<u>Πανεπιστήμιο Θεσσαλίας</u>
<u>Πολυτεχνική Σχολή</u>

Τμήμα Μηχανολόγων Μηχανικών

<u>Διπλωματική εργασία</u>

Αλγόριθμοι δημιουργίας στηλών για βέλτιστο χρονοπρογραμματισμό ιπταμένων με ταυτόχρονη ικανοποίηση για ημέρες ανάπαυσης

Επίθετο: Κιουτσούκη

Όνομα: Μαρία

Α/Μ:1650

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του Διπλώματος Μηχανολόγου Μηχανικού

ΒΟΛΟΣ 2018

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. Κοζανίδης Γεώργιος

Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,

Πανεπιστήμιο Θεσσαλίας

Δεύτερος εξεταστής:

Δρ. Παντελής Δημήτρης

Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,

Πανεπιστήμιο Θεσσαλίας

Τρίτος εξεταστής:

Δρ. Σαχαρίδης Γεώργιο

Επίκουρος καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,

Πανεπιστήμιο Θεσσαλίας

Τμήμα Μηχανολόγων Μηχανικών

Αλγόριθμοι δημιουργίας στηλών για βέλτιστο χρονοπρογραμματισμό ιπταμένων με ταυτόχρονη ικανοποίηση για ημέρες ανάπαυσης

Επίθετο: Κιουτσούκη

Όνομα: Μαρία

Α/Μ:1650

Επιβλέπων καθηγητής: Κοζανίδης Γεώργιος

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του Διπλώματος Μηχανολόγου Μηχανικού

ΒΟΛΟΣ 2018

ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτα από όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας κ. Κοζανίδη Γεώργιο για την πολύτιμη βοήθεια και την αμέριστη συμπαράσταση του καθ' όλη τη διάρκεια εκπόνησης. Επίσης ευχαριστώ τα υπόλοιπα μέλη της τριμελούς επιτροπής.

Θα ήθελα να ευχαριστήσω θερμά όλους μου τους φίλους που πίστεψαν σε εμένα και με ενθάρρυναν σε κάθε στάδιο των σπουδών μου, καθώς και όσους συναδέλφους συμφοιτητές συνέβαλλαν με τα σχόλια, την κρητική και τις γνώσεις τους στην αντιμετώπιση δυσκολιών.

Ιδιαίτερες ευχαριστίες θέλω να εκφράσω προς την οικογένεια μου και κυρίως του γονείς μου για την διαχρονική συμπαράσταση τους και την ηθική και υλική στήριξη των επιλογών μου.

Κιουτσούκη Μαρία

ΠΕΡΙΛΗΨΗ

Η αεροπορική βιομηχανία αποτελούσε πάντα ένα ιδιαίτερα αναπτυσσόμενο και προηγμένο τεχνολογικά κλάδο στον σύγχρονο κόσμο. Η εξέλιξή της ήταν ραγδαία τον προηγούμενο αιώνα και αυτό κυρίως οφείλεται στο κομμάτι της βελτιστοποίησης της λειτουργίας της. Πλέον τα μοντέλα βελτιστοποίησης έχουν έρθει πιο κοντά στην πραγματικότητα και είναι επαρκώς επιλύσιμα. Ο χρόνος επίλυσης προβλημάτων μεγάλης κλίμακας και πολυπλοκότητας έχει γίνει κατά πολύ μικρότερος, με αποτέλεσμα τα προβλήματα αυτά να μπορούν να επιλυθούν και από μικρότερα σε ισχύ συστήματα και να προσφέρουν στους χρήστες τους άμεσα και έγκυρα αποτελέσματα. Έτσι λοιπόν γίνεται αντιληπτό ότι οι μέθοδοι βελτιστοποίησης έχουν γίνει σημαντικό εργαλείο των αεροπορικών εταιριών στην προσπάθεια τους να διατηρήσουν αλλά και να επεκτείνουν την θέση τους στην άκρως ανταγωνιστική αγορά των μεταφορών.

Στην παρούσα διπλωματική εργασία αναπτύσσουμε έναν αλγόριθμο για την επίλυση του προβλήματος της κάλυψης ενός συνόλου πτήσεων από έναν συγκεκριμένο αριθμό ιπταμένων ικανοποιώντας ταυτόχρονα τις επιθυμίες ανάπαυσης τους.

Αρχικά περιγράφουμε τον τρόπο με τον οποίο λειτουργούν οι αεροπορικές εταιρίες και τα βασικά προβλήματα που έχουν να αντιμετωπίσουν.

Στην συνέχεια γίνεται μια αναφορά στις διάφορες μεθόδους/τεχνικές που χρησιμοποιούνται για την επίλυση αυτών των προβλημάτων καθώς και στην μέθοδο που θα χρησιμοποιηθεί σε αυτήν την εργασία, η οποία είναι branch and price. Η μέθοδος αυτή αποτελείται από το master problem και το column generation sub-problem.

Η μαθηματική μοντελοποίηση καθώς και οι εξισώσεις του προβλήματος παραθέτονται με λεπτομέρεια στα κεφάλαια 3 και 4. Τα βήματα που ακολουθεί η μοντελοποίηση είναι η αρχικοποίηση του restricted master problem, η επίλυση του master LP relaxation, η επίλυση του column generation sub-problem και η εφαρμογή του βήματος branching.

Τέλος παρουσιάζονται με λεπτομέρεια όλα τα αποτελέσματα των προβλημάτων που επιλύθηκαν για να αποδειχθεί η αποτελεσματικότητα του αλγορίθμου καθώς και τα συμπεράσματα που προκύπτουν από αυτά.

Πίνακας περιεχομένων

Πίνακας εικόνων

Πίνακας πινάκων

Πίνακας σχημάτων

ΚΕΦΑΛΑΙΟ 1

1.1 Γενική περιγραφή του προβλήματος

Η αποτελεσματική λειτουργία μιας αεροπορικής εταιρίας, απαιτεί τον κατάλληλο προγραμματισμό και τις απαραίτητες μεθοδολογίες, που χρησιμεύουν σε κάθε μία από τις ιδιαίτερα πολύπλοκες και πολυδιάστατες δραστηριότητες μιας τέτοιας επιχείρησης. Η χρήση της επιχειρησιακής έρευνας, και των μοντέλων βελτιστοποίησης ειδικότερα, έχει γίνει απαραίτητη στο χώρο των εναέριων μεταφορών, λόγω της πολυπλοκότητας διαχείρισης συστημάτων τέτοιας μεγάλης κλίμακας. Η επιχειρησιακή έρευνα έχει δημιουργήσει ένα σοβαρό αντίκτυπο στον τρόπο με τον οποίο οι σημερινές αεροπορικές εταιρίες διοικούνται και ανταγωνίζονται η μια την άλλη. Οδηγούμενες από την επιτακτική ανάγκη των διοικήσεων για κατάκτηση ανταγωνιστικού μεριδίου αγοράς, οι αεροπορικές εταιρίες στρέφονται σε ανεπτυγμένες τεχνικές βελτιστοποίησης, ώστε να δημιουργήσουν τα κατάλληλα συστήματα υποστήριξης αποφάσεων, για τον καλύτερο έλεγχο και λειτουργία της "αεροπορικής" επιχείρησης και των διαδικασιών της. Ο κύριος στόχος των αεροπορικών εταιριών είναι η μείωση του κόστους στις καθημερινές τους λειτουργίες. Έχουν γίνει αρκετές έρευνες και μελέτες σχετικά με τη διάρθρωση του κόστους των αεροπορικών εταιριών και τα ευρήματα ήταν ότι μετά το κόστος των καυσίμων που δεν μπορεί να ελεγχθεί από την αεροπορική εταιρία, το δεύτερο υψηλότερο κόστος είναι το κόστος προγραμματισμού του πληρώματος. Το κόστος του πληρώματος καλύπτει βασικά τους μισθούς, τις αποζημιώσεις κατά την έξοδο από τη βάση, το ξενοδοχείο, τις μεταφορές κλπ. Αυτές οι δαπάνες είναι πιθανών ο πιο σημαντικός τομέας για τις ενδεχόμενες εξοικονομήσεις αεροπορικών εταιριών και όταν δεν προγραμματιστεί σωστά, μπορεί να κοστίσει στην αεροπορική εταιρία και ως εκ τούτου, να δημιουργηθεί ένα τεράστιο πρόβλημα. Έχοντας αυτό υπόψη, οι αεροπορικές εταιρίες προσπαθούν να μειώσουν ή να εξοικονομήσουν το κόστος εφαρμόζοντας αποδοτικές και αποτελεσματικές λύσεις για την επίλυση των προβλημάτων προγραμματισμού του πληρώματος. Τα προβλήματα προγραμματισμού του πληρώματος μπορούν να οριστούν ως εκείνοι οι παράγοντες που επηρεάζουν τις λειτουργίες του πληρώματος. Οι παράγοντες μπορεί να είναι ακυρώσεις πτήσεων λόγω καιρού, μη διαθεσιμότητας εξοπλισμού, καθυστερήσεων πτήσεων, μη διαθεσιμότητα του πληρώματος λόγω ασθένειας, αδειών κτλ.

Η παρούσα διπλωματική εργασία ασχολείται με το πρόβλημα της κάλυψης ενός συνόλου διαδρομών (routes), σε ένα ολοκληρωμένο δίκτυο διανομής, από ένα συγκεκριμένο αριθμό πιλότων, ικανοποιώντας ταυτόχρονα και ένα σύνολο ποιοτικών κριτηρίων, όπως η μη κάλυψη αλληλεπικαλυπτόμενων χρονικά αποστολών από τον ίδιο πιλότο, οι μη διαθέσιμες ημερομηνίες πτήσης κάθε πιλότου, η προτίμηση του κάθε πιλότου για ημέρες αδειών κ.α. Η χρονική περίοδος, για την οποία λύνεται το πρόβλημα είναι για διάστημα ενός μήνα.

1.2 Αεροπορικές εταιρίες

Οι αερομεταφορές διαμορφώνουν ένα μοναδικό δίκτυο που ενώνει τους ανθρώπους, τις χώρες και τις κουλτούρες και παίζουν ζωτικό ρόλο στην περαιτέρω ολοκλήρωση και την ανάπτυξη. Γίνονται όλο και περισσότερο προσβάσιμες σε μεγαλύτερο αριθμό ανθρώπων οι οποίοι σήμερα έχουν την οικονομική δυνατότητα να ταξιδεύουν μέσω αέρα τόσο για επαγγελματικούς όσο και για λόγους αναψυχής. Οι αερομεταφορές είναι μια μεγάλη βιομηχανία. Περιλαμβάνει περίπου 18.000 αεροσκάφη που λειτουργούν σε 10.000 αεροδρόμια και έχουν ετήσιο κύκλο εργασιών 260 δισεκατομμύρια δολάρια. Πάνω από 1,6 δισεκατομμύρια επιβάτες παγκοσμίως χρησιμοποιούν τις αεροπορικές εταιρείες για επιχειρηματικά ταξίδια ή ταξίδια αναψυχής, πραγματοποιώντας ετήσια 3.400 δισεκατομμύρια επιβατοχιλιόμετρα.

Η έρευνα του οργανισμού EUROCONTROL δείχνει ότι μέχρι το 2030 αυτός ο αριθμός θα φθάσει τα 2,3 δισεκατομμύρια επιβάτες. Πάνω από τους μισούς διεθνείς τουρίστες (με εξαίρεση εκείνων που ταξιδεύουν εντός της Ευρώπης) πετούν στους προορισμούς των διακοπών τους. Οι αερομεταφορές επίσης βελτιώνουν την ποιότητα της ζωής των καταναλωτών παρέχοντάς τους ευρύτερες επιλογές σε αγαθά και πρόσβαση σε ταξίδια στο εξωτερικό για αναψυχή, πολιτιστική και πολιτική ανταλλαγή. Οι αερομεταφορές είναι υπεύθυνες για τη μεταφορά του 30% έως 40% του διεθνούς φορτίου σε αξία. Είναι ο κύριος πάροχος μεταφορικών υπηρεσιών στον τομέα του τουρισμού και κύριο μέσο μεταφοράς για τα στελέχη της σύγχρονης βιομηχανίας. Σε τοπικό επίπεδο, τα αεροδρόμια είναι ο μεγαλύτερος καταλύτης για την τοπική ανάπτυξη (Button et al, 2004).

Πίνακας 1: Συμβολή των αερομεταφορών

	2018	2030 (πρόβλεψη)
Αεροσκάφη	18.000	
Αεροδρόμια	10.000	
Οικονομικός τζίρος (δις $)	260	
Μεταφορά επιβατών/έτος	1,6 δις	2,3 δις
Διανυόμενες αποστάσεις σε επιβατοχιλιόμετρα	3.400 δις	
Μεταφορά εμπορευμάτων σε αξία επί % συνόλου	30-40	

1.2.1 Το προϊόν και το περιβάλλον λειτουργίας των αεροπορικών εταιριών

Το αντικείμενο μιας αεροπορικής εταιρίας είναι ξεχωριστό. Το προϊόν που προσφέρεται από τις αεροπορικές εταιρίες αντιπροσωπεύεται από πτήσεις που μεταφέρουν επιβάτες ή εμπορεύματα από διάφορες τοποθεσίες σε διαφορετικούς προορισμούς. Η εμπορευσιμότητα και η επιτυχία συνάμα αυτού του προϊόντος εξαρτάται σε μεγάλο βαθμό από τους εξής παράγοντες:

· Ακρίβεια

· Επικαιρότητα

· Λειτουργικότητα

· Ποιότητα

· Κόστος εξυπηρέτησης

· Εύκολη πρόσβαση στο προϊόν

Τα παραπάνω κριτήρια γίνονται αντιληπτά από τον πελάτη επιβάτη ως, ευέλικτα ωράρια πτήσεων, ακριβής αναχώρηση πτήσεων δίχως καθυστερήσεις, ικανοποιητική εξυπηρέτηση εν ώρα πτήσης, κατάλληλη μεταχείριση των αποσκευών και προσιτός εύκολος τρόπος έκδοσης εισιτηρίων. Για να ανταπεξέλθουν επιτυχώς στις παραπάνω απαιτήσεις και για να προσφέρουν ένα υψηλής ποιότητας αλλά και χαμηλού κόστους προϊόν, οι αεροπορικές εταιρίες επενδύουν τεράστια υλικά αλλά και ψυχικά / πνευματικά αποθέματα στην προσπάθεια να εξελίξουν οικονομικά αποδοτικές μεθόδους αντιμετώπισης των κυριότερων προβλημάτων τους, όπως είναι κατηγοριοποίηση των τιμών των εισιτηρίων, ο προγραμματισμός των πτήσεων, σχεδιασμός δικτύου πτήσεων, ακολουθία των πτήσεων, ταίριασμα πληρωμάτων, ανάθεση πυλών αναχώρησης και άφιξης ανά αεροδρόμιο, πρόγραμμα συντήρησης αεροσκαφών, προγράμματα εκπαίδευσης πληρωμάτων καθώς και διαδικασίες διαχείρισης των αποσκευών. Η πολυπλοκότητα που διέπει τα παραπάνω προβλήματα προέρχεται από το ίδιο το περιβάλλον στο οποίο λειτουργεί μια αεροπορική, πιο αναλυτικά:

-Το τεράστιο επιχειρηματικό μέγεθος όσο αφορά, το γεωγραφικό εύρος στο οποίο εκτείνεται η επιχείρηση, τα αποθέματα που υποχρεωτικά πρέπει να υπάρχουν στην διάθεση της επιχείρησης (μηχανήματα, προσωπικό κτλ.), το μέγεθος και η ποικιλομορφία της αγοράς στην οποία απευθύνεται η επιχείρηση. Το εξαιρετικά δυναμικό περιβάλλον μέσα στο οποίο κινείται μια τέτοιου είδους επιχείρηση το οποίο εναλλάσσεται διαρκώς. Δεν είναι λίγες οι φορές που η εταιρία καλείται να αλλάξει τα πρωταρχικά της πλάνα, είτε λόγω καιρού, είτε λόγω μηχανικών προβλημάτων των αεροσκαφών, είτε λόγω ασθένειας του πληρώματος, είτε ακόμα και από τις διακυμάνσεις για παράδειγμα στην τιμή του πετρελαίου, ή μιας απρόσμενης τρομοκρατικής ενέργειας που μπορεί να επηρεάσει το μέγεθος της αγοράς (παράδειγμα η $11^{η}$ Σεπτεμβρίου).

-Οι νομικοί και άλλοι περιορισμοί που διέπουν τις εναέριες μεταφορές και μπορεί να σχετίζονται με την συντήρηση-κατάσταση των αεροσκαφών ή την νομιμότητα όσο αφορά την εκπαίδευση του πληρώματος και την ετοιμότητα του, καθώς και τα μέτρα ασφαλείας που πρέπει να λαμβάνονται.

-Τα στενά οικονομικά όρια που τίθενται στον προγραμματισμό και σχεδιασμό της εταιρίας, ώστε να εκμεταλλευτούν όλοι οι δυνατοί πόροι στη μέγιστη αποδοτικότητα τους, δεν αφήνουν περιθώρια ελιγμών ή λάθους στα συστήματα οργάνωσης και διοίκησης που αφορούν την επιχείρηση.

1.2.2 Η επιχειρησιακή έρευνα στον χώρο των αεροπορικών εταιριών

Για να διατηρήσουν μια σταθερή αποδοτικότητα, οι αεροπορικές εταιρίες, πρέπει να αντιλαμβάνονται άμεσα τις παρούσες συνθήκες της αγοράς προκειμένου να ισορροπήσουν την προσφορά με τη ζήτηση. Λαμβάνοντας λοιπόν υπόψη την γεωγραφική διασπορά της επιχείρησής τους, των υψηλών επενδυτικών δαπανών (τόσο σε χρήμα όσο και σε εργασία) και των αναμενόμενων εμποδίων που απορρέουν από την διεθνή αεροπλοΐα, δεν είναι καθόλου παράξενο, το ότι η αεροπορική βιομηχανία, έχει οδηγήσει άλλες επιστήμες, όπως η επιχειρησιακή έρευνα, στην εφαρμογή ειδικών τεχνικών βελτιστοποίησης για να αντιμετωπίσει τα επιχειρησιακά ζητήματα. Από τεχνολογικής άποψης, η ταχύτητα των επεξεργαστών Η/Υ είναι πλέον ικανοποιητική για την επίλυση πολύπλοκων και μεγάλης κλίμακας προβλημάτων σε πραγματικό χρόνο, η χωρητικότητα των αποθηκευτικών συσκευών των Η/Υ είναι αρκετά μεγάλη για να φιλοξενήσει επιχειρησιακά δεδομένα, το διαδίκτυο είναι αρκετά αξιόπιστο για να μεταφέρει με διάφορους τρόπους ταχύτατα τις απαραίτητες πληροφορίες και δεδομένα στους άμεσα ενδιαφερόμενους. Παράλληλα το κόστος όλου αυτού του απαραίτητου τεχνολογικού εξοπλισμού, είναι αρκετά χαμηλό, που να εξασφαλίζει την άμεση απόσβεση του.

Από μεθοδολογικής άποψης, τις τελευταίες δύο δεκαετίες έχει υπάρξει μια θεαματική ανάπτυξη όσο αφορά τους αλγόριθμους βελτιστοποίησης και τις τεχνικές επίλυσης προβλημάτων. Πλέον τα μοντέλα βελτιστοποίησης έχουν έρθει πιο κοντά στην πραγματικότητα και παρόλα αυτά είναι επαρκώς επιλύσιμα. Ο χρόνος επίλυσης προβλημάτων μεγάλης κλίμακας και πολυπλοκότητας έχει γίνει κατά πολύ μικρότερος, με αποτέλεσμα τα προβλήματα αυτά να μπορούν να επιλυθούν και από μικρότερα σε ισχύ συστήματα και να προσφέρουν στους χρήστες τους άμεσα και έγκυρα αποτελέσματα. Έτσι λοιπόν γίνεται αντιληπτό ότι, οι μέθοδοι της επιχειρησιακής έρευνας που περιλαμβάνουν μεθόδους βελτιστοποίησης, τεχνικές επίλυσης και συστήματα υποστήριξης αποφάσεων έχουν γίνει σημαντικό εργαλείο των αεροπορικών στην προσπάθεια τους να διατηρήσουν αλλά και να επεκτείνουν την θέση τους στην άκρως ανταγωνιστική αγορά των μεταφορών. Μερικές από τις δραστηριότητες-λειτουργίες των αεροπορικών, για τις οποίες τα τελευταία χρόνια έχουν εξελιχθεί μέθοδοι επίλυσης και βελτιστοποίησης από τον χώρο της επιχειρησιακής έρευνας είναι οι παρακάτω:

· Σχεδιασμός δικτύου και κατασκευή προγράμματος πτήσεων

· Ανάθεση αεροσκαφών σε πτήσεις

·Ακολουθίες πτήσεων που ανατίθενται σε αεροσκάφη (δρομολόγηση αεροσκαφών)

· Βελτιστοποίηση ταιριάσματος πληρωμάτων

· Ανάθεση πληρωμάτων σε πτήσεις

· Διαχείριση εισροών

· Διαχείριση μη προγραμματισμένων ενεργειών

· Διαχείριση εναέριας κυκλοφορίας

1.3 Επαγγελματική ικανοποίηση

Η επαγγελματική ικανοποίηση αποτελεί πολύ σημαντικό θέμα διότι σχετίζεται τόσο με τη ψυχική υγεία του εργατικού δυναμικού όσο και με το ενδιαφέρον των επιχειρήσεων να έχουν υψηλή αποδοτικότητα και ικανοποιημένο προσωπικό. Αποτελεί ένα από τα βασικά θέματα της οργανωτικής ψυχολογίας, κυρίως γιατί θεωρείται πως έχει άμεση σχέση τόσο με την ψυχική υγεία του εργατικού δυναμικού, όσο και με το ενδιαφέρον των επιχειρήσεων να έχουν υψηλή αποδοτικότητα και σε πολλές περιπτώσεις σταθερό, μόνιμο και ικανοποιημένο προσωπικό. Η επαγγελματική ικανοποίηση έχει θεωρητικά συσχετιστεί με τα κίνητρα, τις στάσεις και τις αξίες της εργασίας. Ικανοποιημένοι εργαζόμενοι συνεπάγεται αύξηση της αποδοτικότητας και άρα αύξηση των εσόδων της αεροπορικής εταιρίας. Για τον λόγο αυτό στην παρούσα διπλωματική εργασία δίνεται έμφαση στην ικανοποίηση των εργαζομένων σχετικά με τις ημέρες άδειας που αυτοί έχουν ζητήσει. Ωστόσο η επαγγελματική ικανοποίηση δεν αφορά μόνο την ικανοποίηση των εργαζομένων στο κομμάτι των αδειών αλλά περιέχει ένα σύνολο άλλων παραγόντων όπως:

-ικανοποίηση από την εταιρία και την διοίκηση

-ικανοποίηση από τους προϊσταμένους

-ικανοποίηση από τους συναδέλφους

-ικανοποίηση από τις συνθήκες εργασίας

-ικανοποίηση από το αντικείμενο εργασίας

-ικανοποίηση από τις δυνατότητες προαγωγής και το status

1.4 Μεθοδολογία Column Generation

Η μεθοδολογία column generation είναι ένας αποτελεσματικός αλγόριθμος για την επίλυση μεγαλύτερων γραμμικών προγραμμάτων. Η γενική ιδέα είναι ότι πολλά γραμμικά προγράμματα είναι πολύ μεγάλα για να εξετάσουν ρητά όλες τις μεταβλητές. Δεδομένου ότι οι περισσότερες από τις μεταβλητές δεν είναι βασικές και έχουν μηδενική τιμή στην βέλτιστη λύση, μόνο μια υποστοιχία μεταβλητών πρέπει να εξεταστεί θεωρητικά κατά την επίλυση του προβλήματος. Το Column generation αξιοποιεί αυτή την ιδέα για να παράγει μόνο τις μεταβλητές που έχουν τη δυνατότητα να βελτιώσουν την αντικειμενική συνάρτηση δηλαδή να βρει μεταβλητές με αρνητικό μειωμένο κόστος (υποθέτοντας χωρίς βλάβη της γενικότητας ότι το πρόβλημα είναι πρόβλημα ελαχιστοποίησης).

