X1\_17 + X2\_17 + X4\_17 + X5\_17 + X7\_17 + X8\_17 \leq 1$

LL#36:  $X1\_18 + X2\_18 + X4\_18 + X5\_18 + X7\_18 + X8\_18 \leq 1$

LL#37:  $X1\_19 + X2\_19 + X4\_19 + X7\_19 + X8\_19 \leq 1$

LL#38:  $X1\_20 + X2\_20 + X7\_20 + X8\_20 \leq 1$

LL#39:  $X1\_21 + X2\_21 + X7\_21 + X8\_21 \leq 1$

LL#40:  $X1\_22 + X2\_22 + X7\_22 + X8\_22 \leq 1$

LL#41:  $X1\_23 + X2\_23 + X7\_23 \leq 1$

LL#42:  $X1_{45} + X3_{45} + X4_{45} + X5_{45} + X7_{45} \leq 1$   
LL#43:  $X1_{46} + X3_{46} + X4_{46} + X5_{46} + X7_{46} \leq 1$   
LL#44:  $X1_{47} + X3_{47} + X4_{47} + X5_{47} + X7_{47} \leq 1$   
LL#45:  $X1_{48} + X3_{48} + X5_{48} + X7_{48} \leq 1$   
LL#46:  $X1_{49} + X3_{49} + X5_{49} + X7_{49} \leq 1$   
LL#47:  $X1_{50} + X3_{50} + X5_{50} + X7_{50} \leq 1$   
LL#48:  $X1_{51} + X3_{51} + X5_{51} + X7_{51} \leq 1$   
LL#49:  $X1_{52} + X2_{52} + X3_{52} + X4_{52} + X7_{52} + X8_{52} \leq 1$   
LL#50:  $X1_{53} + X2_{53} + X3_{53} + X4_{53} + X8_{53} \leq 1$   
LL#51:  $X1_{54} + X2_{54} + X3_{54} + X4_{54} + X8_{54} \leq 1$   
LL#52:  $X1_{55} + X2_{55} + X4_{55} + X8_{55} \leq 1$   
LL#53:  $X1_{56} + X2_{56} + X4_{56} + X8_{56} \leq 1$   
LL#54:  $X1_{57} + X2_{57} + X4_{57} + X8_{57} \leq 1$   
LL#55:  $X1_{58} + X2_{58} + X4_{58} + X8_{58} \leq 1$   
LL#56:  $X4_{59} + X5_{59} + X8_{59} \leq 1$   
LL#57:  $X4_{60} + X5_{60} + X8_{60} \leq 1$   
LL58:  $X4_{61} + X5_{61} + X8_{61} \leq 1$   
LL59:  $X1_{66} + X4_{66} + X8_{66} \leq 1$   
LL60:  $X1_{73} + X3_{73} + X8_{73} \leq 1$   
LL61:  $X1_{74} + X3_{74} + X8_{74} \leq 1$   
LL62:  $X1_{75} + X3_{75} + X8_{75} \leq 1$

