ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΒΙΟΜΗΧΑΝΙΑΣ

Διπλωματική Εργασία

"Εύρεση αποτελεσματικών λύσεων σε προβλήματα μεικτού ακέραιου πολυκριτήριου προγραμματισμού με τη χρήση του λογισμικού πολυκριτήριας γραμμικής βελτιστοποίησης ADBASE"

Λαζαρίδης Αναστάσιος

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

Φεβρουάριος 2008

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας 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 Λέκτορας, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Λυμπερόπουλος Γεώργιος Αναπληρωτής Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Δρ. Παπαδημητρίου Κωνσταντίνος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών Βιομηχανίας, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Λέκτορα του Τμήματος Μηχανολόγων Μηχανικών Βιομηχανίας Κοζανίδη Γεώργιο, για την ανάθεση της παρούσας εργασίας και για την καθοδήγηση που μου προσέφερε, καθώς και για την πολύτιμη βοήθεια, προσφορά και συνεργασία του.

Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διπλωματικής εργασίας μου, Αναπληρωτή Καθηγητή κ. Λυμπερόπουλο Γεώργιο και Καθηγητή κ. Παπαδημητρίου Κωνσταντίνο για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους.

Τέλος, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στο μεταπτυχιακό φοιτητή Πανταζή Σαράντη για την πολύτιμη βοήθειά του κατά τη συγγραφή της παρούσας εργασίας.

Περιεχόμενα

1.Εισαγωγή

1.1 Η πολυκριτήρια βελτιστοποίηση

Κατά τη διαδικασία της ανάλυσης και λήψης αποφάσεων ερχόμαστε αντιμέτωποι με προβλήματα προγραμματισμού. Αυτά τα προβλήματα ταξινομούνται σε κατηγορίες ανάλογα με τα χαρακτηριστικά που εμφανίζουν. Με βάση τον αριθμό των κριτηρίων, δηλαδή των αντικειμενικών συναρτήσεων, ο διαχωρισμός τους γίνεται σε μονοκριτήρια, όταν έχουμε την ύπαρξη μίας αντικειμενικής συνάρτησης, και σε πολυκριτήρια, όταν οι αντικειμενικές συναρτήσεις είναι δύο ή περισσότερες. Με αυτό τον τρόπο, στα προβλήματα μονοκριτήριας βελτιστοποίησης αναζητούμε τη βέλτιστη μέγιστη ή ελάχιστη τιμή για το κριτήριο που εξετάζουμε λαμβάνοντας υπόψη τους περιορισμούς που συνοδεύουν το πρόβλημα, ενώ στα πολυκριτήρια δεν υπάρχει μία αντικειμενική συνάρτηση προς βελτιστοποίηση αλλά πολλές.

Η διαφορά στη μορφοποίηση των προβλημάτων είναι ορατή παρακάτω.

Ένα πρόβλημα μονοκριτήριου προγραμματισμού (Single Objective Programming - SOP) μορφοποιείται ως εξής:

$$\begin{gathered}\text{Max } \{f(x) = z\} \\ \text{s.t. } x \in S\end{gathered}$$

ενώ σε ένα πολυκριτήριου προγραμματισμού (Multi Objective Programming - MOP) θα έχουμε την ακόλουθη μορφοποίηση:

$$\begin{gathered}\text{Max } \{f_1(x) = z_1\} \\ \vdots \\ \text{Max } \{f_\kappa(x) = z_\kappa\} \\ \text{s.t. } x \in S\end{gathered}$$

όπου f(x) είναι η αντικειμενική συνάρτηση και S το σύνολο των εφικτών λύσεων του προβλήματος.

Όταν το σύνολο των αντικειμενικών συναρτήσεων f(x) είναι γραμμικό και το πεδίο ορισμού S αποτελείται από γραμμικούς περιορισμούς, τότε και το πρόβλημά είναι γραμμικό (Linear-SOLP ή MOLP αντίστοιχα). Σε κάθε άλλη περίπτωση το πρόβλημα είναι μη γραμμικό. Το πρόβλημα του σχεδιασμού πτήσεων και συντήρησης αεροσκαφών που παρουσιάζεται και επιλύεται σε παρακάτω κεφάλαιο είναι γραμμικό, και πιο συγκεκριμένα μεικτό ακέραιο (Mixed Integer Multiobjective). Αυτό σημαίνει πως περιέχει και συνεχείς και ακέραιες μεταβλητές. Αυτό θα φανεί παρακάτω, στη μορφοποίηση του προβλήματος όπου οι περιορισμοί ακεραιότητας αυτόματα εντάσσουν το πρόβλημα σε αυτά του πολυκριτήριου ακέραιου προγραμματισμού.

1.2 Παραδείγματα πολυκριτήριας βελτιστοποίησης

Υπάρχουν πολλά παραδείγματα πολυκριτήριας βελτιστοποίησης με τα οποία ερχόμαστε καθημερινά σε επαφή, είτε στο χώρο της εργασίας μας, είτε σε κάποιες απλές αποφάσεις που καλούμαστε να λάβουμε. Ακολουθούν μερικά παραδείγματα:

-Αγορά ενός προϊόντος (μια τηλεόραση)
με κριτήρια:
Min{κόστος}
Max{ποιότητα}
Max{αριθμός χρωμάτων}
Max{έτη εγγύησης}
και όποια άλλα μας ενδιαφέρουν

-Επένδυση σε μια μετοχή
με κριτήρια:
Min {ρίσκο}
Max {απόδοση}
Max {ρευστότητα}
και διάφορα άλλα

-Σχεδιασμός ενός κινητήρα
με κριτήρια:
Min {βάρος}
Max {ιπποδύναμη}
Min {κατανάλωση}
Min {διαστάσεις}
Max {διάρκεια ζωής}

Αυτά είναι ελάχιστα μόνο από τα άπειρα παραδείγματα τέτοιου είδους που προκύπτουν από τη βιομηχανική δραστηριότητα, από τις επιχειρήσεις, ακόμα και από τις καθημερινές μας δραστηριότητες, για τα οποία η τελική λήψη των αποφάσεων ανάγεται στην επίλυση ενός προβλήματος πολυκριτήριου προγραμματισμού.

1.3 Βιβλιογραφική Ανασκόπηση

Η βιβλιογραφία αναφορικά με την επίλυση μονοκριτήριων προβλημάτων γραμμικής βελτιστοποίησης έχει αναπτυχθεί τα τελευταία 60 περίπου χρόνια. Υπάρχουν εκτεταμένες μελέτες και αρκετές μέθοδοι γύρω από την επίλυση τέτοιων προβλημάτων, με αποτέλεσμα να είναι ένα αντικείμενο γνωστό σε βάθος.

Ωστόσο, η πλειονότητα των σύγχρονων προβλημάτων βελτιστοποίησης περιλαμβάνει περισσότερα από ένα κριτήρια, ανάγοντας την πολυκριτήρια βελτιστοποίηση (είτε αυτή είναι γραμμική είτε όχι) σε μία από τις πιο ενδιαφέρουσες και ταυτόχρονα σημαντικές περιοχές της Διοικητικής Επιστήμης.

Από άποψη μεθοδολογιών, την τελευταία δεκαετία έχουμε παρακολουθήσει σημαντική πρόοδο στη ανάπτυξη αλγορίθμων και τεχνικών επίλυσης. Τα μοντέλα βελτιστοποίησης είναι πιο πρακτικά και επιλύονται ευκολότερα. Σημαντικά μεγαλύτερα και δυσκολότερα προβλήματα έχουν επιλυθεί σε πολύ μικρότερο υπολογιστικό χρόνο.

Ο Steuer (17) αναπτύσσει υπολογιστικές μεθόδους για την επίλυση προβλημάτων MOP με εκτενείς αναφορές στη θεωρία και το μαθηματικό υπόβαθρο γύρω από την πολυκριτήρια βελτιστοποίηση.

Ο Ehrgott (6) προσπαθεί να προβάλλει θεωρητικές ερωτήσεις όπως η ύπαρξη ή μη λύσεων σε πολυκριτήρια προβλήματα και ταυτόχρονα να αναπτύξει μεθοδολογίες για την επίλυσή τους.

Ο Steuer (18) και (19) έχει αναπτύξει ένα λογισμικό επίλυσης γραμμικών πολυκριτήριων προβλημάτων (την ADBASE) και παρουσιάζει μια πλειάδα λυμένων προβλημάτων.

Τέλος, ο Kozanidis (14) μελετά μία επέκταση του Γραμμικού Προβλήματος Σακιδίου Πολλαπλών Επιλογών (Linear Multiple Choice Knapsack Problem) εξετάζοντας δύο αντικειμενικές συναρτήσεις.

1.4 Αποτελεσματικές και βέλτιστες λύσεις

Εκ των πραγμάτων, είναι πολύ σπάνια η περίπτωση που σε ένα MOLP πρόβλημα μια λύση θα βελτιστοποιεί ταυτόχρονα όλα τα κριτήρια. Στην πλειονότητα των προβλημάτων παίρνουμε ένα πλήθος από λύσεις, καθεμιά από τις οποίες βελτιστοποιεί ένα από τα κριτήρια που έχουμε θέσει. Με αυτό τον τρόπο η λήψη της απόφασης πραγματοποιείται με βάση τις προτεραιότητες που θέτει ο χρήστης ή τις απαιτήσεις του εκάστοτε προβλήματος.

Μια λύση για να είναι αποδεκτή πρέπει να είναι αρχικά αποτελεσματική ("efficient"). Αποτελεσματική είναι μια λύση η οποία δεν κυριαρχείται από κάποια άλλη (είναι δηλαδή "nondominated"). Αυτό σημαίνει πως δεν υφίσταται άλλη εφικτή λύση η οποία βελτιώνει την τιμή έστω και ενός από τα κριτήρια χωρίς αυτό να γίνει σε βάρος τουλάχιστον ενός από τα υπόλοιπα κριτήρια. Όπως γίνεται αντιληπτό οι λύσεις που δεν είναι αποτελεσματικές δεν μπορούν να ελεγχθούν για βελτιστότητα.

Εκτός από τις αποτελεσματικές λύσεις, υπάρχει μια κατηγορία λύσεων που ονομάζονται weakly efficient και αντίστοιχα weakly nondominated. Μια λύση ονομάζεται weakly efficient όταν δεν υπάρχει κάποια διαφορετική λύση η οποία να είναι αυστηρά καλύτερη όσον αφορά σε όλα τα κριτήρια.

Στο τέλος, θα επιλέξουμε ως βέλτιστη λύση (σε περίπτωση που έχουμε περισσότερες από μία αποτελεσματικές) αυτή που μας ικανοποιεί περισσότερο, ανάλογα με το κριτήριο στο οποίο δίνουμε τη μεγαλύτερη βαρύτητα. Αυτό όπως εξηγήσαμε συμβαίνει γιατί είναι εξαιρετικά σπάνια η περίπτωση κατά την οποία μια λύση θα βελτιστοποιεί όλα τα κριτήρια ταυτόχρονα.

1.5 Μέθοδοι επίλυσης προβλημάτων πολυκριτήριου προγραμματισμού

Ο πλέον παραδοσιακός τρόπος προσέγγισης προβλημάτων πολυκριτήριου προγραμματισμού είναι μορφοποιώντας ένα μονοκριτήριο πρόβλημα βελτιστοποίησης με τέτοιο τρόπο ώστε να είναι συσχετισμένο με το αρχικό. Αυτός ο συσχετισμός προκύπτει μέσω της δημιουργίας μιας νέας αντικειμενικής συνάρτησης η οποία πρακτικά είναι συνάρτηση των κριτηρίων του αρχικού προβλήματος, βοηθητικών μεταβλητών ή βοηθητικών παραμέτρων. Ελεύθερα, θα μπορούσαμε να χαρακτηρίσουμε αυτές τις τεχνικές ως τεχνικές "αποκλιμάκωσης", αφού μετατρέπουν το αρχικό πολυκριτήριο πρόβλημα σε μονοκριτήριο. Η αντίστοιχη ξένη ορολογία είναι "scalarization". Στις περιπτώσεις

που εισάγονται στη νέα συνάρτηση βοηθητικές μεταβλητές, αντίστοιχα θα γίνει προσθήκη νέων περιορισμών που θα σχετίζονται με αυτές.
Κάποιες από τις πιο γνωστές μεθόδους "αποκλιμάκωσης" είναι οι παρακάτω:

α) Η μέθοδος των σταθμισμένων συντελεστών (weighted sums approach)

β) Η μέθοδος του περιορισμού του συνόλου των εφικτών λύσεων (e-constraint method)

γ) Η υβριδική μέθοδος (Hybrid method – συνδυασμός της μεθόδου των σταθμισμένων συντελεστών και της e-constraint)

δ) Η μέθοδος των ελαστικών περιορισμών (elastic constraints method)

ε) Η μέθοδος του Benson

με πιο διαδεδομένη και αυτή που θα μας απασχολήσει να είναι αυτή των σταθμισμένων συντελεστών.