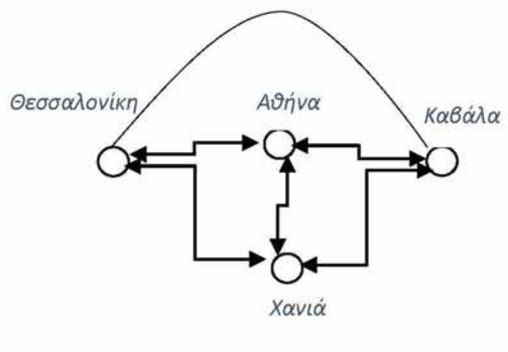
Το πρόβλημα που επιλύεται χωρίζεται σε δύο προβλήματα: το master problem και το column generation sub-problem. Το master problem είναι το αρχικό πρόβλημα με μόνο ένα υποσύνολο μεταβλητών που εξετάζονται. Το column generation sub-problem είναι ένα νέο πρόβλημα που δημιουργείτε για να καθορίσει μια νέα μεταβλητή. Η αντικειμενική συνάρτηση του column generation sub-problem είναι το μειωμένο κόστος της νέας μεταβλητής σε σχέση με τις τρέχουσες δυαδικές μεταβλητές και οι περιορισμοί απαιτούν η μεταβλητή να υπακούει στους φυσικά υπάρχοντες περιορισμούς.

Η διαδικασία λειτουργεί ως εξής. Το master problem επιλύεται, από αυτή τη λύση, αποκτούμε δυικές τιμές για κάθε έναν από τους περιορισμούς στο master problem. Οι πληροφορίες αυτές στη συνέχεια χρησιμοποιούνται στην αντικειμενική συνάρτηση του column generation sub-problem. Το column generation sub-problem λύνεται. Εάν η αντικειμενική τιμή του column generation sub-problem είναι αρνητική, εντοπίζεται μια μεταβλητή με αρνητικό μειωμένο κόστος. Αυτή η μεταβλητή προστίθεται έπειτα στο master problem και το master problem επιλύεται εκ νέου. Η επανεξέταση του master problem θα δημιουργήσει μια νέα σειρά δυικών τιμών και η διαδικασία θα επαναλαμβάνεται έως ότου να μην εντοπιστούν αρνητικές μειωμένες μεταβλητές κόστους. Όταν το column generation sub-problem επιστρέψει μια λύση με μη αρνητικό μειωμένο κόστος, μπορούμε να συμπεράνουμε ότι η λύση στο master problem είναι βέλτιστη. Σε πολλές περιπτώσεις, αυτό επιτρέπει την επίλυση μεγάλων γραμμικών προγραμμάτων τα οποία είχαν θεωρηθεί προηγουμένως ανυπόστατα.

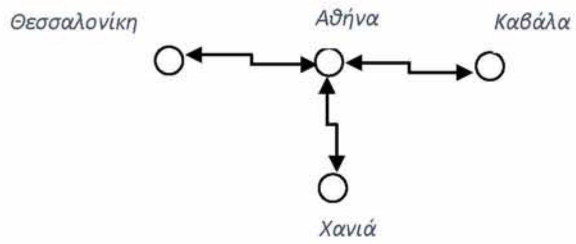
ΚΕΦΑΛΑΙΟ 2

2.1 Σχεδιασμός δικτύου και κατασκευή προγράμματος πτήσεων

Η κατασκευή του προγράμματος πτήσεων ανά προορισμό καθώς και η κατασκευή του δικτύου που θα περιλαμβάνει το σύνολο των προορισμών που εξυπηρετεί μια αεροπορική εταιρία, είναι πρωταρχικής σημασίας για την σωστή και κερδοφόρα λειτουργία της. Το πρόγραμμα των πτήσεων συγκεκριμένα 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, τη διαθεσιμότητα και τα χαρακτηριστικά του στόλου της, το διαθέσιμο ανθρώπινο δυναμικό αλλά και από την συμπεριφορά του ανταγωνισμού. Πριν όμως από την δημιουργία συγκεκριμένου προγράμματος πτήσεων, η αεροπορική εταιρία καλείται να καθορίσει το δίκτυο της, το δίκτυο δηλαδή των επίγειων σταθμών ανάμεσα στους οποίους θα κινείται. Οι αλλαγές στις απαιτήσεις της αγοράς καθώς και οι αλλαγές στα νομικά πλαίσια που διέπουν την αεροπλοΐα και έχουν επιφέρει την απελευθέρωση στην επιλογή προορισμών για κάθε αεροπορική, έχουν αλλάξει τον τρόπο σχεδιασμού των δικτύων πτήσεων, από το πρότυπο «από σημείο σε σημείο», (σχήμα 2.1.α), που επικρατούσε παλαιότερα και περιλάμβανε ουσιαστικά ζεύγη ανταποκρινόμενων σταθμών, στο πρότυπο ενός κεντρικού σταθμού και διαφόρων περιφερειακών σταθμών που βρίσκονται σε μια ακτίνα γύρω από τον κεντρικό (πρότυπο Hub and Spoke) (σχήμα 2.1.β), ανάλογα με την εμβέλεια των δυνατοτήτων της αεροπορικής εταιρίας.



Σχήμα 2.1.α: Δίκτυο πτήσεων βάση του προτύπου "από σημείο σε σημείο»



Σχήμα 2.1.β: Δίκτυο πτήσεων βάση του προτυπου ενός κεντρικού και πολλών περιφερειακών σταθμών

Διάφορες έρευνες έχουν γίνει για την βελτιστοποίηση όχι μόνο του δικτύου πτήσεων αλλά και διαφόρων υπό-προβλημάτων που σχετίζονται με αυτό, όπως η εύρεση κατάλληλης τοποθεσίας για τον κεντρικό σταθμό και η συχνότητα των πτήσεων που καλύπτουν το δίκτυο. Για την επιλογή του κεντρικού σταθμού έχουν χρησιμοποιηθεί μέθοδοι διακλάδωσης και φραγμών, καθώς και μαθηματικού προγραμματισμού. Για τον προσδιορισμό βέλτιστης συχνότητας πτήσεων έχουν προταθεί μέθοδοι όπως η θεωρία ακαθόριστων συνόλων.

2.2 Ανάθεση αεροσκαφών σε πτήσεις (fleet assignment)

Το πρόβλημα αντιστοίχισης στόλου (FAP) ασχολείται με την εκχώρηση τύπων αεροσκαφών, καθένα από τα οποία έχει διαφορετική χωρητικότητα, στις προγραμματισμένες πτήσεις, με βάση τις δυνατότητες και τις διαθέσιμες δυνατότητες εξοπλισμού, το λειτουργικό κόστος και τα πιθανά έσοδα. Τα έσοδα μιας αεροπορικής εταιρίας επηρεάζονται σημαντικά από αυτό και, ως εκ τούτου, αποτελεί βασικό στοιχείο της συνολικής διαδικασίας προγραμματισμού. Ωστόσο, λόγω του μεγάλου αριθμού πτήσεων που προγραμματίζονται κάθε μέρα και της εξάρτησης του FAP από άλλες αεροπορικές διεργασίες, η επίλυση του FAP ήταν πάντα ένα δύσκολο έργο για τις αεροπορικές εταιρίες.
Ο προγραμματισμός ανάθεσης των αεροσκαφών γίνεται στην πραγματικότητα μήνες πριν την πρακτική εφαρμογή του, καθώς δίδεται έτσι το περιθώριο αναθεώρησης και βελτίωσης του αποτελέσματος κάτω από τις πραγματικές υπάρχουσες συνθήκες. Σημαντική παρατήρηση αποτελεί ότι κατά το σχεδιασμό ανάθεσης πτήσεων σε αεροσκάφη, γίνεται η υπόθεση ότι το πρόγραμμα των πτήσεων παραμένει σταθερό και επαναλαμβάνεται περιοδικά (ανά μέρα ή εβδομάδα συνήθως), παρόλο που κάτι τέτοιο μπορεί να μην συμβαίνει στην πραγματικότητα (π.χ. απρογραμμάτιστες πτήσεις charter).

Η πλειονότητα των μεθόδων που έχουν χρησιμοποιηθεί για την βελτιστοποίηση και επίλυση του προβλήματος ανάθεσης αεροσκαφών σε πτήσεις, 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. Ταυτόχρονα υπάρχουν και τόξα που αναπαριστούν την επίγεια παραμονή του αεροσκάφους μέχρι την ώρα της επόμενης αναχώρησης του. Κάθε ένας συνδυασμός κόμβων και τόξων θα πρέπει να ανατίθεται σε ένα και μόνο αεροσκάφος, ώστε να ικανοποιείται ο λογικός περιορισμός που απαιτεί κάθε πτήση να εκτελείται μόνο μια φορά και από ένα αεροσκάφος. Άλλοι περιορισμοί που λαμβάνονται συνήθως υπόψη, είναι ο αριθμός των διαθέσιμων αεροσκαφών αλλά και των χαρακτηριστικών τους (π.χ. πτητική εμβέλεια, χωρητικότητα). Το μοντέλο αυτό καταλήγει να επιλύεται συνήθως με μεθόδους μικτού, ακέραιου προγραμματισμού.

Ένας άλλος τρόπος μοντελοποίησης του προβλήματος, πολύ πιο αναλυτικός είναι με την δημιουργία ενός δικτύου συνδέσεων, όπου οι κόμβοι είναι οι πτήσεις και τα τόξα η σύνδεση μεταξύ των πτήσεων. Μιας και οι πιθανοί τρόποι σύνδεσης των πτήσεων μεταξύ τους είναι πολλοί, συχνά οδηγούμαστε σε μεγαλύτερης κλίμακας δίκτυα από ότι στην περίπτωση του χωροχρονικού δικτύου. Από την άλλη μεριά όμως το δεύτερο μοντέλο είναι πολύ πιο ακριβές και κοντά στο πραγματικό δίκτυο πτήσεων.

2.3 Δρομολόγηση αεροσκαφών (aircraft routing)

Το πρόβλημα δρομολόγησης αεροσκαφών, είναι ουσιαστικά το πρόβλημα δημιουργίας ενός δρομολογίου για κάθε αεροσκάφος του στόλου της αεροπορικής εταιρίας, δεδομένου του συνόλου των πτήσεων που πρέπει να πραγματοποιήσει η εταιρία, έχοντας ως στόχο την ελαχιστοποίηση του κόστους. Ένας δεύτερος τρόπος μοντελοποίησης του προβλήματος είναι έμμεσος και προσπαθεί να μεγιστοποιήσει έμμεσες αξίες όπως την ικανοποίηση της επιθυμίας των πελατών για περισσότερες πτήσεις χωρίς ενδιάμεσους σταθμούς, μεταξύ διαφόρων πόλεων. Τα δημιουργούμενα δρομολόγια θα πρέπει βέβαια να υπακούουν σε κάποιους περιορισμούς. Δυο τυπικοί περιορισμοί στην κατασκευή δρομολογίων είναι ο περιορισμός συντήρησης των αεροσκαφών και ο περιορισμός εξισορρόπησης της διανομής των πτήσεων στα διάφορα αεροσκάφη. Ο περιορισμός συντήρησης, αφορά φυσικά τους τεχνικούς ελέγχους και επισκευές που θα πρέπει να γίνονται ανά τακτά χρονικά διαστήματα, στα αεροσκάφη. Ο χρόνος που θα πετάει ένα αεροσκάφος συνεχόμενα χωρίς συντήρηση είναι περιορισμένος και πρέπει να υπακούει στους κανόνες ασφαλείας. Υπάρχουν πολλών ειδών συντηρήσεις που γίνονται σε αεροσκάφη, υπάρχουν έλεγχοι που κρατάνε λίγες ώρες όπως επίσης και επισκευές που μπορεί να διαρκέσουν και αρκετές μέρες, συνήθως όμως τέτοιου είδους μακροχρόνιες επισκευές δεν λαμβάνονται υπόψη κατά την μοντελοποίηση και επίλυση του προβλήματος, αφού είναι πιο σπάνιες. 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.

2.4 Crew scheduling

Σε αυτήν την φάση του προγραμματισμού τα μέλη του πληρώματος πρέπει να ανατεθούν σε πτήσεις. Από το πρόγραμμα δημιουργείτε μια σειρά από υπηρεσίες που περιέχουν όλες τις ακολουθίες των πτήσεων που μπορούν να εκτελεστούν εντός συγκεκριμένου χρονικού διαστήματος για τα μέλη του πληρώματος. Για μια εγχώρια πτήση, μια περίοδος υπηρεσίας συνήθως αντιπροσωπεύει μία μόνο ημέρα πτήσης. Για πτήσεις μακρινών αποστάσεων, μια πτήση μπορεί να αποτελείται από δύο ημέρες. Οι περισσότερες αποστολές πληρώματος καλύπτουν πολλές ημέρες που συνδέονται μεταξύ τους δημιουργώντας ζεύγη.

2.4.1 Περίοδος υπηρεσίας

Το σύνολο των πτήσεων που μπορούν να πραγματοποιηθούν από ένα πλήρωμα κατά τη διάρκεια μιας ημέρας εργασίας ονομάζεται περίοδος υπηρεσίας (duty period).
Οι υπηρεσίες συνδέονται άμεσα και με μια σειρά περιορισμών, όπως ο συνολικός αριθμός των ωρών πτήσεων που το πλήρωμα μπορεί να πραγματοποιήσει κατά τη διάρκεια μιας περιόδου υπηρεσιών. Ένας ακόμη περιορισμός είναι ότι οι πτήσεις πρέπει να ακολουθούν η μία την άλλη σε χώρο και χρόνο. Επιπλέον, υπάρχει περιορισμός στον ελάχιστο χρόνο αεργίας μεταξύ δύο συνεχόμενων πτήσεων, γνωστό ως χρόνο σύνδεσης ή χρόνο ξεκούρασης. Τέλος ένας περιορισμός είναι ο μέγιστος χρόνος που πρέπει να παρέλθει από περίοδο σε περίοδο υπηρεσίας.

2.4.2 Ζεύγη

Συχνά μια περίοδος υπηρεσίας δεν σταματά στην βάση του πληρώματος και έτσι το πλήρωμα αναγκάζετε να περάσει το βράδυ του σε κάποιο άλλο μέρος μέχρι να ξεκινήσει η επόμενη περίοδος υπηρεσίας. Το πλήρωμα μπορεί να χρειαστεί να περάσει οπουδήποτε από μία έως και πέντε μέρες στη σειρά μακριά από το σπίτι του. Μια σειρά από υπηρεσίες και στάσεις ονομάζεται ζεύγος. Γενικά, το πλήρωμα θα παραμείνει μαζί για όλες τις υπηρεσίες μέσα σε ένα ζεύγος.

2.4.3 Βελτιστοποίηση ταιριάσματος πληρώματος (crew pairing)

Ο στόχος του προβλήματος crew pairing των αεροπορικών εταιριών είναι να ελαχιστοποιηθούν οι σχετικές δαπάνες με την ανάθεση των πληρωμάτων στο σύνολο των πτήσεων που πρέπει να καλύψει η αεροπορική εταιρία. Κάθε πλήρωμα έχει μια βάση (home base) και ένα ζεύγος, που αποτελείται από μια σειρά πτήσεων, η οποία αρχίζει και τελειώνει στην έδρα βάσης του πληρώματος. Κάθε ζεύγος αποτελείται από μια σειρά νόμιμων εργάσιμων ημερών, που ονομάζονται υπηρεσίες 'duty periods' τα οποία χωρίζονται από περιόδους ανάπαυσης. Το crew pairing των αεροπορικών εταιριών επιδιώκει να βρει ένα υποσύνολο νόμιμων ζευγών έτσι ώστε να καλύπτεται κάθε πτήση από μόνο ένα πλήρωμα και έτσι το συνολικό κόστος αυτών των ζευγών να ελαχιστοποιείται.

Για να είναι νόμιμη, η κατασκευή ενός ζεύγους πρέπει να ληφθεί υπόψη ένα σύνολο περιορισμών. Κάποιοι από αυτούς τους περιορισμούς, όπως ο χρονικός και ο χωροταξικός περιορισμός δίνονται από τον ορισμό του προβλήματος, ενώ άλλοι είναι αποτελέσματα των νόμων και των κανονισμών. Συγκεκριμένα:

-<u>Χρονικοί περιορισμοί</u>: Η αναχώρηση μιας πτήσης πρέπει προφανώς να γίνει μετά την άφιξη της προηγούμενης πτήσης του ζεύγους. Επιπλέον, ένα ορισμένο χρονικό διάστημα, που ονομάζεται χρόνος μετάβασης, πρέπει να περάσει μεταξύ της άφιξης μιας πτήσης από την αναχώρηση της επόμενης.

-<u>Χωρικοί περιορισμοί</u>: Για κάθε δύο διαδοχικές πτήσης του ίδιου ζεύγους, η δεύτερη πρέπει να αναχωρήσει από το αεροδρόμιο που φτάνει η πρώτη.

-<u>Πρέπει να υπακούουν στους υπάρχοντες νόμους και σε συλλογικές συμβάσεις:</u> Αυτά συνήθως καθορίζουν τη μέγιστη διάρκεια ενός καθήκοντος (duty), τον μέγιστο χρόνο πτήσης που επιτρέπεται σε μια περίοδο, τον ελάχιστο χρόνο της περιόδου ανάπαυσης μεταξύ δύο καθηκόντων κλπ. Αυτές οι τιμές μπορεί να διαφέρουν ανάλογα με το μήκος ή τον προορισμό μιας πτήσης και συνήθως είναι αρκετά διαφορετικές για το πλήρωμα πιλοτηρίου και το πλήρωμα καμπίνας.

-<u>Περιορισμοί του στόλου</u>: Το πλήρωμα του πιλοτηρίου έχει εκπαιδευθεί να λειτουργεί μόνο ένα τύπου αεροσκάφους. Επομένως, όλες οι πτήσεις σε ένα ζεύγος που δημιουργήθηκαν για το πλήρωμα του πιλοτηρίου πρέπει να εκτελούνται από τον ίδιο τύπο αεροσκάφους. Από την άλλη πλευρά, το πλήρωμα της καμπίνας μπορεί να τοποθετηθεί σε οποιοδήποτε τύπο αεροσκάφους. Συνεπώς, τα ζευγάρια που είναι νόμιμα για το πλήρωμα καμπίνας μπορεί να είναι μη εφικτά για το πλήρωμα του πιλοτηρίου, κάνοντας έτσι το πρόβλημα εκχώρησης πληρώματος του πιλοτηρίου δυσκολότερο να επιλυθεί.

Πέρα από αυτούς τους περιορισμούς είναι πιθανό εκτός από το πλήρωμα που είναι σε υπηρεσία να χρειαστεί να ταξιδέψουν και άλλα μέλη του πληρώματος, οι οποίοι να ταξιδεύουν ως επιβάτες σε ένα συγκεκριμένο αεροδρόμιο και να συνεχίζουν από εκεί την υπηρεσία τους. Αυτή η διαδικασία είναι γνωστή ως «deadheaded crews» και επιφέρει πρόσθετο κόστος στην αεροπορική εταιρία, καθώς τα μέλη αυτά πληρώνονται κανονικά ακόμα και να δεν βρίσκονται σε υπηρεσία, χρησιμοποιώντας ταυτόχρονα θέσεις του αεροπλάνου που θα δίνονταν σε πελάτες/επιβάτες. Με μια πρώτη σκέψη θα έπρεπε να αποφεύγονται αυτές οι πτήσεις παρόλα αυτά είναι πιθανόν μια λύση με πτήσεις «deadheaded» να είναι καλύτερη από μια με χωρίς.