c20#63: 7 Y1\_1 - X1\_45 - X1\_46 - X1\_47 - X1\_48 - X1\_49 - X1\_50 - X1\_51 <= 0  
c65#64: 10 Y1\_2 - X1\_66 - X1\_67 - X1\_68 - X1\_69 - X1\_70 - X1\_71 - X1\_72 - X1\_73 - X1\_74 - X1\_75 <= 0  
c66#65: 7 Y1\_3 - X1\_52 - X1\_53 - X1\_54 - X1\_55 - X1\_56 - X1\_57 - X1\_58 <= 0  
c67#66: 7 Y1\_4 - X1\_17 - X1\_18 - X1\_19 - X1\_20 - X1\_21 - X1\_22 - X1\_23 <= 0  
c68#67: 7 Y2\_1 - X2\_17 - X2\_18 - X2\_19 - X2\_20 - X2\_21 - X2\_22 - X2\_23 <= 0  
c69#68: 7 Y2\_2 - X2\_52 - X2\_53 - X2\_54 - X2\_55 - X2\_56 - X2\_57 - X2\_58 <= 0  
c70#69: 6 Y3\_1 - X3\_24 - X3\_25 - X3\_26 - X3\_27 - X3\_28 - X3\_29 <= 0  
c71#70: 10 Y3\_2 - X3\_45 - X3\_46 - X3\_47 - X3\_48 - X3\_49 - X3\_50 - X3\_51 - X3\_52 - X3\_53 - X3\_54 <= 0  
c72#71: 8 Y3\_3 - X3\_31 - X3\_32 - X3\_33 - X3\_34 - X3\_35 - X3\_36 - X3\_37 - X3\_38 <= 0  
c73#72: 6 Y3\_4 - X3\_73 - X3\_74 - X3\_75 - X3\_76 - X3\_77 - X3\_78 <= 0  
c74#73: 7 Y4\_1 - X4\_52 - X4\_53 - X4\_54 - X4\_55 - X4\_56 - X4\_57 - X4\_58 <= 0  
c75#74: 10 Y4\_2 - X4\_10 - X4\_11 - X4\_12 - X4\_13 - X4\_14 - X4\_15 - X4\_16 - X4\_17 - X4\_18 - X4\_19 <= 0  
c76#75: 7 Y4\_3 - X4\_24 - X4\_25 - X4\_26 - X4\_27 - X4\_28 - X4\_29 - X4\_30 <= 0  
c77#76: 8 Y4\_4 - X4\_59 - X4\_60 - X4\_61 - X4\_62 - X4\_63 - X4\_64 - X4\_65 - X4\_66 <= 0  
c78#77: 10 Y4\_5 - X4\_38 - X4\_39 - X4\_40 - X4\_41 - X4\_42 - X4\_43 - X4\_44 - X4\_45 - X4\_46 - X4\_47 <= 0  
c79#78: 9 Y5\_1 - X5\_10 - X5\_11 - X5\_12 - X5\_13 - X5\_14 - X5\_15 - X5\_16 - X5\_17 - X5\_18 <= 0  
c80#79: 7 Y5\_2 - X5\_59 - X5\_60 - X5\_61 - X5\_62 - X5\_63 - X5\_64 - X5\_65 <= 0  
c81#80: 7 Y5\_3 - X5\_45 - X5\_46 - X5\_47 - X5\_48 - X5\_49 - X5\_50 - X5\_51 <= 0  
c82#81: 6 Y6\_1 - X6\_80 - X6\_81 - X6\_82 - X6\_83 - X6\_84 - X6\_85 <= 0  
c83#82: 6 Y6\_2 - X6\_10 - X6\_11 - X6\_12 - X6\_13 - X6\_14 - X6\_15 <= 0  
c84#83: 7 Y7\_1 - X7\_17 - X7\_18 - X7\_19 - X7\_20 - X7\_21 - X7\_22 - X7\_23 <= 0  
c85#84: 8 Y7\_2 - X7\_45 - X7\_46 - X7\_47 - X7\_48 - X7\_49 - X7\_50 - X7\_51 - X7\_52 <= 0  
c86#85: 6 Y7\_3 - X7\_31 - X7\_32 - X7\_33 - X7\_34 - X7\_35 - X7\_36 <= 0  
c87#86: 6 Y7\_4 - X7\_17 - X7\_18 - X7\_19 - X7\_20 - X7\_21 - X7\_22 <= 0  
c88#87: 8 Y8\_1 - X8\_66 - X8\_67 - X8\_68 - X8\_69 - X8\_70 - X8\_71 - X8\_72 - X8\_73 <= 0  
c89#88: 10 Y8\_2 - X8\_66 - X8\_67 - X8\_68 - X8\_69 - X8\_70 - X8\_71 - X8\_72 - X8\_73 - X8\_74 - X8\_75 <= 0  
c90#89: 10 Y8\_3 - X8\_52 - X8\_53 - X8\_54 - X8\_55 - X8\_56 - X8\_57 - X8\_58 - X8\_59 - X8\_60 - X8\_61 <= 0  
c91#90: 10 Y8\_4 - X8\_10 - X8\_11 - X8\_12 - X8\_13 - X8\_14 - X8\_15 - X8\_16 - X8\_17 - X8\_18 - X8\_19 <= 0  
c92#91: 6 Y8\_5 - X8\_17 - X8\_18 - X8\_19 - X8\_20 - X8\_21 - X8\_22 <= 0