1.6 Η μέθοδος των Σταθμισμένων Συντελεστών (Weighted Sums Approach)

Παραπάνω είδαμε πως κατά την επίλυση προβλημάτων MOLP μια μέθοδος που συναντάμε πολύ συχνά είναι αυτή των σταθμισμένων συντελεστών. Η εν λόγω μέθοδος λειτουργεί ως εξής: Κάθε αντικειμενική συνάρτηση πολλαπλασιάζεται με έναν αυστηρά θετικό συντελεστή w_i και δημιουργείται η νέα αντικειμενική συνάρτηση από το άθροισμα όλων των γινομένων. Γράφοντας πλέον το πρόβλημα σε μητρωική μορφή παίρνουμε μια μορφοποίηση του τύπου max (ή min) $\mathbf{w}^T \times \mathbf{C} \times \mathbf{x}$ όπου **C** ο πίνακας διαστάσεων n × n των συντελεστών c_i της κάθε μεταβλητής απόφασης, **x** ο πίνακας διαστάσεως n × 1 των μεταβλητών απόφασης x_j και $\mathbf{w}^T$ ο πίνακας διαστάσεως 1 × n των σταθμισμένων συντελεστών (χρησιμοποιούμε τον ανάστροφο του αρχικού **w**).
Το άθροισμα των συντελεστών βαρύτητας (όπως αλλιώς λέγονται οι σταθμισμένοι συντελεστές) πρέπει απαραίτητα να είναι ίσο με τη μονάδα, δηλαδή $\sum_{i=1}^{k} w_i = 1$.

Για παράδειγμα, έστω ότι έχουμε το παρακάτω πολυκριτήριο πρόβλημα:

$$\text{Max}\ \{5x_1 + x_2\}$$

$$\text{Max}\ \{-x_1 + 4x_2\}$$

$$\text{s.t. } x \in S$$

Έστω ότι επιλέγουμε συντελεστές 0.8 και 0.2 αντίστοιχα για τις δύο συναρτήσεις. Σε μορφή πινάκων $\mathbf{w}^T \times \mathbf{C} \times \mathbf{x}$ θα έχουμε:

$$(0.8 \quad 0.2)\begin{pmatrix} 5 & 1 \\ -1 & 4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Τελικά μετά από τις πράξεις θα πάρουμε την αντικειμενική συνάρτηση:

$$\max\ \{3.8\ x_1 + 1.6\ x_2 \mid \text{s.t. } x \in S\}$$

Η βέλτιστη τιμή αυτής της συνάρτησης είναι μοναδική (αν το πρόβλημα φυσικά δεν είναι ανέφικτο). Θα συμπεριφέρεται δηλαδή όπως τα SOLP προβλήματα, αλλά ταυτόχρονα θα έχει και τα χαρακτηριστικά του αρχικού πολυκριτήριου προβλήματος που είχαμε χάρη στους συντελεστές βαρύτητας.

Ουσιαστικά, αυτό που πετυχαίνουμε με τη χρήση της μεθόδου των σταθμισμένων συντελεστών είναι να επιλύουμε ένα μονοκριτήριο γραμμικό πρόβλημα (ενώ αρχικά είχαμε περισσότερες από μία αντικειμενικές συναρτήσεις), η επίλυση του οποίου, πέραν του ότι είναι σημαντικά πιο εύκολη και λιγότερο χρονοβόρα, θα μας δώσει μία και μόνο βέλτιστη λύση. Αυτή τη λύση μάλιστα, εν μέρει, θα την έχουμε "κατευθύνει" εμείς με τους συντελεστές που έχουμε θέσει, ανάλογα πάντα με τις απαιτήσεις ή τις προτιμήσεις μας.

Το πλέον ενδιαφέρον και αξιοσημείωτο όμως σε όλη αυτή τη διαδικασία, είναι ότι η βέλτιστη λύση που λαμβάνεται με αυτόν τον τρόπο για το SOLP είναι αποτελεσματική για το πρόβλημα MOLP. Επίσης, κάθε αποτελεσματική λύση του MOLP αποτελεί βέλτιστη λύση του SOLP για κατάλληλο συνδυασμό των συντελεστών βαρύτητας.

Όπως έχει λεχθεί, το πρόβλημα του Σχεδιασμού Πτήσεων και Συντήρησης των Αεροσκαφών της Πολεμικής Αεροπορίας που αναλύεται σε επόμενο κεφάλαιο ανήκει στα προβλήματα μεικτού ακέραιου προγραμματισμού. Σε αυτό το πρόβλημα έχουμε τέσσερα κριτήρια (z_1, z_2, z_3, z_4) τα οποία πολλαπλασιάζονται με τους συντελεστές βαρύτητας w_1, w_2, w_3, w_4. Για λόγους αντιστοίχησης των μονάδων των κριτηρίων, έχουμε πολλαπλασιάσει το πρώτο κριτήριο με 150, το τρίτο με 450 και το τέταρτο με 3. Κατά τη "μετατροπή" του προβλήματος σε SOLP, επιλέξαμε το συνδυασμό $w_1=w_2=w_3=w_4=0.25$. Με αυτό τον τρόπο παίρνουμε ένα μονοκριτήριο πρόβλημα με αντικειμενική συνάρτηση:

$$\text{Max } 0.25*150z_1 + 0.25z_2 + 0.25*450z_3 + 0.25*3z_4$$

2.Λογισμικά επίλυσης προβλημάτων πολυκριτήριου προγραμματισμού

Για τα μεικτά ακέραια πολυκριτήρια προβλήματα, το ιδεατό θα ήταν να χρησιμοποιήσουμε ένα λογισμικό το οποίο θα έλυνε τέτοιου είδους προβλήματα. Όμως η μη ύπαρξη τέτοιου προγράμματος, μας αναγκάζει να καταφύγουμε στη συνδυαστική χρήση δύο λογισμικών, της ADBASE και της AMPL. Το πρώτο αποτελεί λογισμικό πολυκριτήριας γραμμικής βελτιστοποίησης ενώ το δεύτερο έχει τη δυνατότητα επίλυσης μονοκριτήριων μεικτών ακέραιων προβλημάτων. Ο συνδυασμός τους μας δίνει λύσεις για τη γραμμική χαλάρωση του πολυκριτήριου προβλήματος και για το μεικτό ακέραιο μονοκριτήριο, τις οποίες θα συγκρίνουμε για να διαπιστώσουμε αν υπάρχουν λύσεις που ταυτίζονται.

Φυσικά, έχουν αναπτυχθεί διάφορα λογισμικά επίλυσης προβλημάτων πολυκριτήριου προγραμματισμού εκτός από αυτά που χρησιμοποιούνται σε αυτή την εργασία. Κάποια από αυτά αναφέρονται παρακάτω:

NIMBUS (http://nimbus.mit.jyu.fi/)

Το NIMBUS μπορεί να χρησιμοποιηθεί για την επίλυση διαφορήσιμων και μη διαφορήσιμων, πολυκριτήριων και μονοκριτήριων προβλημάτων βελτιστοποίησης, τα οποία έχουν γραμμικούς και μη γραμμικούς περιορισμούς. Δεν είναι γενικό (generic), δηλαδή δεν λύνει όλα τα προβλήματα. Έχει αναπτυχθεί στο πανεπιστήμιο της Jyväskylä στη Φιλανδία, από το τμήμα Mathematical Information Technology.

MOMHLib++ (MultiObjective MetaHeuristics Library in C++) (http://www-dss.cs.put.poznan.pl/%7Ejaszkiewicz/MOMHLib/)

Το MOMHLib++ είναι μια βιβλιοθήκη από προγράμματα της γλώσσας προγραμματισμού C++ που αποτελούν έναν αριθμό από μεταευρετικούς αλγόριθμους πολυκριτήριου προγραμματισμού. Οι στόχοι της δημιουργίας αυτής της βιβλιοθήκης μπορούν να συνοψιστούν στους εξής:

- να υλοποιήσει τον κύριο ευρετικό αλγόριθμο για πολυκριτήρια προβλήματα με ένα συνεπή τρόπο
- να επιτρέψει την εύκολη υιοθέτηση της μεθόδου από ένα δεδομένο πρόβλημα

- να βοηθήσει στην αποτίμηση της εφαρμογής ενός ευρετικού αλγορίθμου σε ένα δεδομένο πρόβλημα
- να επιτρέψει την εύκολη ενσωμάτωση νέων ευρετικών αλγορίθμων για πολυκριτήρια προβλήματα και να βοηθήσει την ανάπτυξη καινούριων μεθόδων

RGDB (http://www.ccas.ru/mmes/mmeda/rgdb/index.htm)

Το RGDB είναι ένα απλό γραφικό εργαλείο το οποίο μπορεί να μας βοηθήσει να αποκλείσουμε ασυνήθιστα αντικείμενα από μεγάλες λίστες.
Λειτουργεί μέσω του παγκόσμιου ιστού με πολύ απλό τρόπο. Υποβάλλουμε μια λίστα από αντικείμενα, η οποία θα αποτελέσει τη βάση δεδομένων του προγράμματος. Έπειτα το πρόγραμμα επεξεργάζεται τα δεδομένα και τελικά επιλέγει έναν αριθμό από αντικείμενα της αρχικής λίστας που ανταποκρίνονται στις προτιμήσεις μας.

The Athena Software (http://www.athenasoft.org)

Το λογισμικό Athena αποτελείται από δύο πακέτα: το Athena standard και το Athena negotiator. Το Athena standard έχει σχεδιαστεί για να υποστηρίξει το συλλογισμό και την επιχειρηματολογία ενώ το Athena negotiator έχει σχεδιαστεί για να διευκολύνει την ανάλυση αποφάσεων και τις διαπραγματεύσεις δύο ομάδων. Κατά μια έννοια, τα δύο αυτά λογισμικά είναι η έκβαση του συνδυασμού της φιλοσοφικής και της μηχανολογίας.

IRIS – VIP (http://www4.fe.uc.pt/lmcdias/english/software.htm)

Το λογισμικό IRIS (Interactive Robustness analysis and parameters' Inference for multicriteria Sorting problems) χρησιμοποιεί μια απαισιόδοξη εκδοχή της μεθόδου Electre Tri [20], ενώ η ανάλυση VIP (Variable Interdependent Parameters Analysis) ενσωματώνει πολλαπλά εργαλεία ανάλυσης για περιπτώσεις που ενδιαφερόμενοι δεν έχουν τη δυνατότητα (ή δεν επιθυμούν) να θέσουν ακριβείς τιμές για τις παραμέτρους.

Υπάρχουν και άλλα λογισμικά για προβλήματα πολυκριτήριου προγραμματισμού για τα οποία μπορούμε να αντλήσουμε πληροφορίες από την εξής ηλεκτρονική διεύθυνση:

www.mit.jyu.fi/MCDM/soft.html13

3.Η ADBASE

Όπως αναφέρθηκε και σε προηγούμενη παράγραφο, το λογισμικό πολυκριτήριας γραμμικής βελτιστοποίησης που χρησιμοποιήθηκε στο πρόβλημα του σχεδιασμού των πτήσεων και συντήρησης των αεροσκαφών είναι η ADBASE.

3.1 Λειτουργία του λογισμικού

Η λειτουργία του λογισμικού είναι εξαιρετικά απλή. Όλα τα στοιχεία για να "τρέξει" η ADBASE περιλαμβάνονται σε ένα φάκελο με την ονομασία Α06 τον οποίο αντιγράφουμε στο σκληρό δίσκο του υπολογιστή (ακριβώς "κάτω" από το δίσκο). Αυτός ο φάκελος έχει με τη σειρά του τρεις υποφακέλους με ονόματα Α, F, L. Τους φακέλους F και L δεν τους πειράζουμε καθόλου. Στον Α θα αντιγράψουμε τα αρχεία που χρειαζόμαστε για να δουλέψει το λογισμικό. Αυτά τα αρχεία είναι δύο, γραμμένα σε σημειωματάριο (notepad) με επεκτάσεις *.qq και *.ii. Για παράδειγμα, θα μπορούσαν να είναι fmp.qq και fmp.ii.

3.2 Σχηματισμός του αρχείου *.qq

Αυτό το αρχείο δίνει στην ADBASE κάποιες απαραίτητες πληροφορίες για το πρόβλημα που θέλουμε να επιλύσουμε. Είναι διάφορες επιλογές του χρήστη. Αν για παράδειγμα επιθυμούμε η ADBASE να αναζητήσει μόνο τα αποτελεσματικά ακραία σημεία (efficient extreme points) ή και τα weakly-efficient extreme points, ή αν θέλουμε να σταματήσουν ή να συνεχίσουν οι υπολογισμοί μετά την εύρεση του πρώτου αποτελεσματικού σημείου προβαίνουμε στις ανάλογες δηλώσεις στο αρχείο *.qq.

Συνολικά πρέπει να καθοριστούν 47 επιλογές, οι οποίες αναφέρονται διεξοδικά στο εγχειρίδιο της ADBASE. Ουσιαστικά αντιγράφουμε τις ονομασίες των παραμέτρων από το εγχειρίδιο και τους δίνουμε την τιμή που επιθυμούμε.

3.3 Σχηματισμός του αρχείου *.ii

Πριν προχωρήσουμε στην κατασκευή του αρχείου πρέπει να προηγηθούν κάποιες ενέργειες. Αρχικά αριθμούμε τις μεταβλητές απόφασης καθώς και τους περιορισμούς του προβλήματος με αύξουσα σειρά. Αν για παράδειγμα οι μεταβλητές που εμφανίζονται στις αντικειμενικές συναρτήσεις και στους περιορισμούς είναι οι x, y, z, w, q τότε θα έχουμε τους αριθμούς 1, 2, 3, 4, 5 αντίστοιχα για την κάθε μεταβλητή. Με τον ίδιο τρόπο αριθμούνται και οι περιορισμοί. Ιδιαίτερη προσοχή χρειάζεται το γεγονός ότι οι περιορισμοί πρέπει να διαχωριστούν ανάλογα με τη φορά τους σε περιορισμούς ≤, περιορισμούς = και περιορισμούς ≥ με αυτή τη σειρά που αναφέρθηκε. Επιπρόσθετα, τα δεξιά μέλη (σταθεροί όροι) των περιορισμών πρέπει να είναι μεγαλύτερα ή ίσα του μηδενός. Αν είναι μικρότερα, ο περιορισμός θα πρέπει να αλλάξει φορά ενώ αν είναι ίσα με το μηδέν απλώς δε δηλώνονται.