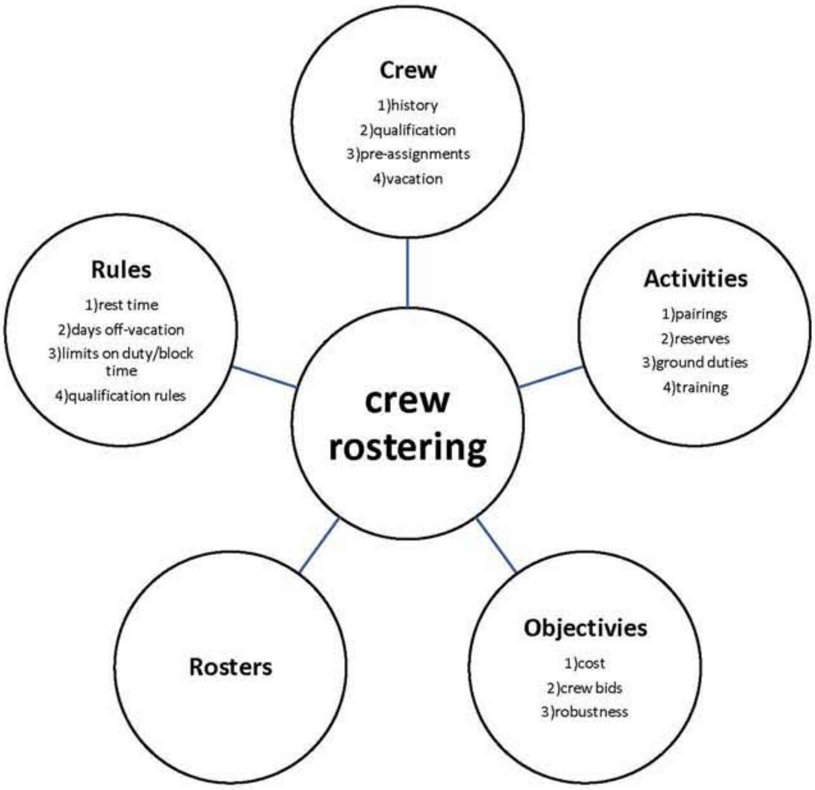
Το κόστος μιας λύσης είναι ένας άλλος παράγοντας που είναι κρίσιμος για το πρόβλημα. Συνήθως το κόστος για κάθε ζεύγος είναι ίσο με τους μισθούς του πληρώματος. Αυτό πάλι, εξαρτάται και από την πολιτική της αεροπορικής εταιρίας, αλλά γενικά αποτελεί συνάρτηση της πτήσης, του χρόνου υπηρεσίας ή του αριθμού των καθηκόντων ενός ζεύγους. Το συνολικό κόστος από μια λύση είναι συνήθως το άθροισμα του κόστους όλων των ζευγών σε αυτό, συν το κόστος από πτήσεις «deadheaded». Εξετάζοντας τη φύση των περιορισμών και τη συνάρτηση κόστους, είναι προφανές ότι η μοντελοποίηση τους δεν είναι ένα τετριμμένο καθήκον. Στην πραγματικότητα, στις περισσότερες περιπτώσεις, είναι αδύνατο να διατυπωθούν τα νόμιμα ζεύγη ή η συνάρτηση κόστους ως γραμμικές εξισώσεις και επομένως χρησιμοποιείτε ένα γραμμικό πλαίσιο προγραμματισμού για να λυθεί ολόκληρο το πρόβλημα κατευθείαν.

2.4.4 Ανάθεση πληρώματος σε πτήσεις (crew rostering)

Η ανάθεση του πληρώματος σε πτήσεις (crew rostering) των αεροπορικών εταιριών αποτελεί σημαντικό μέρος των αερομεταφορών και αποτελεί ένα ενδιαφέρον πρόβλημα για την έρευνα των επιχειρήσεων. Στόχος είναι να αντιστοιχισθούν ανώνυμοι συνδυασμοί πληρώματος είτε σε εξατομικευμένα δρομολόγια ή σε ανώνυμες γραμμές προσφοράς οι οποίες στη συνέχεια θα ανατεθούν σε μεμονωμένα μέλη του πληρώματος. Σε σύγκριση με το πρόβλημα ταιριάσματος του πληρώματος (crew pairing), το πρόβλημα της ανάθεσης σε πτήσεις (crew rostering) έχει λάβει πολύ λιγότερη προσοχή στο ακαδημαϊκό επίπεδο. Σήμερα το επικρατέστερο σύστημα που χρησιμοποιείται είναι το Carmen Crew Roster στις μεγαλύτερες ευρωπαϊκές εταιρίες αερομεταφορών όπως η British Airways, η KLM, η Iberia, η Alitalia και η Scandinavian Airlines (SAS).

Τα δεδομένα που εισάγονται στο πρόβλημα crew rostering περιέχουν γενικές πληροφορίες για τα μέλη του πληρώματος, δραστηριότητες που πρέπει να γίνουν, κανόνες, κανονισμοί και δραστηριότητες για την δημιουργία προγραμμάτων. Κατά την παραγωγή εξατομικευμένων προγραμμάτων, τα προσωπικά αρχεία περιέχουν τα προσόντα, τις προκαθορισμένες δραστηριότητες και τις ημέρες διακοπών. Άλλα δεδομένα που λαμβάνονται υπόψη είναι οι ημέρες εκπαίδευσης του προσωπικού και κάποιες πιθανές εξαιρέσεις λόγω κανονισμών. Για τα μέλη του πληρώματος καμπίνας ένας ακόμα παράγοντας που λαμβάνεται υπόψη είναι η γλωσσική επάρκεια ιδίως για διεθνής πτήσεις.

Το **σχήμα 2.4** παρέχει μια γραφική αναπαράσταση των δεδομένων που πρέπει να ληφθούν υπόψη κατά την εφαρμογή του crew rostering.



Σχήμα 2.4: Δεδομένα του crew rostering

2.4.4.1 Τυπικοί κανόνες και κανονισμοί

Οι κανόνες και οι κανονισμοί εκφράζουν τις συνθήκες κάτω από τις οποίες ένα πρόγραμμα είναι νόμιμο. Ορισμένοι από αυτούς μπορεί να οφείλονται στην νομοθεσία και στις κυβερνητικές υπηρεσίες ενώ άλλες μπορεί να οφείλονται στην ίδια την αεροπορική εταιρία η ακόμα σε συμφωνίες της εταιρίας με συνδικαλιστικές οργανώσεις εργαζομένων.

ΚΕΦΑΛΑΙΟ 3

3.1 Μαθηματικές τεχνικές προγραμματισμού

Το πρόβλημα ταιριάσματος πληρώματος πρέπει να θεωρηθεί ως ένα μοντέλο Δυαδικού Ακέραιου Προγραμματισμού (Binary Integer Programming, BIP). Σε προβλήματα της πραγματικής ζωής, αυτά τα μοντέλα περιλαμβάνουν συνήθως μεγάλο αριθμό μεταβλητών. Τρεις είναι οι πιο γνωστές προσεγγίσεις για μεγάλα προβλήματα, η branch-and-bound, branch-and-cut & branch-and-price και η column generation.

3.1.1 Branch and bound

Η μέθοδος branch-and bound ξεκινά με την επίλυση του γραμμικού προβλήματος χαλάρωσης. Έπειτα, χρησιμοποιώντας τη λύση την οποία αποκομίζει, η οποία είναι πολύ πιθανό να μην είναι ακέραια, δημιουργούνται δύο υπό-προβλήματα προκειμένου να περιοριστεί το εύρος των λύσεων. Οι λύσεις από αυτά τα υπο-προβλήματα οριοθετούν τη λύση του αρχικού προβλήματος. Χρησιμοποιώντας συγκεκριμένους κανόνες η διαδικασία συνεχίζεται έως ότου επιτευχθεί μία άριστη ακέραια λύση. Σύμφωνα με τον Potter (2008) , οι διακλαδώσεις μπορούν να είναι είτε μεταβλητές, είτε περιορισμοί, οι οποίοι μπορούν να επηρεάσουν τόσο τη δυσκολία επίλυσης των υπο-προβλημάτων όσο και το πόσο καλά μπορεί η προσέγγιση να λειτουργεί. Επομένως, γίνεται εύκολα αντιληπτό ότι ο καθορισμός της σειράς των διακλαδώσεων είναι κρίσιμος. Η μέθοδος προτάθηκε αρχικά από τους Ailsa Land και Alison Doig, ενώ διεξήγαγε έρευνα στο London School of Economics το 1960 για διακριτό προγραμματισμό και έγινε το πιο συνηθισμένο εργαλείο για την επίλυση του NP προβλημάτων βελτιστοποίησης. Το όνομα "branch and bound" εμφανίστηκε για πρώτη φορά στο έργο του Little et al. για το πρόβλημα των μετακινούμενων πωλητών.

3.1.2 Branch and price

Η επιτυχής λύση ενός μεγάλου μεγέθους προβλήματος μικτού ακέραιου προγραμματισμού (MIP) απαιτεί συνθέσεις των οποίων οι γραμμικοί περιορισμοί χαλάρωσης (LP) δίνουν μια καλή προσέγγιση εφικτών λύσεων. Την τελευταία δεκαετία μεγάλη προσοχή έχει δοθεί στην προσέγγιση branch and cut για την επίλυση των προβλημάτων μικτού ακέραιου προγραμματισμού (MIP). Η βασική ιδέα της μεθόδου branch and cut είναι απλή. Μέρος των περιορισμών μένει έξω από το πρόβλημα γραμμικής χαλάρωσης, διότι υπάρχουν πάρα πολλοί ώστε να τους διαχειριστεί αποτελεσματικά. Αρκετοί περιορισμοί εξάλλου δεν θα επηρέαζαν καθόλου την βέλτιστη λύση του προβλήματος. Τότε αν η λύση από το πρόβλημα γραμμικής χαλάρωσης (LP relaxation) είναι μη εφικτή ένα υποπρόβλημα που ονομάζεται πρόβλημα διαχωρισμού επιλύεται για να εντοπίσει ποιοι περιορισμοί παραβιάζονται. Αν εντοπιστούν μια ή περισσότερες παραβιαζόμενες ανισότητες ορισμένες από αυτές εισάγονται στο πρόβλημα γραμμικής χαλάρωσης προκειμένου να λυθεί το πρόβλημα της μη εφικτής λύσης. Η μέθοδος branch and cut , η οποία είναι ένας συνδυασμός της μεθόδου διακλάδωσης και οριοθέτησης (branch and bound) με γραμμική χαλάρωση, επιτρέπει την διαδικασία αυτή να γίνεται σε όλους τους κλάδους του δέντρου της μεθόδου branch and bound.

Η μέθοδος branch and price δεν διαφέρει και πολύ από την μέθοδο branch and cut. Η διαφορά τους έγκειται στο γεγονός ότι στο πρόβλημα της γραμμική χαλάρωση αντί να περιορίζουμε τον αριθμό των ανισώσεων, περιορίζουμε τον αριθμό των μεταβλητών.

Στην branch and price μέθοδο, επειδή οι μεταβλητές που μπορεί να προκύψουν είναι πάρα πολλές, ένα μέρος από αυτές λαμβάνεται υπόψη και μετέχουν στο πρόβλημα γραμμικής χαλάρωσης. Αυτό συμβαίνει διότι, λόγω του μεγάλου αριθμού των μεταβλητών δεν μπορεί να γίνει αποτελεσματική διαχείριση όλων. Επίσης, πολλοί από αυτούς, μπορεί να μην επηρεάσουν και καθόλου την λύση, καθώς η τιμή τους μπορεί να είναι μηδέν. Στην συνέχεια για να γίνει ο έλεγχος εάν η λύση του προβλήματος γραμμικής χαλάρωσης είναι βέλτιστη, λύνεται ένα νέο υποπρόβλημα, το οποίο λέγεται column generation subproblem. Το υποπρόβλημα αυτό, χρησιμοποιεί τις δυικές τιμές (duals) της λύσης του προβλήματος της γραμμικής χαλάρωσης, προκειμένου να βρει μεταβλητές που μπορούν να εισαχθούν στο πρόβλημα. Εάν τέτοιες μεταβλητές βρεθούν, εισάγονται στο πρόβλημα γραμμικής χαλάρωσης, το οποίο και λύνεται ξανά, ώστε να δώσει καλύτερη βέλτιστη λύση. Η διακλάδωση στο δέντρο αρχίζει, όταν το column generation subproblem δεν μπορεί να βρει άλλες μεταβλητές για να εισαχθούν στο πρόβλημα και η λύση του προβλήματος γραμμικής χαλάρωσης δεν ικανοποιεί τις συνθήκες του αρχικού προβλήματος.

3.1.3 Column generation

Η μεθοδολογία column generation είναι πολύ αποτελεσματική για την επίλυση προβλημάτων με τεράστιο αριθμό μεταβλητών απόφασης, η οποία λειτουργεί εξετάζοντας ένα μικρό υποσύνολο αυτών. Όπως περιγράφεται εκτενώς από τον Shenoi (1994), ένα περιορισμένο κύριο πρόβλημα δημιουργείται από το αρχικό πρόβλημα αριστοποίησης του πληρώματος, χρησιμοποιώντας ένα συγκεκριμένο αριθμό μεταβλητών, που συνιστά μια εφικτή λύση. Αυτό το νέο υποπρόβλημα επιλύεται και υπολογίζονται οι εξαγόμενες δυικές τιμές και έπειτα σύμφωνα με συγκεκριμένους υπολογισμούς καθορίζεται αν θα προστεθούν περισσότερες στήλες, προκειμένου να επιτευχθεί η άριστη λύση του προβλήματος. Αυτή η διαδικασία επαναλαμβάνεται έως ότου δεν υπάρχουν άλλες στήλες, οι οποίες να μπορούν να προστεθούν. Ορόσημο για την μεθοδολογία column generation αποτελεί η έρευνα των Barnhart et al. στην οποία οι συγγραφείς συζητούν διάφορους τομείς εφαρμογής, εναλλακτικές διαμορφώσεις μοντέλων, καθώς και αρκετά υπολογιστικά θέματα επιδόσεων. Παρουσιάζουν επίσης των τρόπο σύνδεσης της μεθοδολογίας column generation με την branch and price για την επίλυση τεράστιων προβλημάτων. Η επιτυχείς εφαρμογές της μεθόδου column generation έχουν εφαρμοστεί σε διάφορες εφαρμογές σε πραγματικό κόσμο, όπως ο προγραμματισμός του πληρώματος, η δρομολόγηση των οχημάτων, ο προγραμματισμός θέσης κ.λπ.

3.2 Μαθηματική μοντελοποίηση

Η μέθοδος που θα ακολουθήσουμε για να λυθεί το πρόβλημα, όπως αναφέρθηκε και παραπάνω είναι η branch and price. Η μέθοδος αυτή αποτελείται από το Master problem και το column generation sub problem. Παρακάτω για καθένα από αυτά τα δύο προβλήματα αρχικά θα γίνεται μια γενική περιγραφή των εξισώσεων, οι οποίες τα περιγράφουν και στην συνέχεια θα γίνεται η μαθηματική μοντελοποίηση του προβλήματος, το οποίο διαπραγματεύεται η συγκεκριμένη εργασία.

3.2.1 Master problem

Για να γίνει πιο εύκολα κατανοητό, αρχικά θα γίνει παρουσίαση του γενικού προβλήματος που ακολουθεί η μέθοδος master problem. Η αντικειμενική συνάρτηση είναι:

$$\text{minimize} \quad Z_{MP} = \sum_{i \in S} c_i \chi_i \quad \textbf{(1)}$$

με περιορισμούς:

$$\sum_{i \in S} a_{if} \chi_i \geq b_f, \ \forall \ f \in F \quad \textbf{(2)}$$

$$\chi_i \geq 0, \qquad i \in S \quad \textbf{(3)}$$

Όπου c_i , α_{if}, b_f , x_i ανήκουν στο σύνολο των πραγματικών αριθμών R, για κάθε i∈S, όπου S={1,2,………,n},F={1,2,….m} και n,m είναι ο αριθμός των μεταβλητών του προβλήματος και ο αριθμός των περιορισμών αντίστοιχα. Η βέλτιστη λύση του προβλήματος αυτού συμβολίζεται Z^*_{MP}. Γίνεται η υπόθεση ότι ο αριθμός των μεταβλητών του προβλήματος είναι πολύ μεγαλύτερος από τον αριθμό των περιορισμών. Λόγω αυτού, μετατρέπουμε το πρόβλημα σε ένα μικρότερο πρόβλημα, λαμβάνοντας υπόψη λιγότερες μεταβλητές, δηλαδή έχουμε ένα υποσύνολο του S το

S1. Το νέο πρόβλημα είναι το restricted master problem (RMP) και διαμορφώνεται ως ακολούθως:

$$\text{minimize} \quad Z_{MP} = \sum_{i \in S1} c_i \chi_i \quad \mathbf{(4)}$$

με περιορισμούς:

$$\sum_{i \in S1} a_{if} \chi_i \geq b_f, \quad \forall f \in F \quad \mathbf{(5)}$$

$$\chi_i \geq 0, \quad \forall \; i \in S_1 \quad \mathbf{(6)}$$

Η βέλτιστη λύση στο RMP αποτελεί ένα πάνω όριο για το master problem(MP). Επίσης δεδομένου ότι ο αριθμός των περιορισμών είναι ίδιος και στα δύο προβλήματα, κάθε εφικτή δυική λύση του RMP είναι εφικτή και στο MP.

Παραπάνω παρουσιάστηκε η γενική περιγραφή της μεθόδου. Όπως είπαμε σκοπός του προβλήματος είναι η κάλυψη ενός συνόλου διαδρομών S, από ένα σύνολο ιπταμένων I, ικανοποιώντας ταυτόχρονα κάποια ποιοτικά πρότυπα.

Πιο συγκεκριμένα, κάθε διαδρομή έχει μια αφετηρία και έναν προορισμό. Ως διαδρομή ορίζεται ένα ταξίδι από ένα μέρος σε ένα άλλο. Ωστόσο αυτό το ταξίδι μπορεί να μην είναι απευθείας. Ενδιάμεσα μπορεί να έχει και άλλους προορισμούς, μέχρι να φτάσει στον τελικό του προορισμό. Η διάρκεια πτήσης κάθε διαδρομής διαφέρει και μπορεί να είναι και παραπάνω από μια ημέρα. Αυτό οφείλεται στο γεγονός ότι στους ενδιάμεσους προορισμούς, λόγω των πολλών ωρών που μπορεί να ταξιδεύει, θα χρειαστεί να διακόψει την διαδρομή για να ξεκουραστεί το πλήρωμα και να συνεχίσει μετά τον κατάλληλο χρόνο ξεκούρασης. Επίσης πολλές φορές το πλήρωμα χρειάζεται να διανυκτερεύσει σε έναν από τους ενδιάμεσους προορισμούς και να συνεχίσει την πτήση το επόμενο πρωί. Όλα αυτά συμβάλουν στην αύξηση της διάρκεια της ολικής διαδρομής. Το σύνολο των διαδρομών που καλύπτει ο κάθε πιλότος στην διάρκεια ενός μήνα, το ονομάζουμε πρόγραμμα του πιλότου.

Στην παρούσα εργασία ορίζεται ένα άνω όριο για τις συνολικές ώρες που μπορεί να πετάξει ένας πιλότος στις 100 ώρες το μήνα. Για κάθε πιλότο που ξεπεράσει αυτό το όριο υπάρχει ένα κόστος το HoursCost που ισούται με 100.000. Επίσης δηλώνονται οι επιθυμίες για τα μηνιαία ρεπό του κάθε πιλότου. Για κάθε ρεπό που δεν καλύπτεται έχουμε ένα κόστος το οποίο ονομάζουμε UncoverDayoff. Το UncoverDayoff το οποίο στην ουσία είναι το κόστος του penalty που αντιστοιχεί στην εταιρία για τη μη κάλυψη των ρεπό ισούται με 1.000 μονάδες για κάθε ρεπό που μένει ανικανοποίητο. Επίσης αν μια διαδρομή μείνει ακάλυπτη, για κάθε ημέρα που η αποστολή αυτή παραμένει ακάλυπτη ορίζεται ένα κόστος, το κόστος ακάλυπτης διαδρομής. Είναι πολύ σημαντικό η εταιρία να μπορεί να πραγματοποιήσει όλες τις διαδρομές που έχει προγραμματίσει για ένα χρονικό διάστημα (εδώ για διάρκεια ενός μήνα). Για κάθε διαδρομή, η οποία έχει την κατάλληλη ζήτηση, ώστε να προγραμματιστεί και δεν μπορεί τελικά να πραγματοποιηθεί, το κόστος για την εταιρία είναι τεράστιο. Στην συγκεκριμένη εργασία το κόστος την ακάλυπτης διαδρομής ορίζεται ως 1.000.000 μονάδες ανά ημέρα που η διαδρομή παραμένει ακάλυπτη. Τα κόστη αυτά ορίζονται ως παράμετροι του προβλήματος, όμως το κόστος ακάλυπτης διαδρομής, όπως παρατηρούμε, είναι συγκριτικά με το UncoverDayoff πολύ μεγαλύτερο, δεδομένου ότι βασικός στόχος του προβλήματος είναι η κάλυψη όσο το δυνατόν περισσότερων διαδρομών και δευτερεύον στόχος είναι η μείωση του κόστους, το οποίο αντιστοιχεί στην κάλυψη των απαιτούμενων ρεπό.

Επιπλέον, ένα σημαντικό κριτήριο το οποίο πρέπει να ικανοποιηθεί, αφορά την χρονική επικάλυψη των διαδρομών. Κατά την επίλυση του προβλήματος για τον ίδιο πιλότο δημιουργούνται κάποια προγράμματα, κάθε ένα από τα οποία περιέχει το δικό του συνδυασμό διαδρομών. Όλες οι διαδρομές του προγράμματος, που θα επιλεγεί για τον κάθε πιλότο, πρέπει οπωσδήποτε να καλυφθούν. Επομένως, το κάθε πρόγραμμα δεν μπορεί να περιέχει διαδρομές, οι οποίες είναι αλληλεπικαλυπτόμενες χρονικά. Δυο διαδρομές δεν θεωρούνται αλληλεπικαλυπτόμενες χρονικά, όταν η ημέρα επιστροφής της μίας είναι ίδια ή προηγούμενη ημέρα, από την ημέρα αναχώρησης της άλλης διαδρομής. Επίσης όσο αναφορά τους πιλότους, κάποιες ημερομηνίες μπορεί να μην είναι διαθέσιμοι για πτήση, λόγω άδειας, crew rest κ.α. Απόρροια αυτού είναι ότι οι διαδρομές, οι οποίες αντιστοιχούν στις συγκεκριμένες ημερομηνίες, δεν μπορούν να καλυφτούν από πιλότους , οι οποίοι δεν είναι διαθέσιμοι τις ημερομηνίες αυτές.

Παρακάτω θα παρουσιάσουμε την μαθηματική μοντελοποίηση του δικού μας προβλήματος.

Ορίζουμε ως :

<u>Σύνολα</u>

I : Το σύνολο των ιπταμένων.

S : Το σύνολο των διαδρομών.

R_i : Το σύνολο των προγραμμάτων για τον πιλότο i.

Παράμετροι του προβλήματος

f : Το κόστος για τις ακάλυπτες διαδρομές, το οποίο είναι το κόστος ακάλυπτης διαδρομής επί τη διάρκεια που μένει ακάλυπτη η διαδρομή σε ημέρες.

α_{ijs} : δυαδική παράμετρος, η οποία παίρνει την τιμή 1 όταν το πρόγραμμα j του πιλότου i καλύπτει την διαδρομή s, αλλιώς παίρνει την τιμή 0, $i \in I$, $j \in Ri$, $s \in S$.

Μεταβλητές απόφασης

xij : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1 όταν επιλέγεται το πρόγραμμα j του πιλότου i, και 0 διαφορετικά, $i \in I$, $j \in Ri$,

ys : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1 όταν η διαδρομή s παραμένει ακάλυπτη, και 0 διαφορετικά, $s \in S$.