Τελικά παίρνουμε την παρακάτω λύση:

Crew Member 1 was Assigned bid 1 - No rules were Violated  
Crew Member 1 was Assigned bid 2 - No rules were Violated  
Crew member 1 wasn't Assigned bid 3 - Violating Max Number Of Bids  
Crew Member 2 was Assigned bid 1 - No rules were Violated  
Crew Member 2 was Assigned bid 2 - No rules were Violated  
Crew Member 3 was Assigned bid 1 - No rules were Violated  
Crew Member 3 was Assigned bid 2 - No rules were Violated  
Crew member 3 wasn't Assigned bid 3 - Violating Max Number Of Bids  
Crew member 4 wasn't Assigned bid 1 - Violating LimitLines  
Crew Member 4 was Assigned bid 2 - No rules were Violated  
Crew member 4 wasn't Assigned bid 3 - Violating Separation  
Crew Member 4 was Assigned bid 4 - No rules were Violated  
Crew member 4 wasn't Assigned bid 5 - Violating Max Number Of Bids  
Crew member 5 wasn't Assigned bid 1 - Violating LimitLines  
Crew Member 5 was Assigned bid 2 - No rules were Violated  
Crew member 5 wasn't Assigned bid 3 - Violating LimitLines  
Crew Member 6 was Assigned bid 1 - No rules were Violated  
Crew Member 6 was Assigned bid 2 - No rules were Violated  
Crew member 7 wasn't Assigned bid 1 - Violating LimitLines  
Crew member 7 wasn't Assigned bid 2 - Violating LimitLines  
Crew Member 7 was Assigned bid 3 - No rules were Violated  
Crew member 7 wasn't Assigned bid 4 - Violating LimitLines  
Crew member 8 wasn't Assigned bid 1 - Violating LimitLines  
Crew member 8 wasn't Assigned bid 2 - Violating LimitLines  
Crew member 8 wasn't Assigned bid 3 - Violating LimitLines  
Crew member 8 wasn't Assigned bid 4 - Violating LimitLines  
Crew member 8 wasn't Assigned bid 5 - Violating LimitLines

Όπως παρατηρούμαι και εδώ, η ανάθεση των αδειών ξεκινάει κάνοντας πλήρως ανάθεση των αδειών σε κάθε εργαζόμενο σωστά. Παρόλα αυτά ο τελευταίος εργαζόμενος δείχνει να μην του έχει ανατεθεί καμία άδεια. Αυτό οφείλεται στο γεγονός ότι όλες οι μέρες που ο όγδοος εργαζόμενος έχει δηλώσει να ξεκινάνε οι μέρες του, έχουν ήδη ανατεθεί στους προηγούμενους του, λόγω ιεραρχίας. Αναφερόμενοι και στα παραπάνω, γνωρίζουμε πλέον πως ελέγχονται όλες οι επιθυμίες του πρώτου μέλους του πληρώματος, ανατίθενται οι άδειες και μετά το μοντέλο περνά στο επόμενο, έτσι ο κίνδυνος να μην ανατεθεί κάποια άδεια αυξάνεται.

Το πρόβλημα αυτό θα μπορούσε να λυθεί, προσπαθώντας να εφαρμοστεί κάποια άλλη λογική ανάθεσης, όπως η Fair Priority που πολύ σύντομα αναφερθήκαμε σε αυτή ή να εμπλουτίσουμε το κώδικα, να ενημερώνει τον χρήστη πως οι ημερομηνίες που δηλώνονται μπορεί να οδηγήσουν σε μη ανάθεση αδειών.