Έπειτα, κατασκευάζουμε το δεύτερο αρχείο που χρειαζόμαστε, αυτό με την επέκταση *.ii. Όπως αναφέρθηκε, και αυτό το αρχείο δημιουργείται ομοίως στο σημειωματάριο και οφείλει να έχει την ίδια ονομασία με το αντίστοιχο *.qq (για παράδειγμα fmp.qq και fmp.ii). Η ιδιαιτερότητα που παρουσιάζει το *.ii, είναι το γεγονός ότι τα δεδομένα πρέπει να έχουν ακριβή και πλήρη στοίχιση μεταξύ τους για είναι "κατανοητά" από την ADBASE. Οι ακριβείς θέσεις των δεδομένων (των αριθμών δηλαδή) μέσα στο αρχείο αναφέρονται επακριβώς στο εγχειρίδιο της ADBASE.

Ακόμα, υπάρχουν κάποιοι περιορισμοί στις δυνατότητες του λογισμικού. Το άθροισμα του αριθμού των περιορισμών, των γραμμών του πίνακα των κριτηρίων και του αριθμού 1 πρέπει να είναι μικρότερο του 1210. (number of constraints + criterion matrix rows + 1 ≤ 1210). Γι' αυτό το λόγο τα προβλήματα που εξετάζουμε είναι συγκεκριμένου μεγέθους.

Σε προβλήματα μικρής έκτασης, όπου τα δεδομένα για την ADBASE είναι λίγα και προκαθορισμένα, είναι εύκολη η απευθείας πληκτρολόγησή τους στο αρχείο. Εν τούτοις, σε μεγάλα προβλήματα αυτό δεν ενδείκνυται γιατί είναι ιδιαίτερα χρονοβόρο και επίπονο. Αν δε και τα δεδομένα είναι δυναμικά, (αλλάζει δηλαδή ο αριθμός των μεταβλητών απόφασης και των περιορισμών του προβλήματος, άρα και το μέγεθός του), η απευθείας πληκτρολόγησή τους είναι πρακτικά αδύνατη.

Σε αυτές τις περιπτώσεις, στις οποίες συγκαταλέγεται και το πρόβλημα του σχεδιασμού πτήσεων και συντήρησης των αεροσκαφών που επιλύεται παρακάτω, το αρχείο δημιουργείται αποκλειστικά με τη βοήθεια κώδικα σε κάποια γλώσσα προγραμματισμού.

Για το συγκεκριμένο πρόβλημα έγινε χρήση δύο γλωσσών προγραμματισμού, της FORTRAN και της C++. Και οι δύο κώδικες βρίσκονται πλήρως ανεπτυγμένοι στο παράρτημα.

Τα δεδομένα τοποθετούνται στο αρχείο *.ii με την παρακάτω σειρά:

Τίτλος του προβλήματος

Ο τίτλος του προβλήματος αναγράφεται στην πρώτη γραμμή, στις στήλες 1 ως 70. Μπορούμε να δώσουμε τον τίτλο που θέλουμε στο πρόβλημα. Στο συγκεκριμένο είναι "FMP problem with M=* N=* T=*". Ο αστερίσκος δηλώνει ότι οι παράμετροι M, N, T δεν είναι σταθερές αλλά αλλάζουν ανάλογα με το εκάστοτε πρόβλημα που θέλουμε να επιλύσουμε.

Κύριες παράμετροι του προβλήματος

Στη δεύτερη γραμμή αναγράφονται 8 παράμετροι που περιγράφουν το πρόβλημα με στοίχιση από δεξιά 8I8 (δηλαδή η πρώτη στη στήλη 8, η δεύτερη στη στήλη 16 και ούτω καθ' εξής).
Αυτές οι παράμετροι είναι αντίστοιχα:

1. αριθμός του προβλήματος (δεν μπορεί να υπερβαίνει το 99999999
2. αριθμός αντικειμενικών συναρτήσεων (κριτήρια του προβλήματος
3. αριθμός των μεταβλητών απόφασης του προβλήματος
4. αριθμός περιορισμών ≤
5. αριθμός περιορισμών ισότητας (=)
6. αριθμός περιορισμών ≥
7. αυτή η παράμετρος (IFASE0) έχει πάντα τιμή 0 στο συγκεκριμένο πρόβλημα
8. αυτή η παράμετρος (NGRAYS) δεν έχει σημασία ποια τιμή θα πάρει εφόσον η IFASE0 είναι διάφορη της μονάδας

Πίνακες μη μηδενικών συντελεστών

Τα δεδομένα κάθε προβλήματος περιλαμβάνονται σε 8 συνολικά πίνακες μη μηδενικών συντελεστών $\mathbf{A}_k$, $\mathbf{b}_k$, $\mathbf{A}_e$, $\mathbf{b}_e$, $\mathbf{A}_s$, $\mathbf{b}_s$, **C** και έναν ακόμη πίνακα **a**-constant με σταθερούς όρους.
Οι πίνακες $\mathbf{A}_k$, $\mathbf{A}_e$, $\mathbf{A}_s$ και **C** περιλαμβάνουν τους μη μηδενικούς συντελεστές των μεταβλητών απόφασης που βρίσκονται στα αριστερά μέλη των περιορισμών ≤, =, ≥ και στις αντικειμενικές συναρτήσεις αντίστοιχα. Η στοίχιση των δεδομένων γίνεται σε μορφή 4(2I4,F17.0). Σε κάθε γραμμή ενός πίνακα εμφανίζονται 4 ομάδες αριθμών, αποτελούμενες από 3 αριθμούς η κάθε μία. Ο πρώτος αριθμός της κάθε τριάδας (στοιχισμένος σε I4 όπως είπαμε) υποδεικνύει τον αριθμό του περιορισμού που εξετάζουμε (σύμφωνα με την αρίθμηση που πραγματοποιήσαμε αρχικά). Ο

δεύτερος (επίσης I4) υποδεικνύει τον αριθμό της μεταβλητής απόφασης που εμφανίζεται σε αυτό τον περιορισμό. Τέλος, ο τρίτος αριθμός (με μέγεθος 17 στήλες) είναι ο συντελεστής μπροστά από τη μεταβλητή απόφασης.

Δηλαδή, οι αριθμοί των περιορισμών και των μεταβλητών τοποθετούνται στις στήλες:

4 8 29 33 54 58 79 83

και οι τιμές των μη μηδενικών συντελεστών καθορίζονται στις στήλες:

9-25 34-50 59-75 84-100

Τα δεδομένα από τους πίνακες-διανύσματα $\mathbf{b}_k$, $\mathbf{b}_e$ και $\mathbf{b}_s$ των σταθερών όρων που βρίσκονται στα δεξιά μέλη των περιορισμών, τοποθετούνται στο αρχείο *.ii με παρόμοιο τρόπο. Η διαφορά εντοπίζεται στο γεγονός ότι εδώ δεν έχουμε τριάδες αριθμών αλλά δυάδες, αφού δεν υφίσταται αριθμός μεταβλητής. Οπότε η στοίχισή τους γίνεται στις στήλες:

4 29 54 79

Τέλος, έχουμε και τον πίνακα με τους σταθερούς όρους που υπάρχουν σε κάθε μία από τις αντικειμενικές συναρτήσεις. Τα δεδομένα από αυτόν τον πίνακα τοποθετούνται στο αρχείο ακριβώς με τον ίδιο τρόπο όπως αυτά των πινάκων με τα δεξιά μέλη των περιορισμών.

Η γενική διαμόρφωση του αρχείου φαίνεται στο παρακάτω σχεδιάγραμμα:

Τίτλος του προβλήματος

Κύριες παράμετροι του προβλήματος

Συνολικός αριθμός συντελεστών στους περιορισμούς ≤
{ δεδομένα από πίνακα $\mathbf{A}_k$ }

Συνολικός αριθμός σταθερών όρων στα δεξιά μέλη των περιορισμών ≤
{ δεδομένα από πίνακα-διάνυσμα $\mathbf{b}_k$ }

Συνολικός αριθμός συντελεστών στους περιορισμούς =
{ δεδομένα από πίνακα $\mathbf{A_s}$ }

Συνολικός αριθμός σταθερών όρων στα δεξιά μέλη των περιορισμών ισότητας (=)
{ δεδομένα από πίνακα-διάνυσμα $\mathbf{b_s}$ }

Συνολικός αριθμός συντελεστών στους περιορισμούς ≥
{ δεδομένα από πίνακα $\mathbf{A_e}$ }

Συνολικός αριθμός σταθερών όρων στα δεξιά μέλη των περιορισμών ≥
{ δεδομένα από πίνακα-διάνυσμα $\mathbf{b_e}$ }

Συνολικός αριθμός συντελεστών στις αντικειμενικές συναρτήσεις
{ δεδομένα από πίνακα **C** }

Συνολικός αριθμός σταθερών όρων στις αντικειμενικές συναρτήσεις
{ δεδομένα από πίνακα-διάνυσμα **a**-constant}

Είναι βέβαιο πως όλα τα παραπάνω φαίνονται δυσνόητα χωρίς την ύπαρξη ενός παραδείγματος το οποίο να συνοδεύει τη θεωρία. Στην επόμενη σελίδα παρατίθεται ένα απλό αρχείο *.ii για την επεξήγηση και κατανόηση όσων ειπώθηκαν πριν.

```
FMP Problem with M=1 , N=1 , T=1
  1    4   17   24    5    7    0   40
   45 .LE.LHS.COEFFICIENTS
 1  1      1.000000  1  6     -1.000000  2  2      1.000000  2  8     -1.000000
 3  3      1.000000  3  6     -1.000000  4  4      1.000000  4  8     -1.000000
 5  6     -1.000000  5  5      1.000000  5 13      1.100000  6  5     -1.000000
 6  6      1.000000  6 14      1.100000  7  9      1.000000  8 12      1.000000
 9 15    586.000000  9 12     -1.000000 10 15   -586.000000 10 10      1.000000
10 12     -1.000000 11  7      1.000000 11 16    586.000000 12  6      1.000000
12  7   -586.000000 12  9    586.000000 12 16   -586.000000 13 10      1.000000
13 17    586.000000 14  8      1.000000 14  6   -300.000000 15 11      1.000000
15  6    320.000000 16  9      1.000000 16  5   -170.000000 17  9      1.000000
17  7     -1.000000 18 12      1.000000 18 10     -1.000000 19  6      1.000000
20 13      1.000000 21 14      1.000000 22 15      1.000000 23 16      1.000000
24 17      1.000000
   14 .LE.RHS.COEFFICIENTS
 5         1.000000  6         1.000000  7        89.000000  8       585.000000
 9         1.000000 11       586.000000 13       586.000000 15       320.000000
19         1.000000 20         1.000000 21         1.000000 22         1.000000
23         1.000000 24         1.000000
   11 .EQ.LHS.COEFFICIENTS
 1  8      1.000000  1  7     -1.000000  1  9      1.000000  1 13   -300.000000
 2 11      1.000000  2 10     -1.000000  2 12      1.000000  2 14   -320.000000
 3  5      1.000000  4  7      1.000000  5 10      1.000000
```

```
 2 .EQ.RHS.COEFFICIENTS
3        1.000000 4       281.000000
 16 .GE.LHS.COEFFICIENTS
1 13     1.000000 1 6     -1.000000  1 5      1.000000   2 14     1.000000
2 5     -1.000000 2 6      1.000000  3 9      1.000000   4 6      1.000000
5 6      1.000000 5 10   586.000000  5 12  -586.000000   5 17   586.000000
6 8      1.000000 6 6     -0.100000  7 11     1.000000   7 6      0.100000
 3 .GE.RHS.COEFFICIENTS
3       89.000000 5        1.000000  7        0.100000
 4 .OBJ.LHS.COEFFICIENTS
1 1      1.000000 2 2      1.000000  3 3      1.000000   4 4      1.000000
 0 .OBJ.LHS.CONSTANTS
```

Είναι προφανής ο τίτλος του προβλήματος ($1^η$ γραμμή). Οι αριθμοί που αναγράφονται στη δεύτερη γραμμή μας λένε πως ο αριθμός του προβλήματος είναι 1, το πρόβλημα έχει 4 αντικειμενικές συναρτήσεις, 17 μεταβλητές απόφασης, 24 περιορισμούς ≤, 5 περιορισμούς ισότητας (=), και 7 περιορισμούς ≥. Οι δύο τελευταίοι αριθμοί είναι 0 και 40 σε όλα τα προβλήματα FMP.

Έπειτα βλέπουμε πώς δηλώνονται οι συντελεστές σε κάθε είδος περιορισμού (ο συνολικός τους αριθμός και ποιοι είναι αυτοί). Για παράδειγμα εδώ έχουμε 45 συντελεστές στα αριστερά μέλη των περιορισμών ≤. Στον πρώτο περιορισμό η μεταβλητή που έχει αντιστοιχηθεί στον αριθμό 1 έχει συντελεστή 1 (πρώτη τριάδα αριθμών). Η δεύτερη τριάδα μας λέει πως και πάλι στον πρώτο περιορισμό η μεταβλητή που έχει αντιστοιχηθεί στον αριθμό 6 έχει συντελεστή -1.

Με τον τρόπο που περιγράφηκε παραπάνω, συνεχίζουμε και για τα άλλα είδη περιορισμών.

Για τους σταθερούς συντελεστές που βρίσκονται στα δεξιά μέλη των περιορισμών παρατηρούμε πως υπάρχει μόνο ο αριθμός του περιορισμού και η τιμή του συντελεστή.

Στο τέλος του αρχείου δηλώνονται και οι συντελεστές των όρων των αντικειμενικών συναρτήσεων και οι σταθεροί όροι που υπάρχουν σε αυτές (εδώ είναι 0).