Η αντικειμενική συνάρτηση προκύπτει ως εξής:

$$Z_{MP} = \sum_{s \in S} f y_s \quad \textbf{(7)}$$

με περιορισμούς:

$$\sum_{j \in Ri} \chi_{ij} = 1, \forall\, i \in I \quad \textbf{(8)}$$

$$\sum_{i \in I} \sum_{j \in S1} \alpha_{ijs}\chi_{ij} + y_s = 1, \quad \forall\, s \in S \quad \textbf{(9)}$$

$$\chi_{ij}\, y_s \text{ binary}, \ \forall\, i, j, s \quad \textbf{(10)}$$

Η εξίσωση (7) αποτελεί την αντικειμενική συνάρτηση και ελαχιστοποιεί το κόστος, το οποίο αποτελείται από το κόστος για τις ακάλυπτες διαδρομές. Ο περιορισμός (8) εξασφαλίζει ότι για κάθε πιλότο θα επιλεγεί μόνο ένα πρόγραμμα. Ο περιορισμός αυτός περιλαμβάνει και τα προγράμματα τα όποια είναι κενά, δηλαδή αυτά που δεν περιέχουν καμία διαδρομή. Ο περιορισμός (9) δηλώνει ότι κάθε διαδρομή s, είτε θα καλυφθεί από ένα ακριβώς πρόγραμμα, είτε θα μείνει ακάλυπτη και η μεταβλητή y_s

θα πάρει την τιμή 1. Τέλος ο περιορισμός (10) υποδηλώνει ότι οι μεταβλητές απόφασης είναι δυαδικές.

3.2.2 Column generation sub problem

Το column generation sub problem στοχεύει στην αναγνώριση των συνδυασμών προγραμμάτων-ιπταμένων με το μικρότερο reduce cost. Αν ο συνδυασμός αυτός είναι αρνητικός τότε υπάρχει δυνατότητα βελτίωσης της λύσης. Επομένως αν βρεθεί εισάγεται στο πρόβλημα, το οποίο λύνεται ξανά και μας δίνει νέα βέλτιστη και δυική λύση. Αν κανένας συνδυασμός δεν δίνει αρνητικό αποτέλεσμα αυτό σημαίνει ότι καμία άλλη διαδρομή δεν μπορεί να βελτιώσει την βέλτιστη λύση για τη χαλάρωση του master LP και η διαδικασία τερματίζεται.

Για να γίνει πιο εύκολα κατανοητό, αρχικά θα γίνει παρουσίαση του γενικού προβλήματος που ακολουθεί η μέθοδος column generation sub problem και στην συνέχεια θα παρουσιάσουμε την μαθηματική μοντελοποίηση του δικού μας προβλήματος.

Η εξίσωση που θα μας δώσει το *reduce cost* μπορεί να γραφτεί ως εξής:

$$C_{RC\,i} = c_i - \sum_{f \in F} a_{if}\pi_f \quad \textbf{(11)}$$

Όπου c_i είναι το κόστος της μεταβλητής i, a_{if} είναι δυική μεταβλητή, η οποία παίρνει τιμή 1 όταν η μεταβλητή i υπάρχει στον περιορισμό f και π_f είναι η δυική τιμή του περιορισμού f. Η μεταβλητή i που θα μας δώσει το μικρότερο C_{RC} θα εισαχθεί στο RMP ως νέα μεταβλητή. Το πρόβλημα θα λυθεί ξανά και η διαδικασία θα συνεχιστεί έως ότου να μην υπάρχει αρνητικό reduce cost, δηλαδή όταν θα ισχύει $C_{RC} \geq 0$, όπου και το πρόβλημα δεν θα μπορεί να βελτιστοποιηθεί παραπάνω. Η βέλτιστη λύση του RMP θα είναι και η βέλτιστη λύση του MP.

Ας περάσουμε τώρα στο δικό μας πρόβλημα.

Στο πρόβλημα, το οποίο διαπραγματευόμαστε στην συγκεκριμένη εργασία, το column generation sub-problem θα λυθεί με την βοήθεια ενός δικτύου N={V,A}. Πριν αναλύσουμε όμως τον τρόπο με τον οποίο θα χρησιμοποιηθεί το δίκτυο, θα πρέπει να ορίσουμε κάποιες παραμέτρους. Έστω sd_s, ld_s είναι η ημερομηνία αναχώρησης και η ημερομηνία προσγείωσης της διαδρομής s. Το δίκτυο περιλαμβάνει έναν κόμβο για κάθε πιλότο, τον οποίο ονομάζουμε κόμβο 0, έναν κόμβο για κάθε διαδρομή, καθώς και έναν τελικό κόμβο, ο οποίος ονομάζεται fictitious και θα συμβολίζεται ως κόμβος Ε, όπου εκεί τερματίζει κάθε πρόγραμμα. Ο κόμβος fictitious δεν υπάρχει στην πραγματικότητα, δηλαδή είναι φανταστικός κόμβος και τον χρησιμοποιούμε για δική μας ευκολία. Αφού ο κόμβος 0 είναι ξεχωριστός για κάθε πιλότο, καταλαβαίνουμε ότι έχουμε τόσα δίκτυα όσοι είναι και οι πιλότοι. Μεταξύ των κόμβων του δικτύου καθορίσαμε δυο διαφορετικούς συνδυασμούς. Ο πρώτος συνδυασμός περιλαμβάνει το ζευγάρι κόμβων πιλότου-διαδρομής (I-S), ενώ ο δεύτερος το ζευγάρι κόμβων διαδρομής-διαδρομής (S-S). Για τον συνδυασμό I-S, ορίζουμε ένα σύνολο F_i , το οποίο περιλαμβάνει τις διαδρομές s, που είναι διαθέσιμες να κάνει ο πιλότος i, δηλαδή αυτές που μπορούν να είναι οι επόμενοι συμβατοί κόμβοι, με τους οποίους θα συνδυαστεί ο κόμβος του πιλότου. Επίσης ορίζουμε και ένα σύνολο B_s, το οποίο περιλαμβάνει τους πιλότους i, οι οποίοι μπορούν να αποτελέσουν τον προηγούμενο συμβατό κόμβο της διαδρομής s. Ο κόμβος ενός πιλότου θεωρείται ως προηγούμενος συμβατός κόμβος με τον κόμβο της διαδρομής s, όταν ο κόμβος της διαδρομής s μπορεί να αποτελέσει τον επόμενο συμβατό κόμβο του πιλότου i. Για τον συνδυασμό S-S, ορίζουμε δύο σύνολα το P_s και το N_s. Για κάθε διαδρομή s, ορίζουμε το σύνολο N_s, το οποίο περιλαμβάνει όλες τις διαθέσιμες διαδρομές, που μπορούν να αποτελέσουν τις επόμενες συμβατές διαδρομές για να συνδυαστούν με την διαδρομή s. Μια διαδρομή t μπορεί να αποτελέσει την επόμενη συμβατή διαδρομή για να συνδυαστεί με την διαδρομή s, όταν $ld_s \leq sd_t$. Επιπλέον, για κάθε διαδρομή s ορίζουμε το σύνολο P_s, το οποίο αποτελείται από τις διαδρομές, οι οποίες μπορούν να αποτελέσουν προηγούμενους συμβατούς κόμβους της διαδρομής s. Μια διαδρομή t μπορεί να αποτελέσει προηγούμενο συμβατό κόμβο, του κόμβου της διαδρομής s, όταν η διαδρομή s είναι επόμενη συμβατή διαδρομή για να συνδυαστεί με τον κόμβο της διαδρομής t.

Ο στόχος του υποπροβλήματος column generation είναι να προσδιοριστεί το μακρύτερο μονοπάτι διαδρομών σε κάθε δίκτυο που να ξεκινάει από έναν κόμβο πιλότου, να επισκέπτεται τουλάχιστον έναν κόμβο διαδρομής και να τελειώνει στον τερματικό κόμβο. Λύνοντας το master problem παίρνουμε τις τιμές της δυικής λύσης. Έστω c_i, d_s οι δυικές τιμές των περιορισμών των πιλότων και των διαδρομών αντίστοιχα. Τα σύνολα F_i, B_s, P_s και N_s έχουν οριστεί παραπάνω. Παρακάτω παρουσιάζουμε την μαθηματική μοντελοποίηση του προβλήματος column generation για το πρόβλημά μας.

<u>Μεταβλητές απόφασης</u>

z_i : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται αφορά τον πιλότο i, αλλιώς παίρνει την τιμή 0, $i \in I$.

w_{is} : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει απευθείας σύνδεση του κόμβου 0 του πιλότου i με τον κόμβο της διαδρομής s, αλλιώς παίρνει την τιμή 0, $i \in I$, $s \in F_i$.

w_{st} : δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει απευθείας σύνδεση από τον κόμβο της διαδρομής s στον κόμβο της διαδρομής t, αλλιώς παίρνει την τιμή 0. Ο κόμβος t μπορεί να είναι είτε κόμβος μιας άλλης διαδρομής είτε ο τελικός κόμβος E (fictitious), $s \in S$, $t \in N_s \cup \{E\}$.

u_s: δυαδική μεταβλητή απόφασης, η οποία παίρνει την τιμή 1, όταν το πρόγραμμα που δημιουργείται περιλαμβάνει την διαδρομή s, αλλιώς παίρνει την τιμή 0, s ∈ S.

$$\text{minimize RC} = b - \sum_{i \in I} c_i z_i - \sum_{s \in S} d_s u_s \quad \textbf{(12)}$$

με περιορισμούς,

$$\sum_{i \in I} z_i = 1 \quad \textbf{(13)}$$

$$\sum_{s \in Fi} w_{is} = z_i \,, \quad \forall\ i \in I \quad \textbf{(14)}$$

$$\sum_{i \in Bs} w_{is} + \sum_{r \in Ps} w_{rs} = \sum_{t \in Ns \cup \{E\}} w_{st} \,, \quad \forall\ s \in S \quad \textbf{(15)}$$

$$\sum_{i \in Bs} w_{is} + \sum_{r \in Ps} w_{rs} = u_s \,, \quad \forall\ s \in S \quad \textbf{(16)}$$

$$z_i, w_{is}, w_{rs}, w_{st}, u_s \text{ δυικές}, \ \forall\ i, s, r, t. \quad \textbf{(17)}$$

Η εξίσωση (12) αποτελεί και την αντικειμενική συνάρτηση η οποία χρησιμοποιεί τις δυικές τιμές για να βρει το μικρότερο reduce cost. Οι περιορισμοί (13) εξασφαλίζουν ότι κάθε πρόγραμμα θα ανήκει σε έναν μόνο ιπτάμενο. Οι περιορισμοί (14) δηλώνουν ότι το πρόγραμμα που θα βρεθεί, πρέπει να περιλαμβάνει μια απευθείας σύνδεση μεταξύ του κόμβου του πιλότου με τον κόμβο μιας από τις διαδρομές που μπορούν να είναι οι επόμενες συμβατές. Το σύνολο των περιορισμών (15) δηλώνουν ότι σε έναν κόμβο θα εισέρχεται το πρόβλημα είτε από έναν κόμβο πιλότου, είτε μιας άλλης διαδρομής. Επίσης, από τον κόμβο αυτό θα εξέρχεται είτε προς έναν άλλον κόμβο διαδρομής, είτε προς τον τελικό κόμβο. Οι περιορισμοί (16) εξασφαλίζουν ότι μια διαδρομή καλύπτεται από ένα πρόγραμμα, αν και μόνο αν η διαδρομή αυτή έχει συνδυαστεί με προηγούμενες, δηλαδή μόνο εάν υπάρχει κάποια είσοδος σε αυτή την διαδρομή. Τέλος, οι περιορισμοί (17) δηλώνουν το είδος των μεταβλητών του προβλήματος.

Ωστόσο, αξίζει να σημειωθεί, ότι κατά την επίλυση του όλου προβλήματος με το πρόγραμμα IBM ILOG CPLEX Optimization Studio, το πρόβλημα column generation, το οποίο παρουσιάσαμε παραπάνω, δεν επιλύθηκε ως ξεχωριστό πρόβλημα. Για την επίλυσή του εφαρμόστηκε ο αλγόριθμο συντομότερης διαδρομής για μη κυκλικά δίκτυα (acyclic shortest path algorithm, ASPA).

3.3.3 Branching

Το πρόβλημα της γραμμικής χαλάρωσης δεν εγγυάται ότι η λύση που θα βρεθεί θα είναι ακέραια. Για τον λόγω αυτό ο αλγόριθμος προβλέπει την διακλάδωση (branching) του προβλήματος σε άλλα υπο-προβλήματα προκειμένου να συνεχιστεί η έρευνα για την εύρεση βέλτιστης ακέραιης λύσης. Για την εξάλειψη αυτών των δεκαδικών τιμών θα προστεθούν επιπλέον περιορισμοί στο πρόβλημα. Για να δημιουργηθούν αυτά τα υπο-προβλήματα είναι αναγκαία η επιλογή μιας μεταβλητής απόφασης με δεκαδική τιμή. Αν βρεθεί μεταβλητή απόφασης με δεκαδική τιμή τότε το πρόβλημα χωρίζεται σε δύο υπο-προβλήματα, όπου στο ένα η τιμή αυτής της δεκαδικής τιμής τίθεται ίση με μηδέν και στο άλλο ίση με ένα. Έχει αποδειχθεί ότι αν στο πρόβλημα γραμμικής χαλάρωσης βρεθεί μία η και περισσότερες μεταβλητές με δεκαδική τιμή, τότε υπάρχει τουλάχιστον ένας περιορισμός για τους ιπταμένους και ένας για τις διαδρομές, τέτοιοι ώστε το άθροισμα των μεταβλητών, οι οποίες εμφανίζονται και στους δύο περιορισμούς, να προκύπτει δεκαδική τιμή. Αν στο πρόβλημα γραμμικής χαλάρωσης υπάρχουν παραπάνω από μια μεταβλητές με δεκαδική τιμή τότε εμείς ορίσαμε να επιλέγεται η μεταβλητή με την μεγαλύτερη δεκαδική τιμή. Όταν επιλεγεί αυτή η μεταβλητή απόφασης τότε ξεκινάει η μεθοδολογία της διακλάδωσης. Στο αριστερό υπο-πρόβλημα η τιμή τίθεται ίση με το μηδέν ενώ στο δεξί ίση με την μονάδα. Αυτό πρακτικά σημαίνει ότι είτε η τιμή δεν θα επηρεάσει καθόλου την λύση (όταν τίθεται ίση με μηδέν) ή θα επιλεγεί ο συνδυασμός διαδρομών που αυτή αντιπροσωπεύει (όταν τίθεται ίση με μονάδα).

Στο αριστερό υπο-πρόβλημα η μεταβλητή, η οποία έχει επιλεχθεί, θέτεται ίση με το μηδέν και στην ουσία διαγράφεται από το πρόβλημα. Ένας κατάλληλος περιορισμός εισάγεται στο column generation sub-problem, ώστε η μεταβλητή αυτή να μην δημιουργηθεί ξανά. Λύνουμε ξανά το πρόβλημα και οι νέες μεταβλητές, οι οποίες θα παραχθούν, θα διαφέρουν από αυτήν που διαγράφτηκε, το οποίο βοηθάει στην εύρεση καλύτερης λύσης.

Στο δεξί υπο-πρόβλημα η μεταβλητή, η οποία έχει επιλεχθεί, θέτεται ίση με την μονάδα. Είναι σημαντικό να επισημάνουμε ότι αυτό δεν γίνεται με την προσθήκη νέων περιορισμών άλλα με την μορφοποίηση των ήδη υπαρχόντων. Μόλις η μεταβλητή τεθεί ίση με την μονάδα, διαγράφεται από όλους τους άλλους περιορισμούς που εμφανιζόταν και κατάλληλοι περιορισμοί εισάγονται στο column generation sub-problem έτσι ώστε οι μεταβλητές που διαγράφηκαν να μην δημιουργηθούν ξανά. Στην συνέχεια λύνουμε ξανά το πρόβλημα, το οποίο προκύπτει έχοντας κάνει την παραπάνω διαδικασία.

Έχοντας τις λύσεις από τα δύο υπο-προβλήματα, συγκρίνουμε και βρίσκουμε πιο υπο-πρόβλημα μας δίνει την καλύτερη λύση. Το υποπρόβλημα αυτό γίνεται το νέο πρόβλημα και επαναλαμβάνεται η ίδια διαδικασία έως ότου να μην υπάρχει καμία μεταβλητή απόφασης με δεκαδική τιμή.

ΚΕΦΑΛΑΙΟ 4

4.1 Μεθοδολογία επίλυσης

Στο κεφάλαιο 3 έγινε η παρουσίαση της μαθηματικής μοντελοποίησης της κάθε μεθόδου. Το πρόβλημα το οποίο εμείς διαπραγματευόμαστε έχει και κάποιες παραμέτρους που πρέπει να λάβουμε υπόψη για την σωστή επίλυση του προβλήματος, οι οποίες δεν ήταν απαραίτητο να επισημανθούν στην παρουσίαση της μαθηματικής μοντελοποίησης. Για τον λόγο αυτό στο παρών κεφάλαιο θα γίνει παρουσίαση της μεθοδολογίας που θα ακολουθηθεί στο συγκεκριμένο πρόβλημα για την κάθε μέθοδο ξεχωριστά καθώς και ένα αριθμητικό παράδειγμα.

4.2 Αρχικοποίηση του RMP

Για να ξεκινήσει η μέθοδος column generation είναι απαραίτητο να υπάρχει μια αρχική εφικτή λύση. Η λύση αυτή θα προκύψει από την επίλυση του RMP. Ως αρχική εφικτή λύση εμείς θα χρησιμοποιήσουμε την λύση που προκύπτει αν θέσουμε στην αντικειμενική συνάρτηση όλες τις δυικές μεταβλητές ίσες με την μονάδα. Αυτό πρακτικά σημαίνει ότι καμία πτήση δεν θα ανατεθεί σε κανέναν πιλότο (κενά προγράμματα) και άρα καμία πτήση δεν θα πραγματοποιηθεί, με αποτέλεσμα να έχουμε το μέγιστο δυνατό κόστος. Όπως γίνεται αντιληπτό αναφερόμαστε στις ακάλυπτες πτήσης και όχι στα μη ικανοποιημένα ρεπό των πιλότων, γιατί όπως ήδη έχουμε αναφέρει το κόστος των ακάλυπτων πτήσεων συγκριτικά με των μη ικανοποιημένων ρεπό είναι πολύ μεγαλύτερο. Κάνοντας αυτήν την υπόθεση, των κενών προγραμμάτων δημιουργείται η αρχική εφικτή λύση, η οποία έχει το μέγιστο κόστος. Η αρχική εφικτή λύση αποτελεί το πάνω όριο του προβλήματος καθώς καμία άλλη αλληλουχία τιμών δεν θα μας έδινε χειρότερη λύση (μεγαλύτερο κόστος).

4.3 Επιλογή του Paths

Ο αριθμός των μεταβλητών που εισάγονται ανά επανάληψη παίζει πολύ σημαντικό ρόλο στην τελική βέλτιστη λύση. Αν ο αριθμός των μεταβλητών που εισάγονται ανά επανάληψη είναι ένα (paths=1) και η μεταβλητή x έχει καλύτερο reduce cost από την μεταβλητή y σε μια επανάληψη τότε θα επιλεγεί η μεταβλητή x. Ωστόσο στο συνολικό πρόβλημα η μεταβλητή y μπορεί να έχει καλύτερο reduce cost από την μεταβλητή x. Έτσι η τελική βέλτιστη λύση θα ήταν χειρότερη, αν το Paths είχε επιλεγεί να είναι ίσο με την μονάδα από ότι αν ήταν μεγαλύτερο. Για τον λόγο αυτό είναι προτιμότερο να εισάγονται παραπάνω από μια μεταβλητές στο πρόβλημα ώστε να του δίνεται η δυνατότητα να βελτιώνει όσο περισσότερο είναι εφικτό την τελική βέλτιστη λύση. Βέβαια δεν μπορούμε να αυξήσουμε κατά πολύ την τιμή του paths γιατί ο χρόνος επίλυσης θα αυξανόταν επίσης. Για κάθε πρόβλημα που έχουμε να επιλύσουμε, επιλέγουμε τιμή για το paths ανάλογα με τις ανάγκες του εκάστοτε προβλήματος.

4.4 Επίλυση του column generation sub-problem

Για να γίνει κατανοητή η διαδικασία του column generation sub-problem θα ορίσουμε παρακάτω κάποιες μεταβλητές. Οι μεταβλητές αυτές είναι οι ετικέτες των κόμβων και είναι οι εξής:

-Reduce Cost, Previous, Previous Path, Flight Hours, Sum Duals και Sum Days off.

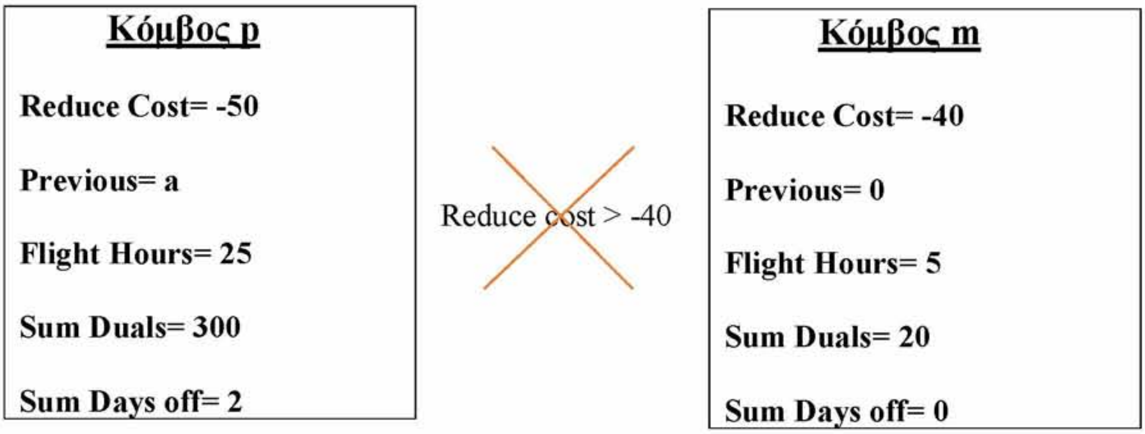
Το Reduce Cost είναι αυτό που καθορίζει εάν δυο εφικτοί κόμβοι θα συνδυαστούν ή όχι και ορίζεται ως η απόσταση μεταξύ των κόμβων. Το Previous, κάθε ετικέτας του κόμβου, δείχνει τον προηγούμενο κόμβο με τον οποίο συνδυάζεται ο υπό εξέταση κόμβος. Το Previous Path χρησιμοποιείται στην περίπτωση που στο πρόβλημα εισάγεται παραπάνω από μια μεταβλητές σε κάθε επανάληψη, για κάθε πιλότο. Όσα είναι τα Path, τόσες είναι και οι μεταβλητές που εισάγονται. Το Previous Path δείχνει από ποιο Path, του προηγούμενου κόμβου με τον οποίο έχει συνδυαστεί, έρχεται η συντομότερη διαδρομή για να συνδυαστεί με το υπό εξέταση Path του κόμβου αυτού. Το Flight hours κάθε κόμβου περιέχει τις συνολικές ώρες πτήσης των διαδρομών που έχουν συνδυαστεί μέχρι και τον κόμβο αυτό, στο πρόγραμμα που εξετάζεται. Το Sum duals του κάθε κόμβου περιλαμβάνει το άθροισμα των δυικών τιμών των κόμβων που έχουν συνδυαστεί μέχρι και τον κόμβο αυτό. Τέλος το Sum Days off περιλαμβάνει το άθροισμα των ρεπό που έχουν καλυφθεί μέχρι και τον κόμβο αυτό, στο πρόγραμμα που εξετάζεται. Κάθε φορά που μια νέα διαδρομή, δηλαδή ένας νέος κόμβος, συνδέεται στο πρόγραμμα, δηλαδή στο shortest path, οι παραπάνω τιμές του κόμβου αυτού αναθεωρούνται. Οι αρχικές τιμές για τις ετικέτες των κόμβων θα γίνει δημιουργώντας ένα shortest path, το οποίο ξεκινάει από τον κόμβο του πιλότου (κόμβος 0) πάει στον κόμβο της διαδρομής και καταλήγει στον κόμβο fictitious. Βέβαια αυτό γίνεται μόνο για τους κόμβους που είναι συμβατοί να συνδυαστούν με τον πιλότο. Αν για παράδειγμα ο κόμβος j΄΄ δεν είναι συμβατός με τον πιλότο i΄ τον οποίο εξετάζουμε, τότε δεν θα αρχικοποιηθεί, αφού δεν μπορεί να συμμετέχει στο πρόγραμμα του συγκεκριμένου πιλότου i΄. Για την επίλυση του Column generation sub-problem δεν θα χρησιμοποιηθεί το λογισμικό IBM ILOG CPLEX 12.5.1. Λόγω του ότι τα δίκτυα που έχουμε είναι μη κυκλικά, για την επίλυση αυτού του προβλήματος θα χρησιμοποιήσουμε τον αλγόριθμο ASPA. Παρακάτω θα παρουσιάσουμε ένα παράδειγμα για να γίνουν πιο κατανοητά τα όσα αναφέραμε νωρίτερα. Έστω ότι έχουμε δύο κόμβους τον κόμβο p και τον κόμβο m, οι οποίοι είναι εφικτό να συνδυαστούν. Έστω ότι ο κόμβος p έχει ήδη συνδυαστεί με άλλους κόμβους και έχει ένα shortest path, ενώ ο κόμβος m δεν έχει συνδυαστεί ακόμη με κανέναν κόμβο και θέλουμε να ελέγξουμε αν ο συνδυασμός του κόμβου p με τον m θα βελτιώσει το shortest path. Στο συγκεκριμένο παράδειγμα θα εισάγουμε μια μεταβλητή ανά επανάληψη, οπότε το Previous Path δεν θα το χρησιμοποιήσουμε. Έστω το shortest path που φθάνει μέχρι και τον κόμβο p έχει ένα reduce cost -50, συνολικό αριθμό ωρών πτήσης 25,συνολικό αριθμό ικανοποιημένων ρεπό 2 και με άθροισμα όλων των δυικών τιμών των κόμβων που έχουν συνδυαστεί, σε αυτό το shortest path, 300. Το Previous μας δείχνει ότι στο πρόγραμμα που δημιουργείται, αμέσως προηγούμενος κόμβος του p είναι ο κόμβος a. Οι τιμές για τον κόμβο m προκύπτουν από την αρχικοποίηση του κόμβου και έστω ότι είναι οι εξής. Το reduce cost είναι -40, οι ώρες πτήσης είναι 5, οι οποίες είναι στην ουσία οι ώρες της διαδρομής, καθώς όπως έχουμε πει και παραπάνω για την αρχικοποίηση δημιουργείται ένα shortest path που ξεκινάει από τον κόμβο 0 του πιλότου, πηγαίνει στον κόμβο της διαδρομής και καταλήγει στον fictitious. Το άθροισμα των δυικών τιμών είναι 20 και είναι το άθροισμα της δυικής τιμής του πιλότου και της διαδρομής. Το Previous είναι το 0, δηλαδή ο κόμβος του πιλότου και τέλος το άθροισμα των ρεπό που έχουν καλυφθεί 0.

Κόμβος p	**Κόμβος m**
Reduce Cost= -50	**Reduce Cost= -40**
Previous= a	**Previous= 0**
Flight Hours= 25	**Flight Hours= 5**
Sum Duals= 300	**Sum Duals= 20**
Sum Days off= 2	**Sum Days off= 0**

Πίνακας 4.1: Τιμές της ετικέτας των κόμβων πριν την εξέταση για συνδυασμό.

Το κριτήριο για την επιλογή του συνδυασμού, των δυο κόμβων θα προκύψει από το reduce cost. Το reduce cost του κόμβου m είναι -40, αν μετά τον συνδυασμό με τον κόμβο p το reduce cost του κόμβου m γίνει μικρότερο του -40, τότε ο κόμβος m θα γίνει ο αμέσως επόμενος κόμβος του κόμβου p. Αν το reduce cost του κόμβου m μετά τον συνδυασμό είναι μεγαλύτερο του -40, τότε ο κόμβος m δεν θα γίνει ο αμέσως επόμενος κόμβος του κόμβου p και οι τιμές της ετικέτας του κόμβου m δεν θα αλλάξουν.

Κόμβος p		**Κόμβος m**
Reduce Cost= -50		**Reduce Cost= -35**
Previous= a	Reduce cost< -40	**Previous= p**
Flight Hours= 25		**Flight Hours= 30**
Sum Duals= 300		**Sum Duals= 315**
Sum Days off= 2		**Sum Days off= 2**

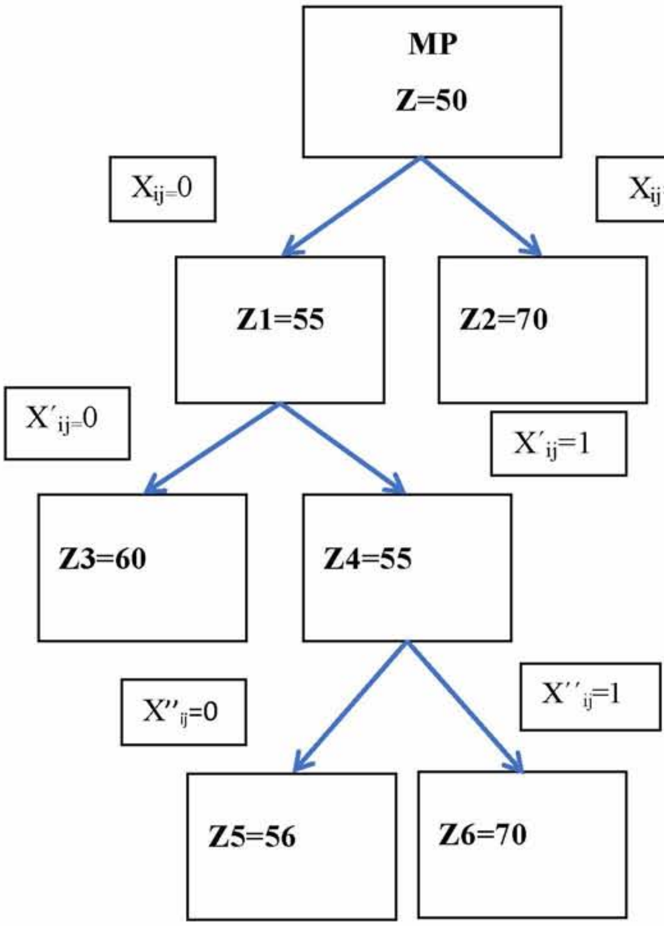


Πίνακας 4.2: Τιμές κόμβου m μετά τον έλεγχο.

4.5 Εφαρμογή της μεθόδου branching

Όπως ήδη έχουμε αναφέρει στο κεφάλαιο 3 μετά το τέλος της μεθόδου column generation sub-problem ακολουθεί η εφαρμογή της μεθόδου branching. Η μέθοδος branching εφαρμόζεται για την εξάλειψη των δεκαδικών τιμών με κύριο πλεονέκτημα, ότι παρόλο που αποτελεί ξεχωριστό πρόβλημα δεν προστίθενται νέοι περιορισμοί. Έτσι μόλις βρεθεί τιμή με δεκαδική τιμή ξεκινάει η διαδικασία της διακλάδωσης. Στο ένα πρόβλημα η τιμή με την μεγαλύτερη δεκαδική τιμή τίθεται ίση με μηδέν ενώ στο άλλο ίση με την μονάδα. Όποιος κλάδος δώσει καλύτερη τιμή της αντικειμενικής συνάρτησης αυτός είναι και ο κλάδος τον οποίο ακολουθούμε. Αξίζει να σημειωθεί ότι σε κανέναν κλάδο δεν μπορούμε να βρούμε καλύτερη τιμή αντικειμενικής συνάρτησης από αυτήν που μας έχει ήδη δώσει το RMP, στην καλύτερη περίπτωση η τιμή θα είναι η ίδια.

Το **σχήμα 4.3** θα μας βοηθήσει να κατανοήσουμε τα όσα αναφέρθηκαν παραπάνω.



Σχήμα 4.1 :Δέντρο διακλάδωσης

Η διαδικασία της διακλάδωσης συνεχίζεται μέχρις ότου να μην υπάρχει καμία δεκαδική τιμή.

4.6 Αριθμητικό παράδειγμα

Θα αναλύσουμε βήμα προς βήμα την επίλυση ενός αριθμητικού παραδείγματος N=10 πιλότων, R=100 διαδρομών και paths=3. Το κόστος κάθε διαδρομής που μένει ακάλυπτη το ορίζουμε 1.000.000 μονάδες, το κόστος για κάθε ρεπό που δεν ικανοποιείται το ορίζουμε 1.000 μονάδες και τέλος το κόστος για τις ώρες πτήσης κάθε πιλότου που υπερβαίνουν τις 100 ώρες το ορίζουμε 100.000 μονάδες. Η διάρκεια, η ημέρα αναχώρησης, η ώρα άφιξης και προσγείωσης της κάθε πτήσης, η συμβατότητα των διαδρομών και των πιλότων για κάθε διαδρομή καθώς και οι επιθυμίες για τις μέρες ανάπαυσης, προκύπτουν τυχαία με μια γεννήτρια τυχαίων αριθμών. Επομένως το πρόβλημα αυτό δεν αντιστοιχεί σε ένα πραγματικό πρόβλημα μιας πραγματικής αεροπορικής εταιρίας. Όλα αυτά φαίνονται στην εικόνα 4.1.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <ilcplex/ilocplex.h>
```

```c
#define N 10//number of crew members
#define R 100// number of route
#define paths 3 // number of paths
#define UndercoverCost 1000000 // the cost for not covering a route
#define UncoverDayoff 1000 // the cost for every dayoff which not cover
#define HoursCost 100000 // the cost for the extra hours
```

Εικόνα 4.1: Στοιχεία του προβλήματος.

Έπειτα πρέπει να οριστεί το RMP. Στις εικόνες 4.2 και 4.3 φαίνονται οι συναρτήσεις που αναπτύχθηκαν για την δημιουργία του RMP ενώ στην εικόνα 4.4 φαίνεται το πρόβλημα με την μορφή εξισώσεων.

```
int initial_variables(CPXENVptr Env, CPXLPptr Lp)
{
    int i, status;

    MemCType(N+R);
    MemForNewCol(N+R);
    for (i=0; i<=R-1; i++){
        SetBoundObj(i, 0 , CPX_INFBOUND, UndercoverCost*DUR[i]);
    }
    for (i=0; i<=N-1; i++){
        SetBoundObj(R+i, 0 , CPX_INFBOUND, 0);
    }
    status = NewCols(  Env,  Lp, R+N, 'C');
    if (status != 0){
       print_error("NewCols");
       return -1;
    }
    FreeMemCType();
    FreeMemForNewCol();
    return 0;
}
```

Εικόνα 4.2: Δημιουργία μεταβλητών του RMP

```
int initial_constrains(CPXENVptr Env, CPXLPptr Lp)
{
    int i, status , count, nzcnt;

    rmatbeg = (int*) malloc((R+N)*sizeof(int));
    MemForIndVal(N+R);
    rmatbeg[0]=0;
    for (i=0; i<=N-1; i++){
        count=0;
        nzcnt=0;
        SetMatindMatval( count,i+R, 1);
        (nzcnt++);
        status=AddRows(NULL, Env, Lp, nzcnt, 1, 'E');
        if (status != 0){
            print_error("AddRows");
        return -1;
        }
        (rmatbeg[i+1]=0);
    }
    for (i=0; i<=R-1; i++){
        count=0;
        nzcnt=0;
        SetMatindMatval( count, i, 1);
        (nzcnt++);
        status=AddRows(NULL, Env, Lp, nzcnt, 1, 'E');
        if (status != 0){
            print_error("AddRows1");
            return -1;
        }
        (rmatbeg[i+1]=0);
    }
    FreeMemForIndVal();
    status = CPXwriteprob(Env, Lp, "initmaster.lp", NULL);
    return 0;
}
```

Εικόνα 4.3: Δημιουργία περιορισμών του RMP

```
Minimize
 obj: 2000000 x1 + 1000000 x2 + 1000000 x3 + 1000000 x4 + 2000000 x5
      + 2000000 x6 + 2000000 x7 + 1000000 x8 + 2000000 x9 + 1000000 x10
      + 1000000 x11 + 2000000 x12 + 1000000 x13 + 2000000 x14 + 1000000 x15
      + 1000000 x16 + 1000000 x17 + 1000000 x18 + 1000000 x19 + 1000000 x20
      + 1000000 x21 + 2000000 x22 + 1000000 x23 + 2000000 x24 + 1000000 x25
      + 2000000 x26 + 1000000 x27 + 1000000 x28 + 1000000 x29 + 1000000 x30
      + 2000000 x31 + 2000000 x32 + 1000000 x33 + 2000000 x34 + 2000000 x35
      + 2000000 x36 + 1000000 x37 + 2000000 x38 + 1000000 x39 + 2000000 x40
      + 1000000 x41 + 1000000 x42 + 2000000 x43 + 2000000 x44 + 2000000 x45
      + 2000000 x46 + 2000000 x47 + 1000000 x48 + 2000000 x49 + 2000000 x50
      + 2000000 x51 + 2000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
      + 1000000 x56 + 1000000 x57 + 2000000 x58 + 1000000 x59 + 1000000 x60
      + 1000000 x61 + 2000000 x62 + 1000000 x63 + 1000000 x64 + 1000000 x65
      + 1000000 x66 + 2000000 x67 + 1000000 x68 + 1000000 x69 + 1000000 x70
      + 2000000 x71 + 1000000 x72 + 1000000 x73 + 1000000 x74 + 2000000 x75
      + 1000000 x76 + 2000000 x77 + 1000000 x78 + 2000000 x79 + 2000000 x80
      + 2000000 x81 + 1000000 x82 + 2000000 x83 + 2000000 x84 + 1000000 x85
      + 1000000 x86 + 1000000 x87 + 1000000 x88 + 2000000 x89 + 1000000 x90
      + 1000000 x91 + 2000000 x92 + 2000000 x93 + 1000000 x94 + 1000000 x95
      + 2000000 x96 + 1000000 x97 + 1000000 x98 + 1000000 x99 + 2000000 x100

Subject To
 c1:    x101  = 1
 c2:    x102  = 1
 c3:    x103  = 1
 c4:    x104  = 1
 c5:    x105  = 1
 c6:    x106  = 1
 c7:    x107  = 1
 c8:    x108  = 1
 c9:    x109  = 1
 c10:   x110  = 1
 c11:   x1  = 1
 c12:   x2  = 1
 c13:   x3  = 1
 c14:   x4  = 1
 c15:   x5  = 1
 c16:   x6  = 1
 c17:   x7  = 1
 c18:   x8  = 1
 c19:   x9  = 1
 c20:   x10  = 1
 c21:   x11  = 1
 c22:   x12  = 1
 c23:   x13  = 1
 c24:   x14  = 1
 c25:   x15  = 1
 c26:   x16  = 1
 c27:   x17  = 1
 c28:   x18  = 1
 c29:   x19  = 1
 c30:   x20  = 1
 c31:   x21  = 1
 c32:   x22  = 1
 c33:   x23  = 1
 c34:   x24  = 1
 c35:   x25  = 1
 c36:   x26  = 1
 c37:   x27  = 1
 c38:   x28  = 1
 c39:   x29  = 1
 c40:   x30  = 1
 c41:   x31  = 1
 c42:   x32  = 1
```

```
c43:  x33  = 1
c44:  x34  = 1
c45:  x35  = 1
c46:  x36  = 1
c47:  x37  = 1
c48:  x38  = 1
c49:  x39  = 1
c50:  x40  = 1
c51:  x41  = 1
c52:  x42  = 1
c53:  x43  = 1
c54:  x44  = 1
c55:  x45  = 1
c56:  x46  = 1
c57:  x47  = 1
c58:  x48  = 1
c59:  x49  = 1
c60:  x50  = 1
c61:  x51  = 1
c62:  x52  = 1
c63:  x53  = 1
c64:  x54  = 1
c65:  x55  = 1
c66:  x56  = 1
c67:  x57  = 1
c68:  x58  = 1
c69:  x59  = 1
c70:  x60  = 1
c71:  x61  = 1
c72:  x62  = 1
c73:  x63  = 1
c74:  x64  = 1
c75:  x65  = 1
c76:  x66  = 1
c77:  x67  = 1
c78:  x68  = 1
c79:  x69  = 1
c80:  x70  = 1
c81:  x71  = 1
c82:  x72  = 1
c83:  x73  = 1
c84:  x74  = 1
c85:  x75  = 1
c86:  x76  = 1
c87:  x77  = 1
c88:  x78  = 1
c89:  x79  = 1
c90:  x80  = 1
c91:  x81  = 1
c92:  x82  = 1
c93:  x83  = 1
c94:  x84  = 1
c95:  x85  = 1
c96:  x86  = 1
c97:  x87  = 1
c98:  x88  = 1
c99:  x89  = 1
c100: x90  = 1
c101: x91  = 1
c102: x92  = 1
c103: x93  = 1
c104: x94  = 1
c105: x95  = 1
c106: x96  = 1
c107: x97  = 1
c108: x98  = 1
c109: x99  = 1
c110: x100  = 1
End
```

Εικόνα 4.4: Εξισώσεις του RMP

Λύνοντας το παραπάνω RMP παίρνουμε τιμή αντικειμενικής συνάρτησης Z=131.000.000 μονάδες. Στην συνέχεια θα πρέπει να λύσουμε και το column generation sub-problem. Όπως ήδη έχουμε αναφέρει και παραπάνω για το column generation δημιουργούμε μια συνάρτηση για την αρχικοποίηση των τιμών των κόμβων και έπειτα με μια άλλη συνάρτηση κάνουμε τον έλεγχο για των συνδυασμό των κόμβων και την δημιουργία του προγράμματος. Μόλις δημιουργηθούν όλα τα δυνατά προγράμματα για κάθε πιλότο, επιλέγονται 3 από αυτά και εισάγονται στο master problem το οποίο λύνεται ξανά. Η λύση του master problem θα μας δώσει νέες δυικές τιμές, οι οποίες θα αντικαταστήσουν τις προηγούμενες. Έχοντας τις νέες δυικές τιμές λύνουμε ξανά το πρόβλημα και βρίσκουμε νέα προγράμματα για κάθε πιλότο. Η διαδικασία συνεχίζεται έως ότου να μην υπάρχει αρνητικό reduce cost. Μόλις συμβεί αυτό η διαδικασία του column generation τερματίζει.

Παρακάτω θα παρουσιάσουμε τις εξισώσεις μετά την πρώτη επανάληψη του column generation. Όπως θα δούμε οι μεταβλητές έχουν αυξηθεί και αυτό οφείλεται στον αριθμό των paths που ισούται με 3. Έτσι σε κάθε επανάληψη του column generation θα προστίθενται 3 μεταβλητές για κάθε πιλότο. Επειδή ο αριθμός των μεταβλητών που εισέρχονται στο πρόβλημα είναι μεγάλος μετά από κάθε επανάληψη, θα παρουσιάσουμε το πρόβλημα με την μορφή εξισώσεων μόνο για την πρώτη επανάληψη.