4.Το Πρόβλημα του Σχεδιασμού Πτήσης και Συντήρησης Αεροσκαφών της Πολεμικής Αεροπορίας (Flight and Maintenance Planning problem-FMP)

4.1 Εισαγωγικά

Κάθε αεροσκάφος, είτε πολιτικό είτε πολεμικό, μόλις συμπληρώσει ένα συγκεκριμένο αριθμό ωρών πτήσης από τον τελευταίο έλεγχο στον οποίο υποβλήθηκε, πρέπει να υποβάλλεται εκ νέου σε συντήρηση για λόγους ασφαλείας. Μάλιστα, οι πολιτικές αεροπορικές εταιρίες συνήθως προγραμματίζουν αυτές τις συντηρήσεις να λαμβάνουν χώρα κατά τη διάρκεια της νύχτας ώστε το αεροσκάφος να έχει τις μικρότερες δυνατές απώλειες από το χρόνο τον οποίο προοριζόταν να πετάει.

Για τις πολιτικές αερογραμμές αυτό αποσκοπεί προφανώς στην ασφάλεια των πελατών της και συνεπώς στην αύξηση των κερδών τους. Στην πολεμική αεροπορία, αντικειμενικός σκοπός αυτού του σχεδιασμού είναι η μεγιστοποίηση της ετοιμότητας των αεροσκαφών, ώστε να μπορούν να ανταποκριθούν άμεσα σε οποιαδήποτε εξωτερική απειλή. Αυτή η διαφορά είναι αρκετά σημαντική και ουσιώδης, ώστε να απαιτείται διαφορετική αντιμετώπιση για τα προβλήματα σχεδιασμού της πολεμικής αεροπορίας σε σχέση με αυτά της πολιτικής.

Οι έλεγχοι στους οποίους υποβάλλονται τα αεροσκάφη μπορούν να διαχωριστούν σε δύο κύριες κατηγορίες. Στους προγραμματισμένους ελέγχους και στους έκτακτους. Οι προγραμματισμένοι έλεγχοι με τη σειρά τους διαχωρίζονται σε δύο υποκατηγορίες, στους ελέγχους με βάση τις ώρες πτήσης του αεροσκάφους και στους ελέγχους με βάση το χρονικό διάστημα που έχει περάσει από τον τελευταίο έλεγχο του.

4.2 Βιβλιογραφική ανασκόπηση

Η έρευνα σχετικά με εφαρμογές των αερογραμμών είναι πολύ πιο ενεργή σήμερα, με εκατοντάδες δημοσιεύσεων κάθε χρόνο να είναι αφιερωμένες στις αερομεταφορές και στα προβλήματα που ανακύπτουν σε αυτές.

Οι Clarke et al. (4) εισήγαγαν ένα μοντέλο σχεδιασμού που περιλαμβάνει μελέτες τόσο για τη συντήρηση όσο και για το προσωπικό. Οι Klabjan et al. (11) απευθύνθηκαν σε ολοκληρωμένα προβλήματα που περιλαμβάνουν σχεδιασμό προγράμματος, δρομολογίων και πληρωμάτων. Στη μελέτη τους, η ώρα αναχώρησης μιας πτήσης μπορεί να αλλάξει ελαφρά, εφόσον παραμένει εντός κάποιων προκαθορισμένων ορίων.

Οι Clarke et al. (5) ανέπτυξαν μια μαθηματική διατύπωση για το πρόβλημα της εναλλαγής των αεροσκαφών (aircraft rotation problem) και το επέλυσαν με τη χαλάρωση Lagrange.

Οι Rushmeier και Kontogiorgis (16) παρουσίασαν ένα μεικτό ακέραιο μοντέλο για ανάθεση πτήσεων μεγάλης κλίμακας, που υπόκειται σε μια ποικιλία περιορισμών (mixed integer multicommodity flow model for large scale fleet assignment).

Οι Gopalan και Talluri (8) και ο Talluri (20) ερεύνησαν το πρόβλημα της εύρεσης της βέλτιστης δρομολόγησης για ένα αεροσκάφος, διασφαλίζοντας ότι ικανοποιούνται κάποιες συγκεκριμένες απαιτήσεις βραχυπρόθεσμων προγραμματισμένων ελέγχων.

Οι Barnhart et al. (2) παρουσίασαν ένα μοντέλο και μια τεχνική προσέγγισης για να επιλύσουν ταυτόχρονα το πρόβλημα ανάθεσης πτήσεων και δρομολόγησης αεροσκαφών (fleet assignment and aircraft routing problem). Οι Gopalan και Talluri (9) αναζήτησαν μοντέλα και τεχνικές επίλυσης για ποικίλα προβλήματα που περιλαμβάνουν αποφάσεις ανάθεσης πτήσεων και δρομολόγησης συντηρήσεων (fleet assignment and maintenance routing decisions).

Οι Feo και Bard (7) μοντελοποίησαν το πρόβλημα δρομολόγησης συντηρήσεων σαν ένα πρόβλημα διαχωρισμού συνόλων (maintenance routing problem as a set partitioning problem), και χρησιμοποίησαν ευρετικούς αλγόριθμους για το maintenance routing problem.

Τέλος, οι Weistroffer και Narula (22) παρέχουν μία έρευνα σχετικά με το επίπεδο στο οποίο βρίσκονται σήμερα τα λογισμικά που υποστηρίζουν τη λήψη αποφάσεων (Decision Support System software - DSSs).

Στη συντριπτική τους πλειοψηφία όμως, όλες αυτές οι μελέτες αναφέρονται σε προβλήματα της πολιτικής αεροπορίας, τα οποία διαφέρουν όπως ειπώθηκε τόσο ως προς τους στόχους, όσο και ως προς τις απαιτήσεις από αυτά της πολεμικής αεροπορίας. Αποτέλεσμα αυτής της διαφοροποίησης είναι η αναγκαιότητα ανάπτυξης ενός νέου μοντέλου για την επίλυση προβλημάτων FMP της πολεμικής αεροπορίας.

4.3 Περιγραφή του προβλήματος

Η Πολεμική Αεροπορία (Π.Α.) είναι υπεύθυνη για την αεράμυνα της χώρας απέναντι στις εξωτερικές απειλές. Διαιρείται σε τέσσερεις Διοικήσεις: στο Αρχηγείο Τακτικής Αεροπορίας, στη Διοίκηση Αεροπορικής Υποστήριξης και στη Διοίκηση Αεροπορικής Εκπαίδευσης, και σε μία τέταρτη που περιλαμβάνει άλλες υπηρεσίες.

Η αφορμή για την ενασχόληση με το συγκεκριμένο πρόβλημα δόθηκε από μια πραγματική εφαρμογή της Π.Α. Αφορά σε ένα συνηθισμένο πρόβλημα αεροπορικών επιχειρήσεων μιας τυπικής Πτέρυγας Μάχης της Π.Α. Κάθε πτέρυγα μάχης διαιρείται σε μοίρες και κάθε μοίρα διαθέτει ένα συγκεκριμένο αριθμό αεροσκαφών διαφόρων τύπων.

Η διοίκηση της πτέρυγας μάχης καθορίζει στην αρχή του ορίζοντα σχεδιασμού τις απαιτήσεις σε ώρες πτήσης για κάθε συνδυασμό μοίρας και χρονικής περιόδου του ορίζοντα. Συνήθως, επιτρέπονται κάποιες μικρές αποκλίσεις από τις τιμές που θέτει η διοίκηση. Φυσικά ενδέχεται να υπάρξει και η επιθυμία για απόλυτη εναρμόνιση με τον επιθυμητό αριθμό ωρών, οπότε δεν μπορεί να υπάρξει καμία απόκλιση από αυτόν.

Κάθε τύπος αεροσκάφους έχει διαφορετικές δυνατότητες πτήσης καθώς και διαφορετικές απαιτήσεις σε συντήρηση. Συνεπώς, αυτή η μελέτη αφορά σε ένα και μόνο τύπο αεροσκάφους. Είναι όμως εύκολο να προσαρμοστεί αυτό το μοντέλο και για άλλους τύπους αεροσκαφών, εφόσον εμπλέκονται στο σχεδιασμό που καταρτίζουμε.

4.4 Δεδομένα του προβλήματος

Ορίζουμε ως υπολειπόμενο χρόνο πτήσης ενός αεροσκάφους το χρόνο που μπορεί να πετάξει ώσπου να είναι απαραίτητο να καθηλωθεί για συντήρηση. Αυτός ο χρόνος είναι αυστηρά θετικός αν και μόνο αν το αεροσκάφος είναι διαθέσιμο να πετάξει (δε βρίσκεται δηλαδή στο σταθμό συντήρησης). Αντίστοιχα, ορίζουμε ως υπολειπόμενο χρόνο συντήρησης ενός μη διαθέσιμου αεροσκάφους το χρόνο που απομένει μέχρι το αεροσκάφος να εξέλθει από το σταθμό συντήρησης και να καταστεί και πάλι διαθέσιμο. Ομοίως, η τιμή του υπολειπόμενου χρόνου συντήρησης είναι αυστηρά θετική αν και μόνο αν το αεροσκάφος τη δεδομένη χρονική στιγμή είναι καθηλωμένο για συντήρηση.

Σε κάθε περίπτωση, ο συνολικός υπολειπόμενος χρόνος πτήσης της κάθε μοίρας ισούται με το άθροισμα των υπολειπόμενων χρόνων πτήσης όλων των αεροσκαφών της μοίρας και ο συνολικός υπολειπόμενος χρόνος πτήσης της πτέρυγας ισούται με το άθροισμα των υπολειπόμενων χρόνων πτήσης όλων των μοιρών.

Για τις ανάγκες συντήρησης των αεροσκαφών της πτέρυγας υπάρχει ένας σταθμός συντήρησης. Ο σταθμός συντήρησης έχει συγκεκριμένη χωρητική και χρονική δυναμικότητα.

Με όλα αυτά τα δεδομένα, και λαμβάνοντας υπόψη τους φυσικούς περιορισμούς που προκύπτουν από τη δυναμικότητα του σταθμού συντήρησης, στόχος είναι να αναπτυχθεί ένα πρόγραμμα σχεδιασμού πτήσεων και συντήρησης για κάθε αεροσκάφος της πτέρυγας μάχης χωριστά.

Όπως έχει ήδη αναφερθεί παραπάνω, αντικειμενικός στόχος αυτού του σχεδιασμού είναι η μεγιστοποίηση του βαθμού ετοιμότητας και ταυτόχρονα της αποτελεσματικότητας του εναέριου στόλου της χώρας, ώστε να είναι ετοιμοπόλεμος στο μέγιστο βαθμό. Το συγκεκριμένο μοντέλο εκφράζει αυτή την ετοιμότητα ως το συνολικό αριθμό των διαθέσιμων αεροσκαφών για συμμετοχή σε εναέριες επιχειρήσεις. Από μόνος του αυτός ο αριθμός όμως δεν μας δίνει πληροφόρηση για τον καταμερισμό του χρόνου πτήσης ανάμεσα στις μοίρες και στο κάθε αεροσκάφος ξεχωριστά. Έτσι, ενώ η πρώτη και η δεύτερη αντικειμενική συνάρτηση της προτεινόμενης μορφοποίησης μεγιστοποιούν τον ελάχιστο αριθμό διαθέσιμων αεροσκαφών και τον υπολειπόμενο χρόνο πτήσης τους αντίστοιχα για όλη την πτέρυγα, η τρίτη και τέταρτη μεγιστοποιούν τα ίδια κριτήρια, αυτή τη φορά όμως για την κάθε μοίρα της πτέρυγας ξεχωριστά.