```
Minimize
 obj: 1000000 x1 + 1000000 x2 + 2000000 x3 + 1000000 x4 + 1000000 x5
      + 2000000 x6 + 2000000 x7 + 2000000 x8 + 1000000 x9 + 1000000 x10
      + 2000000 x11 + 2000000 x12 + 2000000 x13 + 1000000 x14 + 1000000 x15
      + 2000000 x16 + 1000000 x17 + 1000000 x18 + 1000000 x19 + 1000000 x20
      + 1000000 x21 + 1000000 x22 + 2000000 x23 + 1000000 x24 + 1000000 x25
      + 1000000 x26 + 1000000 x27 + 1000000 x28 + 1000000 x29 + 1000000 x30
      + 1000000 x31 + 1000000 x32 + 1000000 x33 + 1000000 x34 + 1000000 x35
      + 1000000 x36 + 2000000 x37 + 2000000 x38 + 1000000 x39 + 2000000 x40
      + 2000000 x41 + 1000000 x42 + 1000000 x43 + 1000000 x44 + 1000000 x45
      + 1000000 x46 + 1000000 x47 + 1000000 x48 + 1000000 x49 + 1000000 x50
      + 1000000 x51 + 2000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
      + 2000000 x56 + 1000000 x57 + 2000000 x58 + 1000000 x59 + 1000000 x60
      + 2000000 x61 + 1000000 x62 + 2000000 x63 + 1000000 x64 + 1000000 x65
      + 1000000 x66 + 2000000 x67 + 1000000 x68 + 2000000 x69 + 1000000 x70
      + 2000000 x71 + 2000000 x72 + 2000000 x73 + 1000000 x74 + 1000000 x75
      + 2000000 x76 + 1000000 x77 + 1000000 x78 + 2000000 x79 + 1000000 x80
      + 1000000 x81 + 1000000 x82 + 1000000 x83 + 1000000 x84 + 1000000 x85
      + 1000000 x86 + 1000000 x87 + 2000000 x88 + 1000000 x89 + 1000000 x90
      + 2000000 x91 + 1000000 x92 + 2000000 x93 + 2000000 x94 + 1000000 x95
      + 1000000 x96 + 2000000 x97 + 2000000 x98 + 1000000 x99 + 1000000 x100
      + 107000 x111 + 107000 x112 + 107000 x113 + 105000 x114 + 105000 x115
      + 105000 x116 + 103000 x117 + 103000 x118 + 103000 x119 + 104000 x120
      + 104000 x121 + 104000 x122 + 103000 x123 + 103000 x124 + 103000 x125
      + 101000 x126 + 101000 x127 + 101000 x128 + 101000 x129 + 101000 x130
      + 101000 x131 + 106000 x132 + 106000 x133 + 106000 x134 + 106000 x135
      + 106000 x136 + 106000 x137 + 105000 x138 + 105000 x139 + 105000 x140
```

```
Subject To
 c1:   x101 + x111 + x112 + x113  = 1
 c2:   x102 + x114 + x115 + x116  = 1
 c3:   x103 + x117 + x118 + x119  = 1
 c4:   x104 + x120 + x121 + x122  = 1
 c5:   x105 + x123 + x124 + x125  = 1
 c6:   x106 + x126 + x127 + x128  = 1
 c7:   x107 + x129 + x130 + x131  = 1
 c8:   x108 + x132 + x133 + x134  = 1
 c9:   x109 + x135 + x136 + x137  = 1
 c10:  x110 + x138 + x139 + x140  = 1
 c11:  x1 + x129 + x139  = 1
 c12:  x2 + x115 + x116 + x130 + x132 + x135 + x136 + x137  = 1
 c13:  x3 + x123 + x124 + x125  = 1
 c14:  x4 + x112 + x122 + x126 + x128 + x133 + x134  = 1
 c15:  x5 + x111 + x113 + x114 + x120 + x121 + x127 + x131 + x138 + x140  = 1
 c16:  x6 + x111 + x112 + x113 + x117 + x118 + x119 + x120 + x121 + x122 + x126
       + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136
       + x137  = 1
 c17:  x7 + x123 + x124 + x125  = 1
 c18:  x8  = 1
 c19:  x9 + x138 + x139 + x140  = 1
 c20:  x10 + x114 + x115 + x116  = 1
 c21:  x11 + x114 + x116 + x120 + x121 + x122 + x126 + x127 + x128 + x129
       + x130 + x131 + x132 + x133 + x135 + x137  = 1
 c22:  x12 + x111 + x112 + x113 + x134 + x136 + x138 + x139 + x140  = 1
 c23:  x13 + x115  = 1
 c24:  x14 + x117 + x118 + x119  = 1
 c25:  x15  = 1
 c26:  x16  = 1
 c27:  x17  = 1
 c28:  x18 + x123 + x124 + x125  = 1
 c29:  x19 + x117 + x118 + x119  = 1
 c30:  x20 + x111 + x112 + x113 + x120 + x122 + x125 + x126 + x127 + x128
       + x132 + x133 + x134 + x138 + x139 + x140  = 1
 c31:  x21 + x121 + x123  = 1
 c32:  x22  = 1
 c33:  x23 + x114 + x115 + x116 + x117 + x118 + x119 + x124 + x129 + x130
       + x131 + x135 + x136 + x137  = 1
 c34:  x24 + x111 + x112 + x113 + x120 + x121 + x122 + x126 + x127 + x128
       + x132 + x133 + x134  = 1
 c35:  x25 + x123 + x125  = 1
 c36:  x26 + x138 + x139 + x140  = 1
```

```
c37:  x27 + x126 + x127 + x128 + x138 + x139 + x140  = 1
c38:  x28 + x123 + x124 + x125 + x126 + x127 + x128 + x129 + x130 + x131
      + x132 + x133 + x134  = 1
c39:  x29 + x114 + x115 + x116 + x135 + x136 + x137  = 1
c40:  x30 + x111 + x112 + x113 + x118  = 1
c41:  x31 + x117 + x119 + x120 + x121 + x122 + x138 + x139 + x140  = 1
c42:  x32  = 1
c43:  x33 + x117 + x118 + x120 + x121 + x122 + x129 + x130 + x131 + x132
      + x133 + x134  = 1
c44:  x34 + x114 + x115 + x116 + x119 + x126 + x127 + x135 + x136 + x137  = 1
c45:  x35 + x123 + x124 + x125  = 1
c46:  x36 + x128  = 1
c47:  x37 + x111 + x112 + x113 + x138 + x139 + x140  = 1
c48:  x38 + x114 + x115 + x116 + x117 + x118 + x119 + x120 + x121 + x122
      + x123 + x124 + x125 + x136 + x137  = 1
c49:  x39  = 1
c50:  x40 + x132 + x133 + x134 + x135  = 1
c51:  x41 + x126 + x127 + x128  = 1
c52:  x42 + x111 + x112 + x129 + x130 + x131 + x139 + x140  = 1
c53:  x43 + x113 + x138  = 1
c54:  x44 + x111 + x112 + x113 + x120 + x122 + x123 + x124 + x125 + x127
      + x128 + x129 + x130 + x131 + x136 + x137  = 1
c55:  x45 + x114 + x115 + x116 + x117 + x118 + x119 + x121 + x126 + x132
      + x133 + x134 + x138 + x139 + x140  = 1
c56:  x46  = 1
c57:  x47 + x135  = 1
c58:  x48  = 1
c59:  x49 + x114 + x115 + x116 + x120 + x121 + x122 + x129 + x130 + x131
      + x132 + x133 + x134 + x135 + x136 + x137  = 1
c60:  x50 + x123 + x124 + x125  = 1
c61:  x51 + x126 + x127 + x128 + x138 + x139 + x140  = 1
c62:  x52 + x114 + x115 + x116 + x120 + x121 + x122 + x126 + x127 + x128
      + x135 + x136 + x137  = 1
c63:  x53 + x111 + x112 + x113 + x117 + x118 + x119 + x132 + x133 + x134  = 1
c64:  x54 + x123 + x124 + x125 + x129 + x130 + x131  = 1
c65:  x55 + x117 + x118 + x119 + x132 + x133 + x134  = 1
c66:  x56 + x111 + x112 + x113 + x129 + x130 + x131 + x138 + x139 + x140  = 1
c67:  x57 + x123 + x124 + x125  = 1
c68:  x58 + x114 + x115 + x116  = 1
c69:  x59 + x120 + x121 + x122 + x126 + x127 + x128  = 1
c70:  x60 + x123 + x124 + x125 + x132 + x133 + x134 + x135 + x136 + x137  = 1
c71:  x61 + x117 + x118 + x119  = 1
```

```
c72: x62  = 1
c73: x63 + x111 + x112 + x113 + x123 + x124 + x125 + x126 + x127 + x128
     + x129 + x130 + x131 + x138 + x139 + x140  = 1
c74: x64  = 1
c75: x65  = 1
c76: x66  = 1
c77: x67 + x120 + x121 + x122 + x132 + x133 + x134 + x135 + x136 + x137  = 1
c78: x68 + x114 + x115 + x116 + x117 + x118 + x119  = 1
c79: x69  = 1
c80: x70  = 1
c81: x71 + x111 + x112 + x113 + x114 + x115 + x116 + x123 + x124 + x125
     + x126 + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x138
     + x139 + x140  = 1
c82: x72 + x120 + x121 + x122  = 1
c83: x73 + x117 + x118 + x119 + x135 + x136 + x137  = 1
c84: x74  = 1
c85: x75  = 1
c86: x76 + x114 + x115 + x116 + x117 + x118 + x119 + x126 + x127 + x128
     + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136 + x137 + x138
     + x139 + x140  = 1
c87: x77 + x111 + x112 + x113  = 1
c88: x78 + x120 + x121 + x122  = 1
c89: x79 + x123 + x124 + x125  = 1
c90: x80  = 1
c91: x81  = 1
c92: x82  = 1
c93: x83  = 1
c94: x84 + x114 + x115 + x116 + x120 + x121 + x122 + x126 + x127 + x128
     + x138 + x139 + x140  = 1
c95: x85 + x111 + x112 + x113 + x117 + x118 + x119 + x132 + x133 + x134
     + x135 + x136 + x137  = 1
c96: x86 + x111 + x112 + x113 + x117 + x118 + x119 + x120 + x121 + x122
     + x123 + x124 + x125 + x126 + x127 + x128  = 1
c97: x87 + x135 + x136 + x137  = 1
c98: x88 + x114 + x115 + x116 + x117 + x118 + x119 + x120 + x121 + x122
     + x123 + x124 + x125 + x138 + x139 + x140  = 1
c99: x89 + x111 + x112 + x113 + x126 + x127 + x128 + x129 + x130 + x131
     + x135 + x136 + x137  = 1
c100: x90 + x129 + x130 + x131  = 1
c101: x91 + x111 + x112 + x113 + x126 + x127 + x128 + x132 + x133 + x134
     + x135 + x136 + x137  = 1
c102: x92 + x114 + x115 + x116 + x117 + x118 + x119 + x123 + x124 + x125

c103: x93 + x138 + x139 + x140  = 1
c104: x94  = 1
c105: x95 + x120 + x121 + x122  = 1
c106: x96  = 1
c107: x97 + x111 + x112 + x113 + x117 + x118 + x119 + x120 + x121 + x122
      + x126 + x127 + x128 + x129 + x130 + x131  = 1
c108: x98 + x123 + x124 + x125 + x135 + x136 + x137  = 1
c109: x99 + x132 + x133 + x134  = 1
c110: x100 + x114 + x115 + x116  = 1
End
```

Εικόνα 4.5: Εξισώσεις RMP μετά την 1η επανάληψη του column generation.

Παρατηρούμε ότι ο αριθμός των μεταβλητών από 110 που ήταν πριν, τώρα έγινε 140. Αυτό είναι λογικό καθώς είπαμε ότι για κάθε πιλότο προσθέτονται 3 μεταβλητές στο πρόβλημα. Έχουμε 10 πιλότους άρα από 110 οι μεταβλητές έγιναν 140. Λύνοντας αυτό το πρόβλημα παίρνουμε τιμή αντικειμενικής συνάρτησης Z1=81189000.20, τιμή πολύ μικρότερη του αρχικού προβλήματος. Παρατηρούμε ότι με την εισαγωγή νέων μεταβλητών η τιμή της αντικειμενικής συνάρτησης βελτιώνεται. Οι λύσεις της αντικειμενικής συνάρτησης για κάθε επανάληψη του column generation sub problem είναι:

Z1=81189000.20

Z2=65593833.33

Z3=50443173.96

Z4=41970600.90

Z5=33660317.43

Z6=28463801.43

Z7=22484188.93

Z8=16592257.96

Z9=12463948.66

Z10=10039842.25

Z11=7086581.56

Z12=6421090.92

Z13=5124599.79

Z14=4335533.95

Z15=3754913.43

Z16=2360141.86

Z17=1196482.72

Z18=466636.25

Z19=354212.74

Z20=284757.05

Z21=232411.93

Z22=172405.69

Z23=129966.41

Z24=79968.04

Z25=66550.50

Z26=48665.74

Z27=44488.01

Z28=43788.84

Z29=43207.93

Z30=43152.87

Z31=42870.02

Όπως παρατηρούμε η τιμή της αντικειμενικής συνάρτησης μειώνεται σε κάθε επανάληψη του column generation. Η μείωση αυτή είναι πολύ μεγάλη αφού από Z=131.000.000 μονάδες μετά από 31 επαναλήψεις, μειώθηκε σε Z31=42.870,02 μονάδες. Παρόλα αυτά, αυτή δεν είναι και η βέλτιστη λύση του προβλήματος, διότι η τιμή των μεταβλητών απόφασης δεν είναι ακέραιοι αριθμοί. Το επόμενο βήμα είναι η μέθοδος branching προκειμένου να εξαλειφθούν οι δεκαδικές τιμές. Κατά το branching θα λύνουμε μόνο τα προβλήματα στα οποία θέτεται η τιμή της μεταβλητής ίση με 1. Όπως έχουμε ήδη αναφέρει η τιμή Z31=42.870,02 αποτελεί το κάτω όριο του δέντρου, αυτό σημαίνει ότι δεν θα βρούμε σε κανέναν κλάδο του δέντρου καλύτερη τιμή από αυτή. Μετά από 9 επαναλήψεις του βήματος branching προκύπτει η βέλτιστη ακέραια λύση Z=129.000 μονάδες. Η τιμή της οποίας φαίνεται και στην παρακάτω εικόνα.

```
The solution of the master problem is master_obj=42870.022394

The solution of the problem as a intiger problem is objective_value=129000.000000

The difference between these values is dif=86129.977606

The uncovered routes are:       0
The pilot 7 has the routes:      6 14 16 42 61 73 83 90 92    with total flight time: 98 hours
The pilot 10 has the routes:     10 12 31 45 51 62 66 79 91 99    with total flight time: 91 hours
The pilot 9 has the routes:      2 36 38 48 52 67 82 89 94    with total flight time: 99 hours
The pilot 8 has the routes:      9 17 22 29 33 43 47 60 70 76 85 96    with total flight time: 96 hours
The pilot 1 has the routes:      4 15 30 39 53 58 69 75 77    with total flight time: 98 hours
The pilot 6 has the routes:      1 19 26 27 32 55 72 81 93    with total flight time: 98 hours
The pilot 5 has the routes:      7 18 25 28 35 41 50 63 71 80 86    with total flight time: 94 hours
The pilot 3 has the routes:      8 20 37 46 57 59 65 87 88 97    with total flight time: 99 hours
The pilot 4 has the routes:      3 11 21 24 40 44 78 84 98    with total flight time: 98 hours
The pilot 2 has the routes:      5 13 23 34 49 54 56 64 68 74 95 100    with total flight time: 118 hours

the solution was achieved after 9 times of branching

Elapsed time: 1125.896 seconds

Press return to continue...
```

Εικόνα 4.6: Τελική λύση του προβλήματος

Παρόλο που καμιά διαδρομή δεν έμεινε ακάλυπτη υπάρχει ένα κόστος 129.000 μονάδες, το οποίο οφείλεται στον πιλότο 2, στον οποίον έχουν ανατεθεί διαδρομές που ξεπερνούν τις 100 ώρες πτήσεων και σε κάποια ρεπό τα οποία δεν μπόρεσαν να καλυφθούν.

Οι τελικές εξισώσεις που οδήγησαν στην βέλτιστη λύση φαίνονται στην παρακάτω εικόνα.

```
Minimize
 obj: 1000000 x1 + 1000000 x2 + 2000000 x3 + 1000000 x4 + 1000000 x5
      + 2000000 x6 + 2000000 x7 + 2000000 x8 + 1000000 x9 + 1000000 x10
      + 2000000 x11 + 2000000 x12 + 2000000 x13 + 1000000 x14 + 1000000 x15
      + 2000000 x16 + 1000000 x17 + 1000000 x18 + 1000000 x19 + 1000000 x20
      + 1000000 x21 + 1000000 x22 + 2000000 x23 + 1000000 x24 + 1000000 x25
      + 1000000 x26 + 1000000 x27 + 1000000 x28 + 1000000 x29 + 1000000 x30
      + 1000000 x31 + 1000000 x32 + 1000000 x33 + 1000000 x34 + 1000000 x35
      + 1000000 x36 + 2000000 x37 + 2000000 x38 + 1000000 x39 + 2000000 x40
      + 2000000 x41 + 1000000 x42 + 1000000 x43 + 1000000 x44 + 1000000 x45
      + 1000000 x46 + 1000000 x47 + 1000000 x48 + 1000000 x49 + 1000000 x50
      + 1000000 x51 + 2000000 x52 + 1000000 x53 + 1000000 x54 + 1000000 x55
      + 2000000 x56 + 1000000 x57 + 2000000 x58 + 1000000 x59 + 1000000 x60
      + 2000000 x61 + 1000000 x62 + 2000000 x63 + 1000000 x64 + 1000000 x65
      + 1000000 x66 + 2000000 x67 + 1000000 x68 + 2000000 x69 + 1000000 x70
      + 2000000 x71 + 2000000 x72 + 2000000 x73 + 1000000 x74 + 1000000 x75
      + 2000000 x76 + 1000000 x77 + 1000000 x78 + 2000000 x79 + 1000000 x80
      + 1000000 x81 + 1000000 x82 + 1000000 x83 + 1000000 x84 + 1000000 x85
      + 1000000 x86 + 1000000 x87 + 2000000 x88 + 1000000 x89 + 1000000 x90
      + 2000000 x91 + 1000000 x92 + 2000000 x93 + 2000000 x94 + 1000000 x95
      + 1000000 x96 + 2000000 x97 + 2000000 x98 + 1000000 x99 + 1000000 x100
      + 1000 x102 + 1000 x103 + 4000 x104 + 5000 x105 + 6000 x106 + 1000 x107
      + 3000 x108 + 2000 x109 + 2000 x110 + 2000 x111 + 2000 x112 + 2000 x113
      + 1000 x114 + 1000 x115 + 3000 x116 + 103000 x117 + 2000 x118
      + 103000 x119 + 4000 x120 + 4000 x121 + 3000 x122 + 2000 x123 + 2000 x124
      + 4000 x125 + 4000 x126 + 104000 x127 + 4000 x128 + 4000 x129
      + 104000 x130 + 104000 x131 + 104000 x132 + 104000 x133 + 3000 x134
      + 3000 x135 + 3000 x136 + 3000 x137 + 3000 x138 + 103000 x139 + 3000 x140
      + 3000 x141 + 3000 x142
```

```
Subject To
 c1:   x106  = 1
 c2:   x101 + x109 + x110 + x113 + x114 + x115 + x116 + x117 + x118 + x119
       + x120 + x121 + x122 + x123 + x124 + x125 + x126 + x127 + x128 + x129
       + x130 + x131 + x132 + x133 + x134 + x135 + x136 + x137 + x138 + x139
       + x140 + x141 + x142  = 1
 c3:   x111  = 1
 c4:   x112  = 1
 c5:   x108  = 1
 c6:   x107  = 1
 c7:   x102  = 1
 c8:   x105  = 1
 c9:   x104  = 1
 c10:  x103  = 1
 c11:  x1 + x107  = 1
 c12:  x2 + x104  = 1
 c13:  x3 + x112  = 1
 c14:  x4 + x106  = 1
 c15:  x5 + x109 + x110 + x113 + x114 + x115 + x118 + x119 + x120 + x121 + x122
       + x123 + x124 + x125 + x126 + x127 + x128 + x129 + x130 + x131 + x132
       + x133 + x134 + x135 + x136 + x137 + x138 + x139 + x140 + x141 + x142
        = 1
 c16:  x6 + x102  = 1
 c17:  x7 + x108  = 1
 c18:  x8 + x111  = 1
 c19:  x9 + x105  = 1
 c20:  x10 + x103  = 1
 c21:  x11 + x112  = 1
 c22:  x12 + x103  = 1
 c23:  x13 + x109 + x110 + x113 + x114 + x115 + x116 + x117 + x118 + x119
       + x120 + x121 + x122 + x123 + x124 + x128 + x129 + x130 + x131 + x132
       + x133 + x134 + x135 + x136 + x138 + x139 + x142  = 1
 c24:  x14 + x102  = 1
 c25:  x15 + x106  = 1
 c26:  x16 + x102  = 1
 c27:  x17 + x105  = 1
 c28:  x18 + x108  = 1
 c29:  x19 + x107  = 1
 c30:  x20 + x111  = 1
 c31:  x21 + x112  = 1
 c32:  x22 + x105  = 1
```

```
c33:  x23 + x109 + x113 + x114 + x115 + x116 + x117 + x118 + x119 + x120
      + x121 + x122 + x123 + x124 + x125 + x126 + x127 + x131 + x132 + x134
      + x137 + x138 + x139 + x140  = 1
c34:  x24 + x112  = 1
c35:  x25 + x108  = 1
c36:  x26 + x107  = 1
c37:  x27 + x107  = 1
c38:  x28 + x108  = 1
c39:  x29 + x105  = 1
c40:  x30 + x106  = 1
c41:  x31 + x103  = 1
c42:  x32 + x107  = 1
c43:  x33 + x105  = 1
c44:  x34 + x109 + x114 + x117 + x118 + x119 + x120 + x124 + x125 + x126
      + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136
      + x137 + x138 + x139 + x140 + x141 + x142  = 1
c45:  x35 + x108  = 1
c46:  x36 + x104  = 1
c47:  x37 + x111  = 1
c48:  x38 + x104  = 1
c49:  x39 + x106  = 1
c50:  x40 + x112  = 1
c51:  x41 + x108  = 1
c52:  x42 + x102  = 1
c53:  x43 + x105  = 1
c54:  x44 + x112  = 1
c55:  x45 + x103  = 1
c56:  x46 + x111  = 1
c57:  x47 + x105  = 1
c58:  x48 + x104  = 1
c59:  x49 + x109 + x113 + x114 + x115 + x116 + x117 + x120 + x121 + x122
      + x123 + x124 + x126 + x127 + x129 + x130 + x132 + x133 + x136 + x137
      + x138 + x139 + x140 + x141 + x142  = 1
c60:  x50 + x108  = 1
c61:  x51 + x103  = 1
c62:  x52 + x104  = 1
c63:  x53 + x106  = 1
c64:  x54 + x116 + x117 + x120 + x121 + x122 + x123 + x124 + x125 + x126
      + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136
      + x137 + x138 + x139 + x140 + x141 + x142  = 1
c65:  x55 + x107  = 1
c66:  x56 + x110 + x113 + x114 + x115 + x116 + x117 + x118 + x119 + x120
```