4.5 Ανάπτυξη του μοντέλου

Παρακάτω παρατίθεται το μαθηματικό μοντέλο που αναπτύχθηκε για το πρόβλημα FMP που περιγράφηκε παραπάνω. Οι συμβολισμοί είναι οι εξής:

Μεταβλητές απόφασης

z_1: *ελάχιστος αριθμός διαθέσιμων αεροσκαφών σε όλη την πτέρυγα σε όλες τις περιόδους,*

z_2: *ελάχιστος υπολειπόμενος χρόνος πτήσης όλης της πτέρυγας σε όλες τις περιόδους,*

z_3: *ελάχιστος αριθμός διαθέσιμων αεροσκαφών της κάθε μοίρας σε όλες τις περιόδους,*

z_4: *ελάχιστος υπολειπόμενος χρόνος πτήσης της κάθε μοίρας σε όλες τις περιόδους,*

a_{mnt}: *δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το αεροσκάφος n της μοίρας m είναι διαθέσιμο την περίοδο t και την τιμή 0 αλλιώς,*

y_{mnt}: *υπολειπόμενος χρόνος πτήσης του αεροσκάφους n της μοίρας m στην αρχή της χρονικής περιόδου t,*

x_{mnt}: *χρόνος πτήσης (που έχει πετάξει) το αεροσκάφος n της μοίρας m κατά τη διάρκεια της χρονικής περιόδου t,*

g_{mnt}: *υπολειπόμενος χρόνος συντήρησης του αεροσκάφους n της μοίρας m στην αρχή της χρονικής περιόδου t,*

h_{mnt}: *χρόνος συντήρησης του αεροσκάφους n της πτέρυγας m κατά τη διάρκεια της χρονικής περιόδου t,*

d_{mnt}: *δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το αεροσκάφος n της μοίρας m εξέρχεται από το σταθμό συντήρησης στην αρχή της περιόδου t και την τιμή 0 αλλιώς,*

f_{mnt}: *δυαδική μεταβλητή απόφασης η οποία παίρνει την τιμή 1 αν το αεροσκάφος n της μοίρας m εισέρχεται στο σταθμό συντήρησης στην αρχή της περιόδου t και την τιμή 0 αλλιώς,*

q_t, p_{mnt}, r_{mnt}: *βοηθητικές δυαδικές μεταβλητές.*

Παράμετροι

M: *σύνολο μοιρών της πτέρυγας, με δείκτη m,*

N_m: *σύνολο αεροσκαφών στη μοίρα m, με δείκτη n,*

T: *διάρκεια του ορίζοντα σχεδιασμού με δείκτη t,*

S_{mt}: *απαιτούμενος χρόνος πτήσης της πτέρυγας m κατά τη διάρκεια της χρονικής περιόδου t,*

B_t: *χρονική δυναμικότητα του σταθμού συντήρησης κατά τη χρονική περίοδο t,*

G: υπολειπόμενος χρόνος συντήρησης ενός αεροσκάφους αμέσως μόλις εισέρχεται στο σταθμό συντήρησης,

Y: υπολειπόμενος χρόνος πτήσης ενός αεροσκάφους αμέσως μόλις εξέρχεται από το σταθμό συντήρησης,

C: μέγιστος αριθμός αεροσκαφών που μπορεί να χειρισθεί ταυτόχρονα ο σταθμός συντήρησης,

$A1_{mn}$: *κατάσταση (0 ή 1) του αεροσκάφους n της μοίρας m την πρώτη περίοδο του ορίζοντα σχεδιασμού,*

$Y1_{mn}$: *υπολειπόμενος χρόνος πτήσης του αεροσκάφους n της μοίρας m την πρώτη περίοδο του ορίζοντα σχεδιασμού,*

$G1_{mn}$: *υπολειπόμενος χρόνος πτήσης του αεροσκάφους n της μοίρας m την πρώτη περίοδο του ορίζοντα σχεδιασμού,*

X_{max}: *μέγιστος χρόνος που μπορεί ένα αεροσκάφος να πετάξει σε μια χρονική περίοδο,*

Y_{min}: *ελάχιστος υπολειπόμενος χρόνος πτήσης ενός διαθέσιμου αεροσκάφους,*

G_{min}: *ελάχιστος υπολειπόμενος χρόνος συντήρησης ενός μη διαθέσιμου αεροσκάφους,*

L,U: πραγματικοί αριθμοί που υποδεικνύουν τη μέγιστη επιτρεπτή απόκλιση από τον απαιτούμενο χρόνο πτήσης (S_{mt}),

K: ένας επαρκώς μεγάλος αριθμός.

Με βάση το παραπάνω μοντέλο η μορφοποίηση του προβλήματος FMP που αναλύθηκε παραπάνω είναι η εξής:

$$\text{Max } z_1 \tag{1}$$

$$\text{Max } z_2 \tag{2}$$

$$\text{Max } z_3 \tag{3}$$

$$\text{Max } z_4 \tag{4}$$

$$\text{s.t. } z_1 \leq \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} a_{mnt},\ t = 2,\ldots,T+1 \tag{5}$$

$$z_2 \leq \sum_{m=1}^{|M|}\sum_{n=1}^{|N_m|} y_{mnt},\ t = 2,\ldots,T+1 \tag{6}$$

$$z_3 \leq \sum_{n=1}^{|N_m|} a_{mnt},\ m = 1,\ldots,|M|,\ t = 2,\ldots,T+1 \tag{7}$$

$$z_4 \leq \sum_{n=1}^{|N_m|} y_{mnt},\ m = 1,\ldots,|M|,\ t = 2,\ldots,T+1 \tag{8}$$

Ήδη έχει αναφερθεί τι δηλώνουν τα ζευγάρια των αντικειμενικών συναρτήσεων (1), (2) και (3), (4). Αυτές σε συνδυασμό με τους περιορισμούς (5), (6) και (7), (8) αντίστοιχα υποδηλώνουν τον ελάχιστο αριθμό διαθέσιμων αεροσκαφών καθώς και χρόνου πτήσης στην πτέρυγα και σε κάθε μοίρα. Κοιτάζοντας τα σύνολα των περιορισμών (5) ως (8) παρατηρούμε ότι σε κανένα δεν περιλαμβάνεται η πρώτη περίοδος (όλα ξεκινούν από t=2). Αυτό συμβαίνει διότι η διαθεσιμότητα της πρώτης περιόδου είναι πάντα γνωστή. Γι' αυτό το λόγο και ο σχεδιασμός επεκτείνεται ως την περίοδο $T+1$, ώστε στην επόμενη εφαρμογή του μοντέλου να υπάρχει και πάλι ως δεδομένη η διαθεσιμότητα της πρώτης χρονικής περιόδου.

$$y_{mnt+1} = y_{mnt} - x_{mnt} + Yd_{mnt+1},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{9}$$

$$d_{mnt+1} \geq a_{mnt+1} - a_{mnt},\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{10}$$

$$a_{mnt+1} - a_{mnt} + 1.1(1-d_{mnt+1}) \geq 0.1,\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{11}$$

$$g_{mnt+1} = g_{mnt} - h_{mnt} + Gf_{mnt+1},\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{12}$$

$$f_{mnt+1} \geq a_{mnt} - a_{mnt+1},\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{13}$$

$$a_{mnt} - a_{mnt+1} + 1.1(1-f_{mnt+1}) \geq 0.1,\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,\ldots,T \tag{14}$$

Το 9° σύνολο περιορισμών "ενημερώνει" τον υπολειπόμενο χρόνο πτήσης του αεροσκάφους στην αρχή της επόμενης περιόδου με βάση τον υπολειπόμενο χρόνο πτήσης του στην αρχή της τρέχουσας περιόδου και αφαιρώντας τον χρόνο που έχει πετάξει σε αυτή. Σε περίπτωση μάλιστα που το αεροσκάφος μέσα στη τρέχουσα χρονική περίοδο εξέλθει από το σταθμό συντήρησης (η δυαδική μεταβλητή d πάρει την τιμή 1), τότε σε αυτόν προστίθεται και ο μέγιστος χρόνος πτήσης Y αμέσως μετά τη συντήρηση.

Με τον ίδιο τρόπο λειτουργεί το 12° σύνολο περιορισμών, το οποίο "ενημερώνει" τον υπολειπόμενο χρόνο συντήρησης ενός αεροσκάφους κατά τη διάρκεια της επόμενης περιόδου με βάση τον υπολειπόμενο χρόνο συντήρησης στην αρχή της τρέχουσας, αφαιρώντας αυτή τη φορά το χρόνο που έχει υποστεί συντήρηση σε αυτήν την περίοδο. Και εδώ έχουμε τη δυαδική μεταβλητή f η οποία παίρνει την τιμή 1 σε περίπτωση που το αεροσκάφος εισέλθει στο σταθμό συντήρησης και θα υποβληθεί σε έλεγχο με το μέγιστο χρόνο συντήρησης G.

Τα σύνολα περιορισμών (10), (11), (13), (14) εξασφαλίζουν ότι οι μεταβλητές d και f θα πάρουν τις σωστές τιμές, βασισμένες στις τιμές της μεταβλητής a. Ας υποθέσουμε ότι έχουμε το n-οστό αεροσκάφος της m-οστής μοίρας. Τότε οι μεταβλητές (a_{mnt} , a_{mnt+1}) μπορούν να πάρουν τις τιμές (0,1), (0,0), (1,0), (1,1) και οι αντίστοιχες διαφορές είναι (a_{mnt+1} - a_{mnt}) 1, 0, -1, 0 αντίστοιχα. Η μεταβλητή d_{mnt} οφείλει να πάρει την τιμή 1 όταν (a_{mnt} , a_{mnt+1}) = (0,1) και αυτό διασφαλίζεται από το 10° σύνολο περιορισμών. Σε κάθε άλλη περίπτωση θα πρέπει να πάρει την τιμή 0 και αυτό μας το εξασφαλίζει το 11° σύνολο περιορισμών. Ομοίως λειτουργούν και τα σύνολα (13) και (14) αυτή τη φορά για τη μεταβλητή f_{mnt}.

$$LS_{mt} \le \sum_{n=1}^{|N_m|} x_{mnt} \le US_{mt},\ m = 1,\dots,|M|,\ t = 1,\dots,T \tag{15}$$

$$\sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} h_{mnt} \le B_t,\ t = 1,\dots,T \tag{16}$$

$$\sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} (1 - a_{mnt}) \le C,\ t = 2,\dots,T+1 \tag{17}$$

$$B_t \le \sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} h_{mnt} + K(1 - q_t),\ t = 1,\dots,T \tag{18}$$

$$\sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} g_{mnt} \le \sum_{m=1}^{|M|} \sum_{n=1}^{|N_m|} h_{mnt} + Kq_t,\ t = 1,\dots,T \tag{19}$$

Το 15° σύνολο περιορισμών διασφαλίζει πως θα καλυφθούν οι πτητικές απαιτήσεις της κάθε μοίρας. Τα U και L αποτελούν τα άνω και κάτω επιτρεπόμενα όρια απόκλισης. Αν για παράδειγμα έχουμε $L=0.9$ και $U=1.1$ τότε η μέγιστη επιτρεπόμενη απόκλιση είναι 10%. Αν $L=U=1$ τότε προφανώς δεν επιτρέπεται καμία παρέκκλιση από τις επιθυμητές τιμές.

Τα σύνολα (16) και (17) διασφαλίζουν πως δεν παραβιάζονται σε καμία χρονική περίοδο οι περιορισμοί χωρητικότητας και χρόνου του σταθμού συντήρησης, ενώ τα σύνολα (18) και (19) εξασφαλίζουν πως το συνεργείο συντήρησης δεν αδρανεί σε περίπτωση που υπάρχει έστω και ένα αεροσκάφος στην ουρά αναμονής. Με την εισαγωγή της βοηθητικής δυαδικής μεταβλητής q επιτυγχάνουμε την εξίσωση του χρόνου συντήρησης που προσφέρεται από το σταθμό είτε με τη συνολική χρονική δυναμικότητα του σταθμού σε κάθε χρονική περίοδο, είτε με τις συνολικές χρονικές απαιτήσεις συντήρησης της περιόδου, ανάλογα με το ποια είναι μικρότερη.

$$y_{mnt} + Kp_{mnt} \leq K,\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{20}$$

$$a_{mnt+1} \leq (y_{mnt} - x_{mnt})K + Kp_{mnt},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{21}$$

$$g_{mnt} + Kr_{mnt} \leq K,\quad m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{22}$$

$$1-a_{mnt+1} \leq (g_{mnt} - h_{mnt})K + Kr_{mnt},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=1,..,T \tag{23}$$

Η εισαγωγή των συνόλων περιορισμών (20) και (21) διασφαλίζει τον τερματισμό της διαθεσιμότητας ενός αεροσκάφους μόλις ο υπολειπόμενος χρόνος πτήσης του μηδενιστεί. Αν το y_{mnt} είναι μεγαλύτερο του μηδενός, τότε η βοηθητική δυαδική μεταβλητή p_{mnt} μηδενίζεται υποχρεωτικά. Έπειτα, το 21° σύνολο περιορισμών αναγκάζει τη μεταβλητή a_{mnt+1} να μηδενιστεί αν ισχύει $y_{mnt}=x_{mnt}$, δηλαδή ο υπολειπόμενος χρόνος πτήσης του αεροσκάφους εξαντλείται κατά την περίοδο t. Με τον ίδιο τρόπο, τα σύνολα (22) και (23) διασφαλίζουν πως ένα αεροσκάφος ξαναγίνεται διαθέσιμο όταν ο χρόνος συντήρησής του μηδενίζεται κατά την περίοδο t.

$$y_{mnt} \leq Ya_{mnt},\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,\ldots,T+1 \tag{24}$$

$$g_{mnt} \leq G(1-a_{mnt}),\ m=1,\ldots,|M|,\ n=1,\ldots,|N_m|,\ t=2,\ldots,T+1 \tag{25}$$

$$x_{mnt} \leq X_{max} a_{mnt}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m|,\ t = 1,\dots,T \tag{26}$$

$$y_{mnt} \geq Y_{min} a_{mnt}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m|,\ t = 2,\dots,T+1 \tag{27}$$

$$g_{mnt} \geq G_{min}(1 - a_{mnt}), \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m|,\ t = 2,\dots,T+1 \tag{28}$$

$$x_{mnt} \leq y_{mnt}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m|,\ t = 1,\dots,T \tag{29}$$

$$h_{mnt} \leq g_{mnt}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m|,\ t = 1,\dots,T \tag{30}$$

Το 24ο σύνολο περιορισμών περιορίζει τον υπολειπόμενο χρόνο πτήσης σε τιμές μικρότερες της μέγιστης δοθείσας (*Y*) και εξασφαλίζει αυτός θα ότι θα μηδενίζεται όταν το αεροσκάφος δεν είναι διαθέσιμο. Ομοίως και το 25^o σύνολο, αυτή τη φορά όμως για το χρόνο συντήρησης του αεροσκάφους.

Το 26^o σύνολο επιβάλλει ένα άνω όριο στο χρόνο που μπορεί ένα αεροσκάφος να πετάξει μέσα σε μια χρονική περίοδο, για τεχνικούς λόγους προφανώς. Τα σύνολα (27) και (28) με τον ίδιο τρόπο επιβάλλουν ένα κατώτατο όριο για τον υπολειπόμενο χρόνο πτήσης και συντήρησης ενός διαθέσιμου αεροσκάφους. Αυτοί οι περιορισμοί εισάγονται για να εξαλείψουν την πιθανότητα να υπάρχει ένα διαθέσιμο αεροσκάφος με μηδαμινό υπολειπόμενο χρόνο πτήσης.

Το σύνολο περιορισμών (29) εξασφαλίζει ότι ο χρόνος πτήσης ενός αεροσκάφους σε μια συγκεκριμένη περίοδο δεν υπερβαίνει τον υπολειπόμενο χρόνο πτήσης που διέθετε στην αρχή της περιόδου, ενώ το 30^o εισάγεται για τον ίδιο ακριβώς σκοπό, αλλά για το χρόνο συντήρησης.

$$a_{mn1} = A1_{mn}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m| \tag{31}$$

$$y_{mn1} = Y1_{mn}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m| \tag{32}$$

$$g_{mn1} = G1_{mn}, \quad m = 1,\dots,|M|,\ n = 1,\dots,|N_m| \tag{33}$$

Τα επόμενα τρία σύνολα περιορισμών (31, 32 και 33) εισάγονται για να λάβουν αρχικές τιμές οι μεταβλητές *a*, *y* και *g* ώστε να εκκινήσει η λειτουργία του μοντέλου. Όταν ένα αεροσκάφος εξέρχεται από το σταθμό συντήρησης ή εισέρχεται σε αυτόν, οι υπολειπόμενοι χρόνοι πτήσης και συντήρησης ενημερώνονται αυτόματα. Γι' αυτό το λόγο οι μεταβλητές f_{mn1} και d_{mn1} δεν χρησιμοποιούνται.

$$x_{mnt}, h_{mnt} \geq 0;\ m = 1,\ldots,|M|,\ n = 1,\ldots,|N_m|,\ t = 1,\ldots,T \quad (34)$$

$$y_{mnt}, g_{mnt} \geq 0;\ m = 1,\ldots,|M|,\ n = 1,\ldots,|N_m|,\ t = 1,\ldots,T+1 \quad (35)$$

$$p_{mnt}, r_{mnt}, q_t \text{ binary},\ m = 1,\ldots,|M|,\ n = 1,\ldots,|N_m|,\ t = 1,\ldots,T \quad (36)$$

$$a_{mnt}, d_{mnt}, f_{mnt} \text{ binary},\ m = 1,\ldots,|M|,\ n = 1,\ldots,|N_m|,\ t = 1,\ldots,T+1 \quad (37)$$

Τέλος, τα σύνολα 34, 35 και 36, 37 εκφράζουν τους περιορισμούς μη αρνητικότητας και ακεραιότητας αντίστοιχα. Οι μεταβλητές p_{mnt}, r_{mnt}, q_t, a_{mnt}, d_{mnt} και f_{mnt} πιο συγκεκριμένα είναι δυαδικές.

4.6 Επίλυση του μοντέλου FMP με την ADBASE

Όπως είδαμε η ADBASE δεν μπορεί να λύσει ακέραια προβλήματα. Οπότε οι περιορισμοί ακεραιότητας (δυαδικότητας) για τις μεταβλητές a_{mnt}, d_{mnt}, f_{mnt}, q_t, p_{mnt} και r_{mnt} του αρχικού προβλήματος, κατά την επίλυση του από την ADBASE δίνουν τη θέση τους σε περιορισμούς όπου οι παραπάνω μεταβλητές πρέπει να είναι μικρότερες ή ίσες της μονάδας και φυσικά μεγαλύτερες του μηδενός. Για παράδειγμα $a_{mnt} \leq 1$ m=1,...,|M|, n=1,...,|N_m|, t=1,..,T+1.

Επίσης, αναφέρθηκε νωρίτερα πως η ADBASE έχει ένα άνω όριο όσον αφορά στο μέγεθος των προβλημάτων που μπορεί να επιλύσει. Το άθροισμα του αριθμού των περιορισμών του προβλήματος, των γραμμών του πίνακα των κριτηρίων και του αριθμού 1 δεν πρέπει να υπερβαίνει το 1210. Στο μοντέλο που αναπτύξαμε ο συνολικός αριθμός των περιορισμών δίνεται ανάλογα με τις τιμές των παραμέτρων M, N και T από την εξής σχέση: $22{*}M{*}N{*}T + 7{*}T + 4{*}M{*}T + 3{*}M{*}N$ ενώ ο αριθμός των γραμμών του πίνακα των κριτηρίων είναι 4. Άρα τελικά θα πρέπει να έχουμε $22{*}M{*}N{*}T + 7{*}T + 4{*}M{*}T + 3{*}M{*}N \leq 1205$. Για οποιονδήποτε συνδυασμό των παραμέτρων M, N, T που δεν παραβιάζει τον παραπάνω περιορισμό μπορούμε να πάρουμε λύση (εφόσον φυσικά το πρόβλημα είναι εφικτό).

Για να εφαρμόσουμε την ADBASE σε ένα πρόβλημα πολυκριτήριας γραμμικής βελτιστοποίησης πρέπει προηγουμένως να έχουμε κατασκευάσει τα δύο αρχεία *.qq και *.ii στο σημειωματάριο. Το αρχείο *.qq για το συγκεκριμένο πρόβλημα είναι το εξής:

fmp.qq

```
***--****1********-------****3*********4*********5****** ADBASE06 MODE=1 SECTION
  1. NUMB              1  (NUMBER OF PROBLEMS TO BE SOLVED)
----------------------------
  2. MODE              1  (REGULAR/MOLP,TRANS) 1/3,4
----------------------------
  3. IFASE2            2  (PHASE II OPTION) 1 TO 5
  4. IFASE3            1  (PHASE III OPTION) 0,1,2
  5. IZFMT             3  (ZFILE EXPONENTIAL/FIXED FORMAT) 0 TO 10
  6. IXFMT             5  (XFILE EXPONENTIAL/FIXED FORMAT) 0 TO 10
  7. IWEAK             0  (EFFICIENT OR WEAKLY-EFFICIENT) 0,1
  8. MLISTB      2800000  (MAX NUMB EFFICIENT BASES) <=2800000
  9. FV              3.0  (RE-INVERSION TIMES M3 FACTOR) >2.0
----------------------------
 10. IPRINT(1)         0  (IN SFILE: OBVIOUS ERRORS/ANGLES) 0/1,2
 11. IPRINT(2)         0  (IN SFILE: PROBLEM COEFFICIENTS) 0,1
 12. IPRINT(3)         4  (IN SFILE: NOTHING/BASES/EXTREME PTS) 0/1,2,3/4,5,6
 13. IPRINT(4)         0  (IN SFILE: EFFICIENCY TOTALS) 0,1
 14. IPRINT(5)         0  (IN SFILE: INCREMENT FOR RUNNING TOTALS) 2 MINIMUM
 15. IPRINT(6)         0  (IN SFILE: PROBLEM/CUMULATIVE STATS) 0,1/2
 16. IPRINT(7)         0  (IN SFILE: CODE LISTS) 0,1
 17. IPRINT(8)         0  (ZFILE: OFF/WIDE/WIDE WITH HEADER) 0,1,2
 18. IZVBEG            1   (ZFILE BEGINS AT THIS BASIS)
 19. IZVEND      2800000   (ZFILE ENDS AT THIS BASIS)
 20. IZCL              1   (BEGINNING Z-VECTOR COMPONENT)
 21. IZCU             20   (ENDING Z-VECTOR COMPONENT)
 22. IPRINT(9)         0  (XFILE: OFF/WIDE/WIDE WITH HEADER) 0,1,2
 23. IXVBEG            1   (XFILE BEGINS AT THIS BASIS)
 24. IXVEND         1000   (XFILE ENDS AT THIS BASIS)
 25. IXCL              1   (BEGINNING X-VECTOR COMPONENT)
 26. IXCU             25   (ENDING X-VECTOR COMPONENT)
 27. IPRINT(10)        0  (IN SFILE: REDUCED COSTS/TABLEAUS) 0,1,2
 28. ITSBEG            1   (TABLEAUS BEGIN AT THIS BASIS)
 29. ITSEND           10   (TABLEAUS END AT THIS BASIS)
 30. ITVCL             1   (BEGINNING TABLEAU VARIABLE)
 31. ITVCU             6   (ENDING TABLEAU VARIABLE)
 32. IPRINT(11)        0  (LFILE: OFF/ON) 0,1
 33. ILBEG             0   (LFILE BEGINS ON WAY TO THIS BASIS)
 34. ILEND            50   (LFILE ENDS AT THIS BASIS)
 35. IPRINT(12)        0   (PREMULTIPLICATION T-MATRIX) 0,1
***--*************-------************************************** MODE>=3 SECTION
******  FOLLOWING COMMON PARAMETERS ARE SPECIFIED IN TFILE (9I8)      ******
******     NOMB    (PROBLEM NUMBER)                                   ******
******     NOBJS   (NUMBER OF OBJECTIVES)                             ******
******     NCONS   (NUMBER OF CONSTRAINTS/SUPPLIES)                   ******
******     NVARS   (NUMBER OF STRUCTURAL VARIABLES/DEMANDS)           ******
******     JLC     (C-COEFFICIENT LOWER LIMIT)                        ******
******     JUC     (UPPER LIMIT)                                      ******
******     JCDEN   (% C-MATRIX NONZERO DENSITY)                       ******
******     JADEN   (% A-MATRIX NONZERO DENSITY)                       ******
***--****1********-------****3*********4*********5*********6*********7*********8
 36. IPRINT(13)        0  (PFILE: OFF/WITH ARRAY STORAGE IN SFILE) 0,1/2
 37. KSEEDC        44187  (SEED FOR GENERATING C-MATRICES) <65536
 38. KSEEDA         6467  (SEED FOR GENERATING CONSTRAINTS) <65536
----------------------------
 39. JLA              -1  (A-COEFFICIENT LOWER LIMIT)
 40. JUA               8  (UPPER LIMIT)
 41. JLB              50  (B VALUE/A-ROW NORM LOWER LIMIT)
 42. JUB             100  (UPPER LIMIT)
----------------------------
```

```
 43. IDISP          20    (CENTER OF FEASIBLE REGION DISPLACEMENT)
***--****1*********-------****3*********4*********5*********6*********7*********8
 44. TABS       1.0D-11   (SMALLEST ZERO TOLERANCE IN MACHINE) TRAD
                                                              VALUE -12
 45. TCJZ       1.0D-7   (Cj-Zj ZERO TOLERANCE) TRAD VALUE -7
 46. TDEG       1.0D-7   (DEGENERACY ZERO TOLERANCE) TRAD VALUE -8
 47. TPIV       1.0D-6   (PIVOT ELEMENT ZERO TOLERANCE) TRAD VALUE -6
```

Είναι εμφανείς οι 47 παράμετροι που προαναφέρθηκαν καθώς και οι τιμές που τις επιλέγει ο χρήστης.

Εν τούτοις, το αρχείο *.ii διαφοροποιείται κάθε φορά που αλλάζουμε τα δεδομένα του προβλήματος, με αποτέλεσμα να χρειαζόμαστε τη βοήθεια των γλωσσών προγραμματισμού για την κατασκευή του. Σε αυτό το πρόβλημα χρησιμοποιούνται δύο κώδικες, ένας σε FORTRAN και ένας σε C++ .

Το όνομα που έχουμε δώσει στο αρχείο που θα "τρέχει" η ADBASE είναι fmp.ii (και το *.qq έχει την ίδια ονομασία).

Για να κατασκευάσουμε το αρχείο fmp.ii με τη χρήση των γλωσσών προγραμματισμού δουλεύουμε ως εξής:

Αρχικά δημιουργούμε ένα αρχείο στο σημειωματάριο με το όνομα fmp_jat***.dat . Όπου υπάρχουν οι αστερίσκοι βάζουμε τις τιμές από τις παραμέτρους M, N, T του προβλήματος που θέλουμε να λύσουμε. Αν για παράδειγμα τα M, N, T είναι 2, 3, και 2 αντίστοιχα, τότε το αρχείο θα ονομάζεται fmp_jat232.dat . Αυτό το αρχείο θα μας δίνει τις παραμέτρους του προβλήματος και θα είναι διαμορφωμένο ως εξής:

M
N
T
G
Y
C
L
U
Xmax
Ymin
Gmin
BIG
B_t
S_{mt}
$A1_{mn}$

$Y1_{mn}$
$G1_{mn}$

Οι παράμετροι B_t, S_{mt}, $A1_{mn}$, $Y1_{mn}$, $G1_{mn}$ είναι πίνακες. Η παράμετρος B_t είναι πίνακας στήλη, η S_{mt} είναι δυσδιάστατος πίνακας με m αριθμό γραμμών και t αριθμό στηλών, ενώ οι $A1_{mn}$, $Y1_{mn}$, $G1_{mn}$ έχουν m αριθμό γραμμών και n αριθμό στηλών.

Παρακάτω δίνονται ενδεικτικά ένα αρχείο fmp_jat111.dat με *M=1*, *N=1*, *T=1* και ένα αρχείο fmp_jat333.dat με *M=3*, *N=3*, *T=3*, για να αποσαφηνιστεί ο τρόπος με τον οποίο τοποθετούμε τα δεδομένα όταν αυτά είναι σε στήλες.

fmp_jat111.dat :

1
1
1
320
300
1
0.90
1.10
170
0.1
0.1
552
552
119
1
243
0

fmp_jat333.dat :

3
3

3
320
300
1
0.90
1.10
70
0.1
0.1
320
534
595
574
118 92 84
94 102 89
105 102 118
1 1 1
1 1 1
1 1 1
115 161 288
265 179 164
78 108 280
0 0 0
0 0 0
0 0 0

Στη συνέχεια αυτό το αρχείο χρησιμοποιείται από τον κώδικα της fortran με το όνομα fmp_fortran.f90. Στην όγδοη γραμμή του κώδικα υπάρχει η έκφραση Open(20,file='fmp_jat***.dat'). Εδώ πρέπει να δίνουμε το όνομα του αρχείου που δημιουργούμε κάθε φορά (δηλαδή στη θέση των αστερίσκων να αλλάζουμε τις τιμές των παραμέτρων *M*, *N*, *T* για να το χρησιμοποιήσει ο κώδικας. Αφού λειτουργήσει ο κώδικας κατασκευάζεται ένα νέο αρχείο με την ονομασία fmp_fortran.ii. Το συγκεκριμένο αρχείο περιλαμβάνει τα δεδομένα που θα χρησιμοποιηθούν από την ADBASE για την επίλυση του προβλήματος, αλλά όχι στη μορφή (στη στοίχιση) που απαιτείται.