```
c67:  x57 + x111  = 1
c68:  x58 + x106  = 1
c69:  x59 + x111  = 1
c70:  x60 + x105  = 1
c71:  x61 + x102  = 1
c72:  x62 + x103  = 1
c73:  x63 + x108  = 1
c74:  x64 + x109 + x110 + x116 + x117 + x118 + x119 + x122 + x123 + x124
      + x125 + x127 + x128 + x130 + x131 + x132 + x133 + x139 + x140 + x141
      + x142  = 1
c75:  x65 + x111  = 1
c76:  x66 + x103  = 1
c77:  x67 + x104  = 1
c78:  x68 + x109 + x116 + x117 + x118 + x119 + x120 + x121 + x125 + x126
      + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136
      + x137 + x138 + x139 + x140 + x141 + x142  = 1
c79:  x69 + x106  = 1
c80:  x70 + x105  = 1
c81:  x71 + x108  = 1
c82:  x72 + x107  = 1
c83:  x73 + x102  = 1
c84:  x74 + x109 + x113 + x114 + x115 + x116 + x117 + x118 + x119 + x122
      + x123 + x124 + x125 + x126 + x127 + x128 + x129 + x130 + x131 + x132
      + x133 + x134 + x135 + x136 + x137 + x138 + x139 + x140 + x141 + x142
       = 1
c85:  x75 + x106  = 1
c86:  x76 + x105  = 1
c87:  x77 + x106  = 1
c88:  x78 + x112  = 1
c89:  x79 + x103  = 1
c90:  x80 + x108  = 1
c91:  x81 + x107  = 1
c92:  x82 + x104  = 1
c93:  x83 + x102  = 1
c94:  x84 + x112  = 1
c95:  x85 + x105  = 1
c96:  x86 + x108  = 1
c97:  x87 + x111  = 1
c98:  x88 + x111  = 1
c99:  x89 + x104  = 1