Η διευθέτηση των δεδομένων γίνεται με τη βοήθεια ενός νέου κώδικα σε γλώσσα C++ ο οποίος ονομάζεται fmp_adbase.c, έτσι ώστε να αποκτήσουν την ακριβή μορφή που χρειάζεται για να δουλέψει η ADBASE.

Από το φάκελο που υπάρχει το αρχείο fmp_fortran.f90 αντιγράφουμε το νέο αρχείο που δημιουργήθηκε (fmp_fortran.ii) στο φάκελο που βρίσκεται το πρόγραμμα της C++. Με αυτό τον τρόπο ο κώδικας της C++ θα "διαβάσει" τα δεδομένα από το fmp_fortran.ii και θα τα στοιχίσει στις κατάλληλες θέσεις. Από τη χρήση του προγράμματος της C++ παίρνουμε ένα νέο αρχείο με την ονομασία fmp.ii, το οποίο είναι και το τελικό που θα χρησιμοποιήσουμε.

Αυτό είναι το αρχείο που αντιγράφεται στη θέση C:\A06\A του σκληρού δίσκου του υπολογιστή. Από αυτό το σημείο μας μένει μόνο να "τρέξουμε" το πρόβλημα. Αυτό γίνεται στη γραμμή εντολών του υπολογιστή (ή αλλιώς "εκτέλεση" του μενού έναρξη). Όταν μας ανοίξει το παράθυρο της γραμμής εντολών πληκτρολογούμε την εντολή "command" ή "cmd" και μας ανοίγει το περιβάλλον DOS. Εκεί θα πρέπει να πάμε στον υποφάκελο A του φακέλου A06 της ADBASE με τη χρήση της εντολής "cd\" και enter αρχικά για να πάμε στο δίσκο C:\ και μετά την εντολή "cd A06\A". Πλέον "βρισκόμαστε" στο φάκελο που θέλουμε. Αυτό που μένει είναι να πληκτρολογήσουμε την εντολή για να τρέξει η ADBASE, οποία είναι η : "ajt fmp qwe". Αυτή η εντολή ουσιαστικά σημαίνει πως θα τρέξει τα αρχεία fmp.qq και fmp.ii και θα μας δώσει τις λύσεις σε ένα αρχείο με την ονομασία qwe.ss το οποίο βρίσκεται και αυτό στο φάκελο A. Στο αρχείο που παίρνουμε τις λύσεις μπορούμε να δώσουμε όποια ονομασία επιθυμούμε, απλά πληκτρολογώντας άλλο όνομα αντί για αυτό του παραδείγματος (qwe).

5.Εφαρμογή του μοντέλου

Σε αυτή την ενότητα παρατίθεται ένα λυμένο παράδειγμα για τη γραμμική χαλάρωση του προβλήματος FMP. Όπως αναφέρθηκε παραπάνω, εκτός του πολυκριτήριου προβλήματος, έχουν λυθεί και δύο μονοκριτήρια (ένα γραμμικό και το αντίστοιχο ακέραιο) τα οποία προέκυψαν από το αρχικό με τη μέθοδο που περιγράφηκε στην ενότητα 1.5.

Αρχικά θα πρέπει να γίνουν κάποιες παρατηρήσεις για να αποφευχθεί η οποιαδήποτε σύγχυση.

Ένα σημείο στο οποίο πρέπει να σταθούμε είναι οι αριθμήσεις που παρουσιάζονται στο αρχείο .ss των αποτελεσμάτων της ADBASE. Στο αριστερό μέρος κάθε λύσης βλέπουμε μια αρίθμηση η οποία δεν είναι συνεχής, παρουσιάζει κενά δηλαδή. Είναι μια κωδικοποίηση που χρησιμοποιεί η ADBASE και δηλώνει τον αριθμό της efficient base που επισκέφθηκε το πρόγραμμα και βρήκε τη συγκεκριμένη λύση. Θέτοντας την επιλογή IPRINT(4) του αρχείου *.qq (είναι η επιλογή 13) ίση με τη μονάδα το λογισμικό μας τυπώνει τον συνολικό αριθμό των efficient bases που επισκέφθηκε, τον αριθμό των αποτελεσματικών λύσεων που βρέθηκαν καθώς και τον αριθμό των unbounded efficient edges. Για να γίνει κατανοητό, στο παρακάτω παράδειγμα η αρίθμηση των ακραίων σημείων που βρέθηκαν ξεκινά από το 1 και συνεχίζει στο 2, 3, 6, 7 και συνεχίζει. Στους ενδιάμεσους αριθμούς των efficient bases που λείπουν το λογισμικό δεν βρήκε αποτελεσματικές λύσεις.

Επίσης, στις λύσεις της ADBASE οι μεταβλητές παρουσιάζονται με τους αριθμούς στους οποίους έχουν αντιστοιχηθεί, και οι μεταβλητές που παίρνουν την τιμή μηδέν δεν αναφέρονται στις λύσεις. Για παράδειγμα, η μεταβλητή X(1) είναι η z_1, η X(5) είναι η a_{111} και ούτω καθεξής, ενώ για παράδειγμα στο ακραίο σημείο 1 οι μεταβλητές που αντιστοιχούν στους αριθμούς 10, 11, 12, 14, 15, 16 και 17 είναι ίσες με μηδέν.

Τέλος, να αναφέρουμε πως κατά τη σύγκριση των αποτελεσμάτων της ADBASE με αυτά της AMPL υπάρχει μία ιδιομορφία. Η AMPL τυπώνει (με τιμή 0) και τις μεταβλητές d_{mn1} και f_{mn1}, οι οποίες όπως είδαμε δεν χρησιμοποιούνται πουθενά στο μοντέλο. Στην αρχική αρίθμηση των μεταβλητών για την ADBASE όμως οι συγκεκριμένες μεταβλητές δεν εμφανίζονται πουθενά. Οπότε όταν πάμε να συγκρίνουμε τα αποτελέσματα των δύο λογισμικών, κατά την αντιστοίχηση των μεταβλητών (για παράδειγμα η a_{111} στην AMPL αντιστοιχίζεται στη μεταβλητή X(5) της ADBASE και ούτω καθεξής) θα πρέπει να παραλείπονται οι μεταβλητές d_{mn1} και f_{mn1}.

Παρακάτω επιλύεται ένα πρόβλημα FMP με τιμές για τις παραμέτρους *M*, *N*, *T* ίσες με τη μονάδα. Στο παράρτημα παρατίθενται και άλλα λυμένα προβλήματα FMP με διαφορετικές τιμές για τις παραμέτρους *M*, *N* και *T*. Πιο συγκεκριμένα υπάρχουν δύο επιπλέον με *M*=*N*=*T*=*1*, και δύο με *M*=*N*=*T*=*2*. Επίσης, σε αυτά τα προβλήματα διαφοροποιούνται και άλλες παράμετροι όπως οι *L* και *U*.

M=1, N=1, T=1

Στο παρακάτω παράδειγμα οι επιτρεπόμενες αποκλίσεις από τον απαιτούμενο χρόνο πτήσης που ορίζεται από τη διοίκηση της πτέρυγας είναι αρκετά μεγάλες (10%). Προβλήματα με μικρότερες επιτρεπτές αποκλίσεις υπάρχουν στο παράρτημα. Ο σταθμός συντήρησης έχει δυνατότητα επισκευής ενός αεροσκάφους τη φορά (*C*=*1*).

M=1
N=1
T=1
G=320
Y=300
C=1
L=0.9
U=1.1
X_{max}=170
Y_{min}=0.1
G_{min}=0.1
K=552

B_1=552

S_{11}=119

$A1_{11}$=1

$Y1_{11}$=243

$G1_{11}$=0

Λύσεις ADBASE

FMP Problem with M=1 , N=1 , T=1

EXTREME POINT	CRITERION VALUES	NONZERO STRUCTURAL BASIC VARIABLE VALUES
1	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
2	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
		X(16) = 0.560579
3	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
		X(17) = 1.000000
6	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
7	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000

		X(9) = 130.900009
		X(13) = 0.626333
		X(15) = 0.001808
14	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
		X(16) = 0.560579
		X(17) = 1.000000
17	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(16) = 0.560579
18	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
		X(15) = 0.001808
		X(16) = 0.560579
25	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(17) = 1.000000
26	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000

		X(9) = 130.900009
		X(13) = 0.626333
		X(15) = 0.001808
		X(17) = 1.000000
46	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(15) = 0.001808
61	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(16) = 0.560579
		X(17) = 1.000000
62	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 130.900009
		X(13) = 0.626333
		X(15) = 0.001808
		X(16) = 0.560579
		X(17) = 1.000000
82	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(15) = 0.001808
		X(16) = 0.560579
98	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000

		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(15) = 0.001808
		X(17) = 1.000000
142	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 243.000000
		X(8) = 300.000000
		X(9) = 107.099998
		X(13) = 0.547000
		X(15) = 0.001808
		X(16) = 0.560579
		X(17) = 1.000000

NUMBER OF EFFICIENT BASES VISITED = 220
NUMBER OF EFFICIENT EXTREME POINTS = 16
NUMBER OF UNBOUNDED EFFICIENT EDGES = 0

Λύσεις AMPL (γραμμικό πρόβλημα)

Input = 0.003999
Solve = 0
Output = 0.001

Optimal solution: objective 450

z1 = 1
z2 = 300
z3 = 1
z4 = 300

a :=
m1 n1 1 1
m1 n1 2 1

y :=
m1 n1 1 243
m1 n1 2 300

x :=
m1 n1 1 107.1

g :=
m1 n1 1 0
m1 n1 2 0

h :=
m1 n1 1 0

d :=
m1 n1 1 0
m1 n1 2 0.547

f :=
m1 n1 1 0
m1 n1 2 0

q [*] :=
1 0

p :=
m1 n1 1 0

r :=
m1 n1 1 1

Παρατηρούμε ότι η λύση που προκύπτει από τη AMPL ταυτίζεται με αυτή που παίρνουμε από την ADBASE στο extreme point 25.

Λύσεις AMPL (ακέραιο πρόβλημα)

Input = 0.006
Solve = 0
Output = 0.001

Optimal integer solution: objective 285.9

z1 = 1
z2 = 135.9
z3 = 1
z4 = 135.9

a :=
m1 n1 1 1
m1 n1 2 1

y :=
m1 n1 1 243
m1 n1 2 135.9

x :=
m1 n1 1 107.1

g :=

m1 n1 1 0
m1 n1 2 0

d :=
m1 n1 1 0
m1 n1 2 0

f :=
m1 n1 1 0
m1 n1 2 0

q [*] :=
1 0

p :=
m1 n1 1 0

r :=
m1 n1 1 0

6.Συμπεράσματα-Προτάσεις

6.1 Συμπεράσματα από την εφαρμογή του μοντέλου

Στην παρούσα εργασία έχει αναπτυχθεί ένα μοντέλο για την επίλυση του μικτού ακέραιου πολυκριτήριου προβλήματος του σχεδιασμού πτήσεων και συντήρησης των αεροσκαφών σε μια πτέρυγα μάχης της Ελληνικής Πολεμικής Αεροπορίας. Έχοντας παρακολουθήσει την ανάλυση που έγινε στα προηγούμενα κεφάλαια καθώς και τα αποτελέσματα των δύο λογισμικών που χρησιμοποιήθηκαν για τη επίλυση του προβλήματος, καταλήγουμε σε αρκετά και χρήσιμα συμπεράσματα.

- Αρχικά παρατηρούμε πως ο αριθμός των αποτελεσματικών λύσεων αυξάνει ταχύτατα καθώς αυξάνει το μέγεθος του προβλήματος. Στα προβλήματα με τιμές παραμέτρων *M=1*, *N=1*, *T=1* οι λύσεις που παίρνουμε είναι λιγότερες. Μπορεί να συμβεί (σε σπάνιες περιπτώσεις βέβαια) κάποια μεγαλύτερα προβλήματα να έχουν λιγότερες ή ακόμα και μία αποτελεσματική λύση. Αυτό παρατηρούμε στο πρώτο πρόβλημα με *M*=2, *N*=2, *T*=2 που παρατίθεται στο παράρτημα. Υπήρξαν και προβλήματα με *M*=2, *N*=2 και *T*=2 που επιλύθηκαν και εμφάνισαν πολλές λύσεις (ίσως και εκατοντάδες). Μάλιστα, κατά την προσπάθεια επίλυσης κάποιων προβλημάτων με *M*=3, *N*=3 και *T*=3 το λογισμικό είχε αφεθεί να λειτουργεί για αρκετές ώρες και τελικά δεν κατέστη δυνατό να επιλυθεί το πρόβλημα.

- Το μονοκριτήριο πρόβλημα επιλύεται σαφώς πιο εύκολα. Χαρακτηριστικό είναι πως σε οποιοδήποτε πρόβλημα εξετάστηκε, η AMPL πάντα έδινε λύσεις. Σε ελάχιστες περιπτώσεις δεν πήραμε λύση και αυτό συνέβη μόνο στα ακέραια προβλήματα. Η εύρεση πολλών αποτελεσματικών λύσεων δεν επιτυγχάνεται εύκολα και απαιτεί πολύ χρόνο όσο το μέγεθος των προβλημάτων προς επίλυση αυξάνεται.