c100: x90 + x102  = 1
c101: x91 + x103  = 1
c102: x92 + x102  = 1
c103: x93 + x107  = 1
c104: x94 + x104  = 1
c105: x95 + x109 + x110 + x113 + x119 + x120 + x121 + x122 + x125 + x126
      + x127 + x128 + x129 + x130 + x131 + x132 + x133 + x134 + x135 + x136
      + x137 + x138 + x139 + x140 + x141 + x142  = 1
c106: x96 + x105  = 1
c107: x97 + x111  = 1
c108: x98 + x112  = 1
c109: x99 + x103  = 1
c110: x100 + x109 + x110 + x113 + x114 + x115 + x116 + x117 + x118 + x119
      + x120 + x121 + x122 + x123 + x124 + x125 + x126 + x127 + x131 + x132
      + x133 + x134 + x135 + x136 + x137 + x138 + x139 + x140 + x141 + x142
       = 1
End
```

Εικόνα 4.7: Τελικές εξισώσεις του προβλήματος.

ΚΕΦΑΛΑΙΟ 5

5.1 Υπολογιστική εμπειρία

Προκειμένου να ελεγχθεί η ορθότητα και η αποτελεσματικότητα του αλγορίθμου θα παρουσιάσουμε στη συνέχεια μια σειρά από παραδείγματα ώστε να ελεγχθεί η ορθότητα του αλγορίθμου καθώς και να γίνει σύγκριση και ανάλυση των αποτελεσμάτων τους. Τα παραδείγματα που θα εξετάσουμε θα διαφέρουν ως προς το μέγεθος του πληρώματος, το μέγεθος των διαδρομών, καθώς και ως προς την παράμετρο paths. Όπως και στο προηγούμενο παράδειγμα ορίζουμε 1.000.000 μονάδες το κόστος ακάλυπτης διαδρομής, 1.000 μονάδες το κόστος για κάθε ρεπό που δεν καλύπτεται και 100.000 μονάδες το κόστος για την υπέρβαση των συνολικών ωρών πτήσης. Στα κεφάλαια που θα ακολουθήσουν θα γίνει παρουσίαση για κάθε περίπτωση ξεχωριστά. Για κάθε μια περίπτωση επιλύθηκαν είκοσι διαφορετικά προβλήματα για πιο αξιόπιστα αποτελέσματα.

	Time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered Routes
N=10 R=100 Paths=1	0:00:50	9	87.247	164.000	76.752	0
N=65 R=550 Paths=1	0:12:16	63	7.423	146.700	139.275	0
Paths=3	0:16:03	63	6.180	105.011	98.830	0
Paths=4	0:18:04	63	6.388	98.200	91.811	0
Paths=10	0:31:48	63	7.171	68.996	61.824	0
N=45 R=550 Paths=1	0:11:09	43	1.322.197	1.567.400	245.201	0
N=70 R=700 Paths=1	1:40:25	68	507.068	605.250	98.181	0

Πίνακας 5.1: Συγκεντρωτικός πίνακας αποτελεσμάτων.

5.2 Εφαρμογές

Οι εφαρμογές που θα πραγματοποιηθούν είναι επτά (7), για κάθε μία θα εξαχθούν 20 διαφορετικοί υπολογισμοί. Στην εφαρμογή 1, ορίζουμε N=10, R=100 και Paths=1. Στην εφαρμογή 2 ορίζουμε, N=70, R=700 και Paths=1. Στην εφαρμογή 3 ορίζουμε N=45, R=550 και paths=1. Τέλος στην εφαρμογή 4, 5, 6, 7 ορίζουμε N=65, R=550, Paths=1, Paths=3, Paths=4 και Paths=10.

5.2.1 Εφαρμογή 1η

Στην πρώτη εφαρμογή ορίζουμε N=10, R=100 και paths=1

Στον πίνακα που ακολουθεί παραθέτουμε τις μέσες τιμές καθώς και την μέγιστη και την ελάχιστη τιμή από τους 20 διαφορετικούς υπολογισμούς που πραγματοποιήθηκαν.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	0:00:38	7	4.086	15.000	10.914	0
Max	0:00:73	9	173.361	332.000	179.505	0
Average	0:00:50	9	87.247	164.000	76.752	0

Πίνακας 5.2: Αποτελέσματα 1ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι τα 73 δευτερόλεπτα, ενώ ο ελάχιστος τα 38 δευτερόλεπτα με μέση τιμή τα 50 δευτερόλεπτα. Οι μέσες τιμές του MP κυμαίνονται από 4.086 έως 173.361, με μέση τιμή 87.247. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 164.000 με 15.000 και 332.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 76.752 ενώ η μέση τιμή του branching είναι 9. Οι διαδρομές καλύφθηκαν σε όλα τα προβλήματα αλλά οι συνολικές ώρες πτήσεων για κάποιους πιλότους ξεπέρασαν το επιτρεπτό όριο για αυτό και το αυξημένο κόστος στην τελική βέλτιστη λύση. Υπενθυμίζουμε ότι το κόστος για την υπέρβαση των 100 ωρών είναι 100.000 μονάδες.

5.2.2 Εφαρμογή 2η

Ορίζουμε N=70, R=700 και paths=1

Στην προηγούμενη εφαρμογή ο χρόνος επίλυσης ήταν πάρα πολύ μικρός. Για το λόγο αυτό στην εφαρμογή αυτή θα αυξήσουμε και τον αριθμό των πιλότων αλλά και τον αριθμό των διαδρομών. Στον παρακάτω πίνακα παραθέτουμε τις τιμές που προέκυψαν μετά την επίλυση των 20 διαφορετικών υπολογισμών.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	01.20.52	67	298.564	317.896	11.250	0
Max	02.15.58	69	766.005	848.000	227.319	0
Average	01.40.25	68	507.068	605.250	98.181	0

Πίνακας 5.3: Αποτελέσματα 2ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι 2 ώρες 15 λεπτά και 58 δευτερόλεπτα, ενώ ο ελάχιστος 1 ώρα 20 λεπτά και 52 δευτερόλεπτα με μέση τιμή 1 ώρα 40 λεπτά και 25 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 298.564 έως 766.005, με μέση τιμή 507.068. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 605.250 με 317.896 και 848.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 98.181 ενώ η μέση τιμή του branching είναι 68. Αυξάνοντας τις τιμές των διαδρομών και των ιπταμένων αυξήθηκε και ο χρόνος επίλυσης του προβλήματος καθώς και η τιμή του branching. Οι διαδρομές καλύφθηκαν όλες και το κόστος οφείλεται στην υπέρβαση των ωρών πτήσης αλλά και σε κάποιες μέρες ανάπαυσης που δεν ικανοποιήθηκαν.

5.2.3 Εφαρμογή 3η

Ορίζουμε N=45, R=550 paths=1

Στον επόμενο πίνακα παραθέτουμε τα αποτελέσματα, τα οποία προέκυψαν από την επίλυση 20 διαφορετικών υπολογισμών.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	00.08.28	42	1.055.310	1.306.000	80.641	0
Max	00.18.16	44	1.568.944	1.823.000	360.178	0
Average	00.11.09	43	1.322.197	1.567.400	245.201	0

Πίνακας 5.4: Αποτελέσματα 3ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι τα 18 λεπτά και 16 δευτερόλεπτα, ενώ ο ελάχιστος τα 8 λεπτά και 28 δευτερόλεπτα με μέση τιμή τα 11 λεπτά και τα 9 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 1.055.310 έως 1.568.944, με μέση τιμή 1.322.197. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 1.567.400 με 1.306.000 και 1.823.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 245.201 ενώ η μέση τιμή του branching είναι 43. Οι συνολικές ώρες πτήσεων για κάποιους πιλότους ξεπέρασαν και πάλι κατά πολύ αυτήν την φορά το επιτρεπτό όριο για αυτό και το αυξημένο κόστος στην τελική βέλτιστη λύση. Παρά το γεγονός ότι καλύφθηκαν όλες οι πτήσεις αυτό που κατανοούμε από τα αποτελέσματα είναι ότι ο αριθμός των ιπταμένων δεν είναι επαρκής για τις συγκεκριμένες διαδρομές, με αποτέλεσμα να ξεπερνάτε το επιτρεπτό όριο των 100 ωρών και να έχουμε αυτήν την σημαντική αύξηση στην τελική βέλτιστη λύση.

5.2.4 Εφαρμογή 4η

Ορίζουμε N=65, R=550 paths=1

Σε αυτήν την εφαρμογή θα αυξήσουμε τον αριθμό των πιλότων ενώ θα κρατήσουμε σταθερό τον αριθμό των διαδρομών και των μεταβλητών που εισάγονται ανά επανάληψη. Στην παραπάνω εφαρμογή ενώ καλύφθηκαν και στους 20 υπολογισμούς οι διαδρομές είδαμε ότι υπήρχε ένα υψηλό κόστος, το οποίο οφείλεται στον χαμηλό αριθμό ιπταμένων σε σχέση με τις διαδρομές και στην υπέρβαση των συνολικών επιτρεπτών ωρών πτήσης. Ας δούμε τι θα συμβεί αν ο αριθμός των ιπταμένων είναι επαρκής για τις συγκεκριμένες διαδρομές R=550.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	0:08:13	61	2.360	7.000	-151	0
Max	0:18:17	64	14.229	335.000	326.113	0
Average	0:12:16	63	7.423	146.700	139.275	0

Πίνακας 5.5: Αποτελέσματα 4ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι τα 18 λεπτά και 17 δευτερόλεπτα, ενώ ο ελάχιστος τα 8 λεπτά και 13 δευτερόλεπτα με μέση τιμή τα 12 λεπτά και τα 16 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 2.360 έως 14.229, με μέση τιμή 7.423. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 146.700 με 7.000 και 335.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 139.275 ενώ η μέση τιμή του branching είναι 63. Παρατηρούμε ότι με την αύξηση των ιπταμένων υπάρχει μία αύξηση στην τιμή του branching (από 43 σε 63) ενώ σχετικά σε ίδιες τιμές κυμαίνετε ο χρόνος επίλυσης. Αυτό που αξίζει να σχολιασθεί είναι η μείωση της τιμής του master problem. Ενώ στην προηγούμενη εφαρμογή η μέση τιμή του master problem ήταν 1.322.197, τώρα είναι 7.423. Αυτό οφείλεται στην καλύτερη διανομή των διαδρομών και στο γεγονός ότι δεν ξεπερνάτε το όριο των συνολικών ωρών πτήσεων στους ιπταμένους. Παρόλα αυτά όπως ήδη έχουμε αναφέρει η λύση του master problem δεν μπορεί να θεωρηθεί βέλτιστη γιατί περιέχει μη ακέραιες τιμές. Η τελική βέλτιστη λύση είναι 146.700 και αυτή η αύξηση σχετικά με την τιμή του master problem προκύπτει από την αναδιανομή των δρομολογίων ώστε να προκύψουν ακέραιες τιμές.

Ένα άλλο που παρατηρούμε για πρώτη φορά είναι το αρνητικό gap, δηλαδή η τιμή του master problem είναι μεγαλύτερη από την τελική βέλτιστη λύση. Τυπικά αυτό δεν μπορεί να συμβεί, καθώς η λύση του master problem μας δίνει τη μικρότερη δυνατή λύση. Η εμφάνιση αρνητικού gap έχει να κάνει κυρίως με τον αριθμό των paths. Όπως έχουμε αναφέρει παραπάνω, κάποιες φορές μια μεταβλητή που εισάγεται στο πρόβλημα και θεωρείται ως η καλύτερη, αλλά στο τελικό πρόβλημα ίσως μια άλλη μεταβλητή που δεν θεωρούνταν αρχικά η καλύτερη να το βελτιώνει περισσότερο. Για την εξάλειψη του φαινομένου θα δοθούν στην παρακάτω εφαρμογή μεγαλύτερες τιμές στη τιμή paths.

5.2.4.1 Εφαρμογή 5η

Ορίζουμε N=65, R=550 και paths=3

Σε αυτήν την εφαρμογή θα κρατήσουμε ίδιο τον αριθμό των ιπταμένων και των διαδρομών με την προηγούμενη εφαρμογή αλλά θα μεταβάλλουμε την μεταβλητή paths. Στον ακόλουθο πίνακα παραθέτουμε τα αποτελέσματα που προέκυψαν μετά το πέρας 20 διαφορετικών εφαρμογών.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	0:10:31	60	0	5.000	128	0
Max	0:26:27	64	11.138	253.000	251.499	0
Average	0:16:03	63	6.180	105.011	98.830	0

Πίνακας 5.6: Αποτελέσματα 5ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι τα 26 λεπτά και 27 δευτερόλεπτα, ενώ ο ελάχιστος τα 10 λεπτά και 31 δευτερόλεπτα με μέση τιμή τα 16 λεπτά και τα 3 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 0 έως 11.138, με μέση τιμή 6.180. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 105.011 με 5.000 και 253.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 98.830 ενώ η μέση τιμή του branching είναι 63.

Η αύξηση της τιμής του paths μείωσε την τελική βέλτιστη λύση από 146.700 σε 105.011. Αντίθετα ο συνολικός χρόνος επίλυσης όπως και η τιμή του branching δεν έχουν μεταβληθεί κατά πολύ. Ο χρόνος επίλυσης αυξήθηκε κατά τέσσερα λεπτά ενώ η τιμή του branching παρέμεινε ίδια.

5.2.4.2 Εφαρμογή 6η

Ορίζουμε N=65, R=550 και paths=4

Θα κρατήσουμε και πάλι ίδιο τον αριθμό των ιπταμένων και των διαδρομών και θα αυξήσουμε την τιμή του paths κατά μια μονάδα. Στην προηγούμενη εφαρμογή με την αύξηση του paths τα αποτελέσματα ήταν βελτιωμένα. Ας δούμε τι θα συμβεί αν αυξήσουμε και άλλο την τιμή.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	0:11:19	61	0	8.000	89	0
Max	0:30:23	64	12.248	167.000	159.253	0
Average	0:18:04	63	6.388	98.200	91.811	0

Πίνακας 5.7: Αποτελέσματα 6ης εφαρμογής.

Ο μέγιστος χρόνος επίλυσης είναι τα 30 λεπτά και 23 δευτερόλεπτα, ενώ ο ελάχιστος τα 11 λεπτά και 19 δευτερόλεπτα με μέση τιμή τα 18 λεπτά και τα 4 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 0 έως 12.248, με μέση τιμή 6.388. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 98.200 με 8.000 και 167.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 91.811 ενώ η μέση τιμή του branching είναι 63.

Ο χρόνος επίλυσης του προβλήματος δεν μεταβλήθηκε σχεδόν καθόλου. Από 16 λεπτά και 3 δευτερόλεπτα έγινε 18 λεπτά και 4 δευτερόλεπτα σε σχέση με την προηγούμενη εφαρμογή. Περεταίρω μείωση παρατηρείτε στην τιμή της τελικής βέλτιστης λύση η οποία από 105.011 μονάδες μειώθηκε σε 98.200 μονάδες.

5.2.4.3 Εφαρμογή 7η

Ορίζουμε N=65, R=550 και paths=10

Τέλος θα κρατήσουμε και πάλι ίδιο τον αριθμό των διαδρομών και των ιπταμένων και θα αυξήσουμε την τιμή του paths σε 10. Οι μέσες τιμές καθώς και οι μέγιστες και οι ελάχιστες μετά την επίλυση 20 διαφορετικών εφαρμογών παρουσιάζονται στον ακόλουθο πίνακα.

	time	Branching	Λύση MP	Τελική βέλτιστη λύση	GAP	Uncovered
Min	0:25:35	62	1.785	5.000	73	0
Max	0:51:25	64	10.096	158.000	151.182	0
Average	0:31:48	63	7.171	68.996	61.824	0

Πίνακας 5.8: Αποτελέσματα 7^{ης} εφαρμογής.

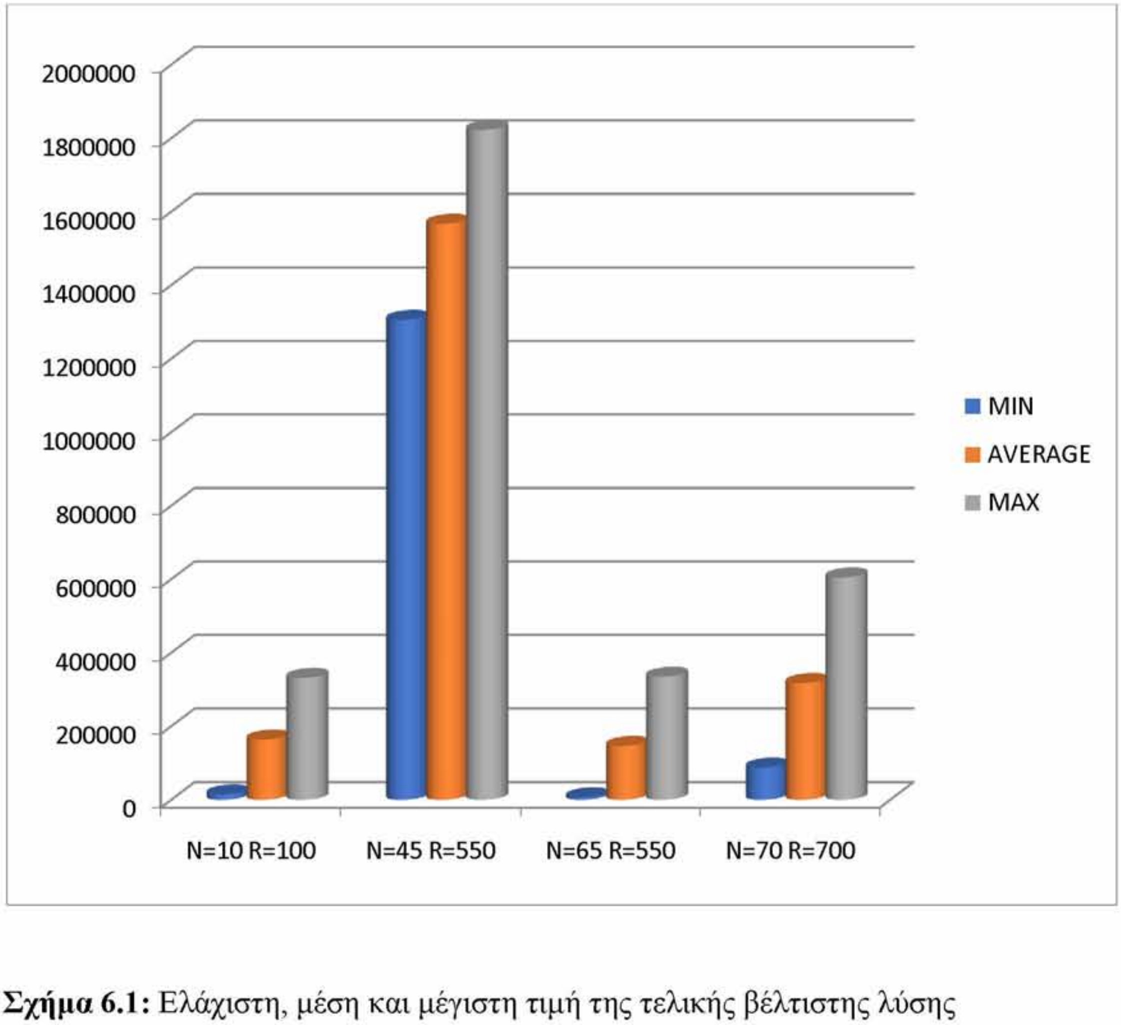
Ο μέγιστος χρόνος επίλυσης είναι τα 51 λεπτά και 25 δευτερόλεπτα, ενώ ο ελάχιστος τα 25 λεπτά και 35 δευτερόλεπτα με μέση τιμή τα 31 λεπτά και τα 48 δευτερόλεπτα. Οι τιμές του MP κυμαίνονται από 1.785 έως 10.096, με μέση τιμή 7.171. Η μέση τιμή της τελικής βέλτιστης λύσης είναι 68.996 με 5.000 και 158.000 ελάχιστη και μέγιστη τιμή αντίστοιχα. Η μέση τιμή του GAP είναι 61.824 ενώ η μέση τιμή του branching είναι 63.

Η αύξηση της τιμής paths επέφερε αύξηση όπως ήταν αναμενόμενο στον χρόνο επίλυσης της εφαρμογής, από 18 λεπτά που ήταν της προηγούμενης εφαρμογής σε 31 λεπτά. Η τιμή του branching δεν είχε σημαντικές μεταβολές όπως και η τιμή της τελικής βέλτιστης λύσης η οποία παρουσίασε μια μικρή μείωση αλλά όχι σημαντική.

ΚΕΦΑΛΑΙΟ 6

6.1 Συμπεράσματα

Στο κεφάλαιο 5 ολοκληρώθηκαν με επιτυχία όλες οι εφαρμογές, και όπως ήταν αναμενόμενο υπήρχαν αρκετές διαφοροποιήσεις τόσο από εφαρμογή σε εφαρμογή όσο και μέσα στην ίδια εφαρμογή. Οι διαφοροποιήσεις που υπήρχαν ανάμεσα στις εφαρμογές οφείλονται κυρίως στον διαφορετικό αριθμό των πιλότων και των διαδρομών που ορίζαμε σε κάθε μία από αυτές αλλά και τον αριθμό των μεταβλητών που εισάγαμε ανά επανάληψη (paths). Παρόλα αυτά διαφοροποιήσεις παρατηρήσαμε και μέσα στις ίδιες τις εφαρμογές, οι οποίες οφείλονται στον τυχαίο τρόπο που παράγονται οι τιμές μέσα στον κώδικα. Παρακάτω θα αναπαρασταθούν τα αποτελέσματα των εφαρμογών με τη μορφή γραφημάτων.

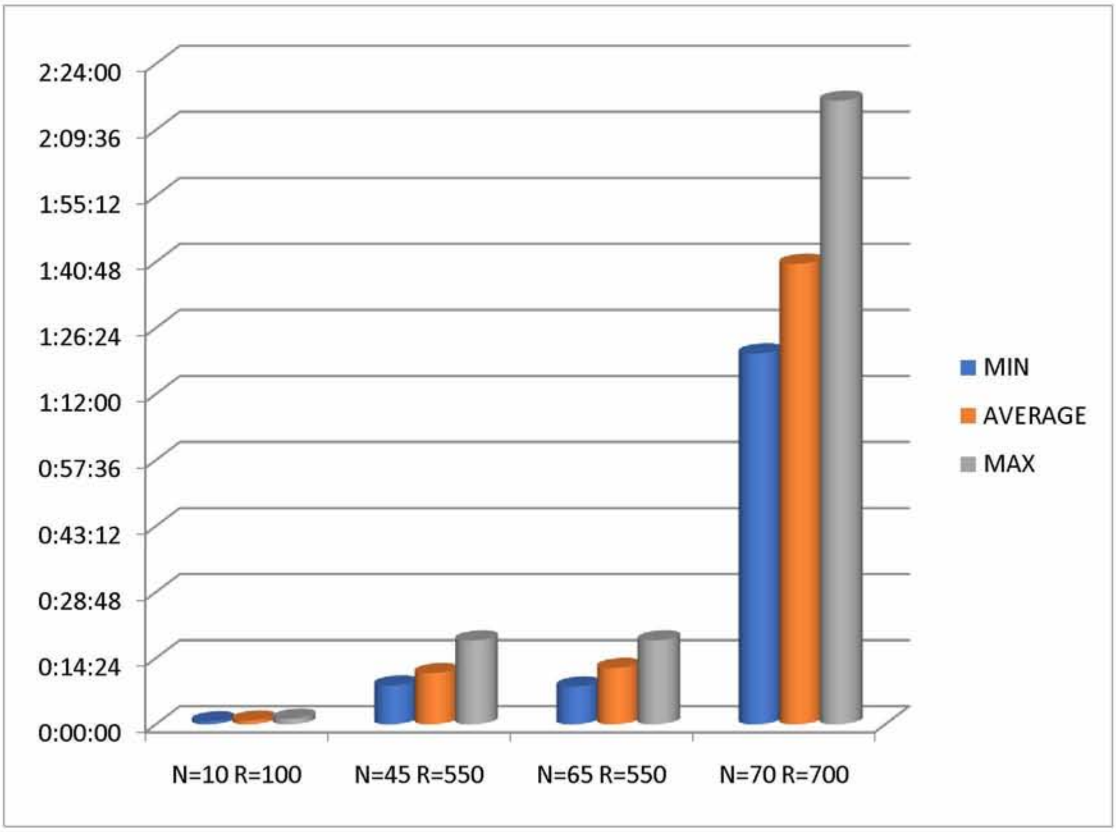


Σχήμα 6.1: Ελάχιστη, μέση και μέγιστη τιμή της τελικής βέλτιστης λύσης

Η μεγαλύτερη απόκλιση παρατηρείτε στην εφαρμογή 3 όπου N=45 και R=550. Αυτό συμβαίνει γιατί ο αριθμός των ιπταμένων δεν επαρκεί για τις συγκεκριμένες διαδρομές και προκειμένου να μην μείνει καμία διαδρομή ακάλυπτη αναθέτονται παραπάνω πτήσεις σε κάποιους ιπταμένους, με αποτέλεσμα να ξεπερνάτε το όριο των συνολικών ωρών πτήσης και να προστίθεται το κόστος των 100.000 για κάθε ιπτάμενο που ξεπέρασε το όριο αυτό.

Αυτό μπορεί εύκολα να γίνει αντιληπτό αν γίνει σύγκριση ανάμεσα στην εφαρμογή 3 (N=45, R=550) με την εφαρμογή 4 (N=65, R=550), όπου ο αριθμός των διαδρομών και η τιμή του paths παραμένουν σταθερά αλλά μια αύξηση των ιπταμένων κατά 20 μονάδες έχει επιφέρει τεράστια μείωση στην τελική βέλτιστη λύση. Στην εφαρμογή αυτή, οι ιπτάμενοι είναι περισσότεροι και η κατανομή των διαδρομών έχει γίνει με καλύτερο τρόπο, με αποτέλεσμα να υπάρχει αυτή η απόκλιση ανάμεσα στις δυο λύσεις.

Στο **σχήμα 6.2** παρουσιάζουμε την επίδραση του χρόνου επίλυσης για συγκεκριμένο paths=1.

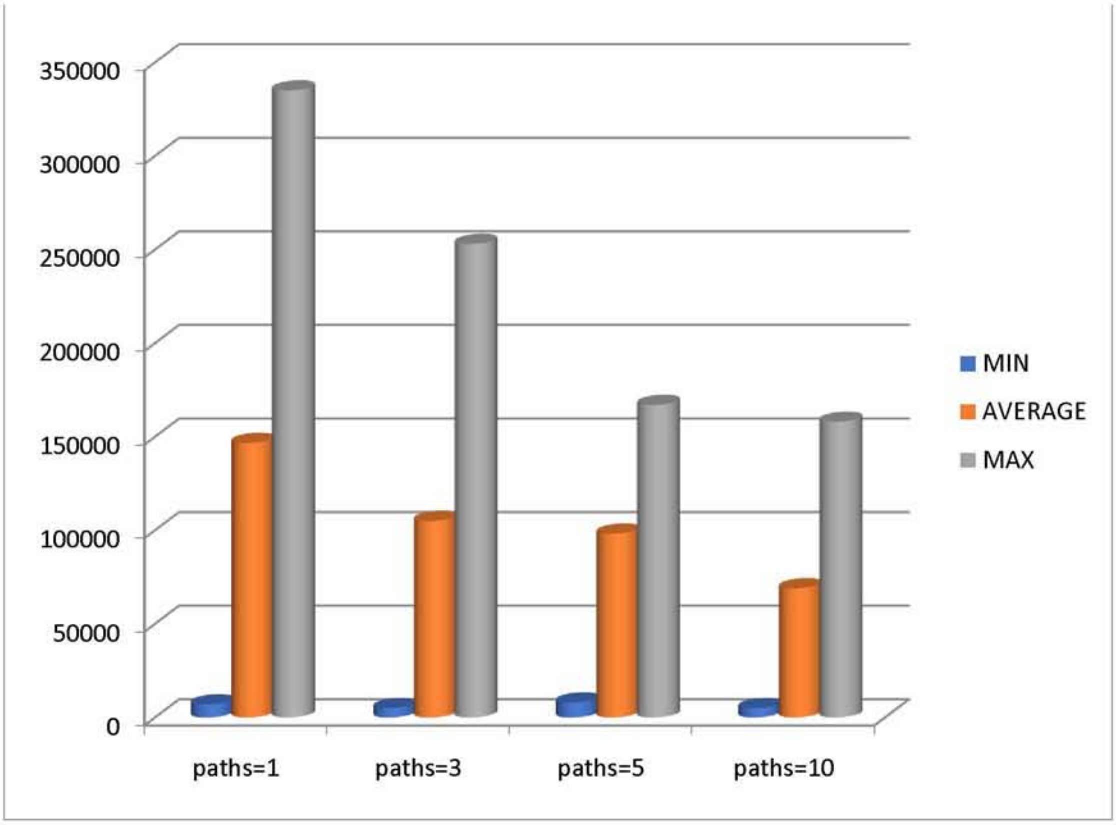


Σχήμα 6.2: Επίδραση του μεγέθους του προβλήματος στον χρόνο επίλυσής του.

Ο χρόνος επίλυσης των εφαρμογών έχει να κάνει με τον αριθμό των διαδρομών, των ιπταμένων αλλά και με τον αριθμό των μεταβλητών που εισάγουμε ανά επανάληψη. Κατανοούμε ότι όσο μεγαλύτερος ο αριθμός των τριών, τόσο μεγαλύτερος και ο χρόνος επίλυσης της εφαρμογής, όπως φαίνεται και στο παραπάνω σχήμα. Ο χρόνος επίλυσης για N=10, R=100 είναι μόνο λίγα λεπτά, για N=45, R=550 και N=65, R=550 είναι περίπου 12 λεπτά, ενώ για N=70,R=700 περίπου 2 ώρες.

Σε όλες τις εφαρμογές που εκτελέστηκαν καμία διαδρομή δεν έμεινε ακάλυπτη παρόλα αυτά σε εφαρμογές που ο αριθμός των ιπταμένων δεν ήταν επαρκής σε σχέση με τις διαδρομές, είχαμε ένα αυξημένο κόστος το οποίο οφείλεται κατά κύριο λόγο στην υπέρβαση του ορίου των ωρών πτήσης, ώστε να μην μείνει καμία διαδρομή ακάλυπτη και στην μεγάλη τιμή του penalty που έχουμε ορίσει.

Ας δούμε τι επίδραση είχε η αύξηση της τιμής paths στις εφαρμογές που πραγματοποιήθηκαν στο κεφάλαιο 5.

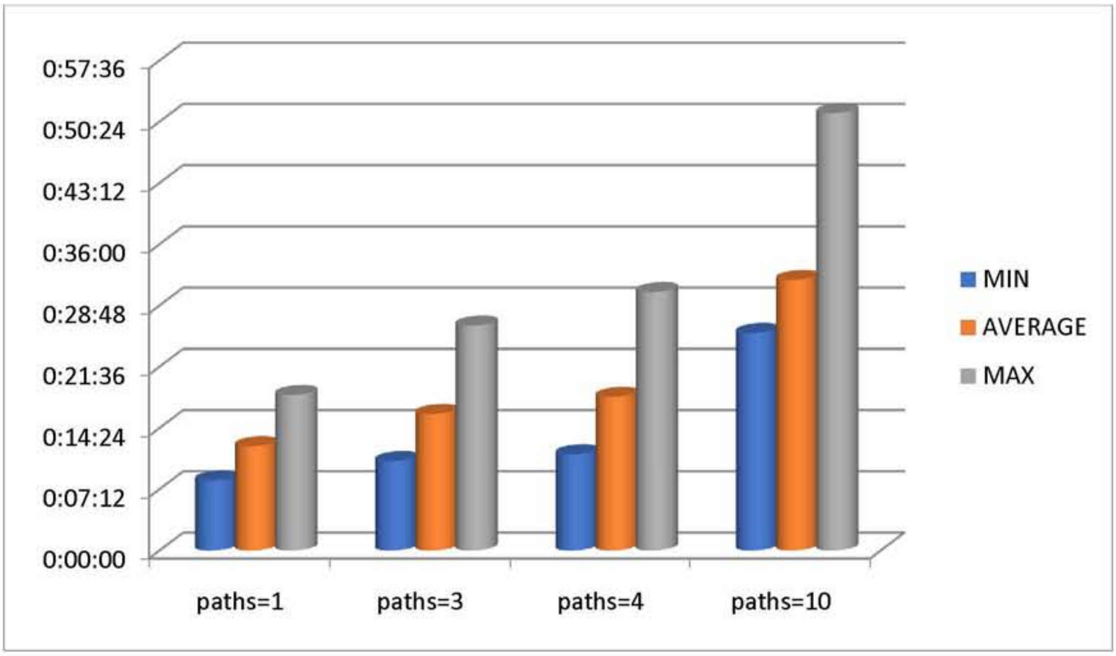


Σχήμα 6.3: Επίδραση της τιμής paths

Είναι φανερό ότι όσο αυξάνει η τιμή του paths τόσο οι τιμές της βέλτιστης τελικής λύσης είναι καλύτερες. Σε κάθε περίπτωση ανάλογα με τις απαιτήσεις της εκάστοτε εφαρμογής επιλέγουμε τη βέλτιστη τιμή του paths, ώστε να έχουμε τις επιθυμητές τιμές κόστους αλλά και στον επιθυμητό χρόνο.

Η μεγάλη διαφορά ανάμεσα στη μέγιστη και την ελάχιστη τιμή της τελικής βέλτιστης λύσης είναι αποτέλεσμα της γεννήτριας που παράγει τα δεδομένα του προβλήματος με τυχαίο τρόπο. Πιο συγκεκριμένα επειδή σε όλες τις εφαρμογές καμία διαδρομή δεν έμενε ακάλυπτη το κόστος που προκύπτει είναι αποτέλεσμα της υπέρβασης των συνολικών ωρών πτήσης αλλά και των ρεπό που δεν ικανοποιήθηκαν. Έτσι σε ένα πρόβλημα τα δεδομένα που εισήχθησαν (με τυχαίο τρόπο, μέσω αυτής της γεννήτριας) μπορεί να είχαν συνολικό άθροισμα ωρών πτήσης μικρότερο από ένα άλλο, ή σε ένα άλλο πρόβλημα πολύ μεγαλύτερο. Στο πρώτο λοιπόν θα ήταν ευκολότερη η κατανομή των διαδρομών και η βέλτιστη λύση καλύτερη, από ότι στο δεύτερο που θα υπήρχε αυτή η υπέρβαση των ωρών πτήσης. Δηλαδή το penalty για την υπέρβαση των ωρών πτήσης είναι αυτό που δημιουργεί αυτήν την μεγάλη διασπορά και αυτό γιατί η τιμή του είναι αρκετά υψηλή. Αν για παράδειγμα η τιμή του penalty ήταν 1000 μονάδες αντί για 100.000 η διασπορά θα ήταν πολύ μικρότερη.

Ας δούμε τώρα τι επίδραση είχε η αύξηση της τιμής του paths στο χρόνο επίλυσης της εφαρμογής.



Σχήμα 6.4: Επίδραση χρόνου στις διάφορες τιμές paths.

Όπως αναφέραμε και παραπάνω οποιαδήποτε αύξηση στην τιμή των ιπταμένων, των διαδρομών αλλά και του paths οδηγεί στην αύξηση του χρόνου επίλυσης του προβλήματος. Αυτό γίνεται κατανοητό και από το παραπάνω γράφημα όπου ο χρόνος επίλυσης για paths=1 κυμαίνατε περίπου στα 12 λεπτά ενώ για paths=10 στα 30 λεπτά.

Με βάση όλα τα παραπάνω, μπορούμε πλέον με ασφάλεια να συμπεράνουμε πως ο αλγόριθμος λειτουργεί αποτελεσματικά σε διαφόρου μεγέθους προβλήματα, βρίσκοντας την τελική βέλτιστη λύση.

6.2 Σύνοψη

Σκοπός αυτής της διπλωματικής εργασίας ήταν η ανάπτυξη ενός αλγορίθμου, ο οποίος είχε σαν στόχο την κάλυψη ενός συνόλου διαδρομών από ένα σύνολο ιπταμένων με ταυτόχρονη ικανοποίηση των επιθυμιών τους για ανάπαυση.

Στο πρώτο και δεύτερο κεφάλαιο έγινε αναφορά στον τρόπο λειτουργίας των αεροπορικών εταιριών σε σχέση με την ανάθεση των αεροσκαφών, την δρομολόγηση των διαδρομών, την ανάθεση των ιπταμένων καθώς και την εργασιακή ικανοποίηση των ιπταμένων.

Στο τρίτο κεφάλαιο αναπτύχθηκε το μαθηματικό μοντέλο το οποίο περιγράφει το πρόβλημα που μελετήθηκε. Ενώ στο τέταρτο κεφάλαιο αναπτύχθηκαν οι μαθηματικές εξισώσεις του αντίστοιχου προβλήματος καθώς και ένα αριθμητικό παράδειγμα.

Το πέμπτο κεφάλαιο είναι και το πιο σημαντικό αυτής της εργασίας καθώς αποτελεί το πειραματικό μέρος. Σε αυτό έγιναν διάφορες δοκιμές ως προς τον αριθμό των διαδρομών, τον αριθμό των ιπταμένων αλλά και τον αριθμό των μεταβλητών που εισάγονται ανά επανάληψη (παράμετρος paths).

Στο παρόν και τελευταίο κεφάλαιο γίνεται η συλλογή αλλά και η ανάλυση των αποτελεσμάτων που προέκυψαν από το κεφάλαιο 5 και εξάγονται τα συμπεράσματα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Booz-Allen & Hamilton Report. 2001, Punctuality: *How Airlines Can Improve On Time Performance*. : p.1-13

- Hanif D. Sherali*, Ebru K. Bish, Xiaomei Zhu. 2005: *Airline fleet assignment concepts, models and algorithms*. : p.1-30

- Cynthia Barnhart, Barry Smith. 2000: *Quantitative Problem Solving Methods in the Airline Industry*. International Research & Management Science, Dallas, USA.

- George Kozanidis. (2017): *Branch and price for covering shipments in a logistic distribution network with a fleet of aircraft* System Optimization Laboratory, Department of Mechanical Engineering, University of Thessaly, Volos, Greece.

- Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W.P. Savelsbergh, Pamela H. Vance. (1996):* Branch and price: Column Generation for Solving Huge Integer Programs*.Atlanta, Georgia Institute of Technology. : p. 316-329.

- Maximilian M. Etschmater: University of Lowell, Lowell of Massachusetts, Dennis F. Mathaisel: Massachusetts Institute of Technology. (1985): *Airline scheduling: an overview* : p.127-134

- Niklas Kohl, Carmen consulting, Copenhagen, Denmark. Stefan E. Karisch, Carmen Systems Ltd, 1800 McGill College Avenue, Montreal. (2004):* Airline crew rostering: Problem types, modeling and optimization*.: p. 223-257

- E. L. Lawler, D.E. Wood: Published online (1966): *Branch and bound methods: A survey*. University of Michigan: p.699-719

- Martin Savelsbergh (1994) : *A branch and price algorithm for the generalized assignment problem*.Georgia Institute of Technology school of Industrial and Systems Engineering. Atlanta, USA. :p.1172-1184

- R.G.Shenoi (1994): *Solving the long haul crew pairing problem*. Department of civil and environmental engineering, Massachusetts Institute of technology.

- Douglas Potter, 2008 * An implementation of a constraint branching algorithm for optimally solving airline crew pairing problems*, Department of mathematical sciences, Chalmers university of technology and Gothenburg university, Sweden.

ΠΑΡΑΡΤΗΜΑ 1

N=10 / R=100 / Paths=1						
	Λύση MP	**Τελική βέλτιστη λύση**	**GAP**	**Branching**	**Time**	**Uncovered**
1	60003	123000	62997	8	2,447	0
2	64664	129000	64336	8	2,450	0
3	29446	126000	96554	9	2,326	0
4	41020	121000	79980	9	2,304	0
5	127724	231000	103276	9	2,504	0
6	172742	332000	159258	8	2,760	0
7	23696	121000	97304	9	2,941	0
8	47037	126000	78963	9	2,863	0
9	9500	112000	102500	8	4,320	0
10	82550	127000	4450	7	3,010	0
11	65925	128000	62075	8	2,866	0
12	173361	216000	42639	9	2,994	0
13	157767	231000	73233	8	2,730	0
14	151309	220000	68691	9	2,646	0
15	142577	227000	84423	9	4,029	0
16	104571	127000	22429	9	3,082	0
17	36655	119000	82345	9	4,070	0
18	102809	122000	19191	8	3,046	0
19	4086	15000	10914	9	4,404	0
20	147494	327000	179506	9	2,786	0

ΠΑΡΑΡΤΗΜΑ 2

N=45 / R=550 / Paths=1						
	Λύση MP	**Τελική βέλτιστη λύση**	**GAP**	**Branching**	**Time**	**Uncovered**
1	1359821	1720000	360179	44	497,731	0
2	1362407	1607000	244593	43	787,553	0
3	1410209	1597000	186791	43	631,091	0
4	1170580	1510000	339420	44	730,331	0
5	1234358	1315000	80642	42	567,415	0
6	1289039	1619000	329961	44	809,128	0
7	1439117	1732000	292883	44	853,204	0
8	1463264	1737000	273736	44	872,963	0
9	1340406	1506000	165594	44	1090,159	0
10	1310294	1623000	312706	43	884,824	0
11	1300354	1518000	217646	43	793,383	0
12	1568944	1823000	254056	44	744,743	0
13	1361153	1528000	166847	43	560,765	0
14	1055310	1306000	250690	44	701,037	0
15	1264721	1525000	260279	43	605,055	0
16	1232933	1526000	293067	44	610,87	0
17	1233706	1412000	178294	43	749,547	0
18	1315338	1617000	3016622	44	515,659	0
19	1286946	1508000	221054	42	555,308	0
20	1445046	1619000	173954	44	738,583	0

ΠΑΡΑΡΤΗΜΑ 3

N=70 / R=700 / Paths=1

	Λύση MP	Τελική βέλτιστη λύση	GAP	Branching	Time	Uncovered
1	666204	754000	87796	69	5889,442	0
2	468747	562000	93253	67	8150,831	0
3	766005	848000	81995	69	6487,369	0
4	389643	421000	31357	69	5904,717	0
5	554305	656000	101695	69	6791,582	0
6	587178	759000	171822	69	5884,134	0
7	600661	740000	139339	69	5682,512	0
8	576440	655000	78560	69	5570,982	0
9	484689	546000	61311	67	7722,774	0
10	456147	648000	191853	69	5464,58	0
11	383605	429000	45395	69	5329,669	0
12	520891	551000	30109	68	6804,794	0
13	307931	449000	141069	69	5229	0
14	596650	643000	46350	69	4840,426	0
15	460717	652000	191283	69	6504,457	0
16	534680	762000	227320	69	4905,633	0
17	527123	558000	30877	68	5634,961	0
18	312866	324116	11250	67	5985,32	0
19	298564	317896	19332	68	5256,27	0
20	648321	830000	181679	68	6271,151	0

ΠΑΡΑΡΤΗΜΑ 4

N=65 / R=550 / Paths=1

	Λύση MP	Τελική βέλτιστη λύση	GAP	Branching	Time	Uncovered
1	2360	162000	159640	64	602,32	0
2	4473	157000	152527	64	588,345	0
3	9907	166000	156093	63	826,128	0
4	6623	157000	150377	63	648,785	0
5	6550	9000	2450	64	688,742	0
6	7788	142000	134212	64	1090,45	0
7	7133	156000	148867	63	644,908	0
8	5677	152000	146323	64	688,476	0
9	7911	115000	107089	63	648,07	0
10	8886	335000	326114	63	746,19	0
11	5076	152000	146924	64	846,194	0
12	8641	274000	265359	64	741,417	0
13	6981	153000	146019	64	639,149	0
14	7151	7000	-151	64	833,891	0
15	14229	16000	1771	64	920,054	0
16	10678	254000	243322	64	797,407	0
17	4298	175000	170702	64	685,919	0
18	2819	13000	10181	61	488,13	0
19	7765	116000	108235	64	822,213	0
20	6721	109000	102279	64	647,497	0