- Για μικτά ακέραια προβλήματα μεγάλου μεγέθους, είναι εξαιρετικά σπάνιο κάποια από τις αποτελεσματικές λύσεις της γραμμικής χαλάρωσης να ικανοποιεί τους περιορισμούς ακεραιότητας.

- Ο χρόνος που απαιτείται από την ADBASE για την εύρεση όλων των αποτελεσματικών λύσεων της γραμμικής χαλάρωσης ενός μεγάλου ακέραιου προβλήματος είναι απαγορευτικός. Ενώ σε μικρά προβλήματα ο απαιτούμενος χρόνος είναι εξαιρετικά μικρός (μικρότερος από δευτερόλεπτο), όταν η κλίμακα του προβλήματος ανεβαίνει το πρόγραμμα χρειάζεται όλο και περισσότερο χρόνο για να το επιλύσει. Στα προβλήματα με $M=2$, $N=2$ και $T=2$ που επιλύθηκαν το λογισμικό χρειάστηκε λίγο περισσότερο χρόνο, αλλά σε καμία περίπτωση αυτός δεν ήταν υπερβολικός. Εν τούτοις, στα προβλήματα με δεδομένα $M=3$, $N=3$ και $T=3$ ο χρόνος επίλυσης αυξήθηκε δραματικά με το αποτέλεσμα που αναφέρθηκε και σε προηγούμενη παράγραφο.

- Για την εύρεση όλων των αποτελεσματικών λύσεων ενός προβλήματος μεικτού ακέραιου πολυκριτήριου προγραμματισμού απαιτείται η χρήση εξειδικευμένου λογισμικού. Γι' αυτό το λόγο, ακόμα και σήμερα η επίλυση τέτοιου είδους προβλημάτων μεγάλου μεγέθους είναι εξαιρετικά δύσκολη και σε αρκετές περιπτώσεις αδύνατη.

- Επί του παρόντος, τα λογισμικά μεικτού ακέραιου πολυκριτήριου προγραμματισμού που έχουμε στη διάθεσή μας είναι εξαιρετικά περιορισμένα και προορίζονται μόνο για την επίλυση ειδικών προβλημάτων. Κάποια από αυτά αναφέρθηκαν στο δεύτερο κεφάλαιο. Συχνά χρησιμοποιούνται ευρετικοί αλγόριθμοι, οι οποίοι λόγω της φύσης τους (δεν είναι γενικής χρήσης), δεν μπορούν να έχουν ικανοποιητικά αποτέλεσμα σε όλα τα προβλήματα.

6.2 Προτάσεις

- Προς το παρόν, για τη μελλοντική έρευνα πάνω στο αντικείμενο της πολυκριτήριας βελτιστοποίησης, αυτό που είναι πιο εύκολα πραγματοποιήσιμο είναι η χρήση ευρετικών (ή μεταευρετικών) αλγόριθμων για μικτή ακέραια πολυκριτήρια βελτιστοποίηση. Ήδη υπάρχουν ιστοσελίδες που αποτελούν μεγάλες βάσεις δεδομένων για έτοιμους ευρετικούς αλγόριθμους στις οποίες μπορούμε να ανατρέξουμε ανά πάσα στιγμή. Μία τέτοια είναι η MOMHLib++ που αναφέρθηκε και σε προηγούμενο κεφάλαιο και λειτουργεί ως βιβλιοθήκη μεταευρετικών αλγορίθμων για την πολυκριτήρια βελτιστοποίηση. Μάλιστα, οι συγκεκριμένοι αλγόριθμοι μπορούν να δουλέψουν σε οποιονδήποτε τυπικό compiler της C++.

- Η μεγάλη πρόκληση για τον τομέα της λήψης αποφάσεων είναι η έρευνα για την περαιτέρω ανάπτυξη εξειδικευμένων αναλυτικών αλγόριθμων πολυκριτήριας βελτιστοποίησης, οι οποίοι θα έχουν τη δυνατότητα εύρεσης του πλήρους μετώπου των αποτελεσματικών λύσεων πάνω σε τέτοια προβλήματα. (Kozanidis G. (2008))

7.Παράρτημα

Ο κώδικας της fortran είναι ο εξής:

```
Program fmp_fortran
Implicit None

Integer::M,N,T,j,k,i,NOMB,NOBJS,N1,IK,IE,IS,IFASE0,NGRAYS,noumero1,midenika17,
midenika31,midenika32,midenika33
Real::G,Y,C,L,U,Xmax,Ymin,Gmin,BIG
Real,allocatable::B(:),S(:,:),A1(:,:),G1(:,:),Y1(:,:)

Open(20,file='fmp_jat111-3.dat')
Open(30,file='fmp_fortran.ii')

Read(20,*) M
Read(20,*) N
Read(20,*) T
Read(20,*) G
Read(20,*) Y
Read(20,*) C

Read(20,*) L
Read(20,*) U
Read(20,*) Xmax
Read(20,*) Ymin
Read(20,*) Gmin
Read(20,*) BIG
Read(20,*)

BIG=BIG+1

Allocate(B(T),S(M,T),A1(M,N),G1(M,N),Y1(M,N))

Do i=1,T
        Read(20,*) B(i)
End do
Read(20,*)

Do j=1,M
        Read(20,*) (S(j,i),i=1,T)
End do
Read(20,*)

Do j=1,M
        Read(20,*) (A1(j,k),k=1,N)
End do
Read(20,*)

Do j=1,M
        Read(20,*) (Y1(j,k),k=1,N)
End do
Read(20,*)

Do j=1,M
        Read(20,*) (G1(j,k),k=1,N)
End do

write(30,'(I1,1x,I1,1x,I1)') M,N,T
NOMB=1
NOBJS=4
N1=4 + T + 6*M*N*T + 3*(M*N*(T+1))
IK=6*T + 3*M*T + 15*M*N*T
```

```
IE=2*M*N*T + 3*M*N
IS=5*M*N*T + M*T + T
IFASE0=0
NGRAYS=40
write(30,'(8(I8))') NOMB,NOBJS,N1,IK,IE,IS,IFASE0,NGRAYS


write(30,'(I8)') 5*T + 2*M*T + 38*M*N*T


!le.LHS


! 5o set periorismwn
noumero1=0
Do i=1,T

        noumero1=noumero1+1

        write(30,'(I5,I5,f17.6)') noumero1,1,1.

        Do j=1,M
                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , -1.

                End do
        End do
End do

! 6o set periorismwn
noumero1=T
Do i=1,T

        noumero1=noumero1+1

        write(30,'(I5,I5,f17.6)') noumero1,2,1.

        Do j=1,M
                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i + 1 , -1.

                End do
        End do
End do

! 7o set periorismwn
noumero1=2*T
Do j=1,M
        Do i=1,T

                noumero1=noumero1+1

                write(30,'(I5,I5,f17.6)') noumero1,3,1.

                Do k=1,N
```

```
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , -1.

                End do
        End do
End do

! 8o set periorismwn
noumero1=2*T + M*T
Do j=1,M
        Do i=1,T

                noumero1=noumero1+1

                write(30,'(I5,I5,f17.6)') noumero1,4,1.

                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i + 1 , -1.

                End do
        End do
End do

! 11o set periorismwn
noumero1=2*T + 2*M*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i       +  1
, -1.
                Write(30,'(I5,I5,f17.6)') noumero1,        4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i ,
1.
            Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 2*M*N*T + (j-1)*N*T + (k-
1)*T + i , 1.1

                End do
        End do
End do

! 14o set periorismwn
noumero1=2*T + 2*M*T + M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i , -1.
                Write(30,'(I5,I5,f17.6)') noumero1,        4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i
        + 1 , 1.
            Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 3*M*N*T+ (j-1)*N*T + (k-1)*T
+ i , 1.1

                End do
        End do
End do

! 15o set(b) periorismwn
```

```
noumero1=2*T + 2*M*T + 2*M*N*T
Do j=1,M
        Do i=1,T

                noumero1=noumero1+1

                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
1.

                End do
        End do
End do

! 16o set periorismwn
noumero1=2*T + 3*M*T + 2*M*N*T
Do i=1,T

        noumero1=noumero1+1

        Do j=1,M
                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-1)*N*T + (k-
1)*T + i , 1.

                End do
        End do
End do

! 18o set periorismwn
noumero1=3*T + 3*M*T + 2*M*N*T
Do i=1,T

        noumero1=noumero1+1

        write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + i, BIG

        Do j=1,M
                Do k=1,N

                        Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-
1)*N*T + (k-1)*T + i , -1.

                End do
        End do
End do

! 19o set periorismwn
noumero1=4*T + 3*M*T + 2*M*N*T
Do i=1,T

        noumero1=noumero1+1

        Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + i, -BIG

        Do j=1,M
                Do k=1,N

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i, 1.
```

```
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-1)*N*T + (k-
1)*T + i, -1.

                End do
        End do
End do

! 20o set periorismwn
noumero1=5*T + 3*M*T + 2*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + T + (j-
1)*N*T + (k-1)*T + i, BIG

                End do
        End do
End do

! 21o set periorismwn
noumero1=5*T + 3*M*T + 3*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i , -BIG
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
BIG
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + T + (j-
1)*N*T + (k-1)*T + i , -BIG

                End do
        End do
End do

! 22o set periorismwn
noumero1=5*T + 3*M*T + 4*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 5*M*N*T + T + (j-
1)*N*T + (k-1)*T + i , BIG

                End do
        End do
End do

! 24o set periorismwn
```

```
noumero1=5*T + 3*M*T + 5*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , -Y


                End do
        End do
End do

! 25o set periorismwn
noumero1=5*T + 3*M*T + 6*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , G


                End do
        End do
End do

! 26o set periorismwn
noumero1=5*T + 3*M*T + 7*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i , -Xmax


                End do
        End do
End do

! 29o set periorismwn
noumero1=5*T + 3*M*T + 8*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1)+(j-1)*N*(T+1) + (k-1)*(T+1)
+ i , -1.
```

```
                End do
        End do
End do

! 30o set periorismwn
noumero1=5*T + 3*M*T + 9*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-1)*N*T + (k-
1)*T + i , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i , -1.

                End do
        End do
End do

! periorismoi xalarwsis dyadikwn metavlitwn ->a
noumero1=5*T + 3*M*T + 10*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1, 1.

                End do
        End do
End do

! periorismoi xalarwsis dyadikwn metavlitwn ->d
noumero1=5*T + 3*M*T + 11*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 2*M*N*T + (j-1)*N*T +
(k-1)*T + i, 1.

                End do
        End do
End do

! periorismoi xalarwsis dyadikwn metavlitwn ->f
noumero1=5*T + 3*M*T + 12*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 3*M*N*T+ (j-1)*N*T +
(k-1)*T + i, 1.
```

```
                End do
        End do
End do

! periorismoi xalarwsis dyadikwn metavlitwn ->q
noumero1=5*T + 3*M*T + 13*M*N*T
Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + i, 1.

End do

! periorismoi xalarwsis dyadikwn metavlitwn ->p
noumero1=6*T + 3*M*T + 13*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 4*M*N*T + T + (j-
1)*N*T + (k-1)*T + i, 1.

                End do
        End do
End do

! periorismoi xalarwsis dyadikwn metavlitwn ->r
noumero1=6*T + 3*M*T + 14*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 5*M*N*T + T + (j-
1)*N*T + (k-1)*T + i, 1.

                End do
        End do
End do


write(30,'(I8)') 3*T + M*T + 10*M*N*T


!le.RHS


! RHS le 11o set
noumero1=2*T + 2*M*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1
```

```
                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS le 14o set
noumero1=2*T + 2*M*T + M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS le 15o set (b)
noumero1=2*T + 2*M*T + 2*M*N*T
Do j=1,M
        Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , U*S(j,i)

        End do
End do

! RHS le 16o set
noumero1=2*T + 3*M*T + 2*M*N*T
Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , B(i)

End do

! RHS 18o set
noumero1=3*T + 3*M*T + 2*M*N*T
Do i=1,T

        noumero1=noumero1+1

        write(30,'(I5,f21.6)') noumero1, BIG-B(i)

End do

! RHS le 20o set
noumero1=5*T + 3*M*T + 2*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , BIG
```

```
                End do
        End do
End do

! RHS le 22o set
noumero1=5*T + 3*M*T + 4*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , BIG

                End do
        End do
End do

! RHS le 25o set
noumero1=5*T + 3*M*T + 6*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , G

                End do
        End do
End do

! RHS apo xalarwsi dyadikwn metavlitwn ->a
noumero1=5*T + 3*M*T + 10*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS apo xalarwsi dyadikwn metavlitwn ->d
noumero1=5*T + 3*M*T + 11*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS apo xalarwsi dyadikwn metavlitwn ->f
noumero1=5*T + 3*M*T + 12*M*N*T
```

```
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS apo xalarwsi dyadikwn metavlitwn ->q
noumero1=5*T + 3*M*T + 13*M*N*T
Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

End do

! RHS apo xalarwsi dyadikwn metavlitwn ->p
noumero1=6*T + 3*M*T + 13*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS apo xalarwsi dyadikwn metavlitwn ->r
noumero1=6*T + 3*M*T + 14*M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do



write(30,'(I8)') 8*M*N*T + 3*M*N



!eq.LHS



! 9o set periorismwn
noumero1=0
```

```
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1) + (j-1)*N*(T+1) + (k-
1)*(T+1) + i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1) + (j-1)*N*(T+1) + (k-
1)*(T+1) + i , -1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 2*M*N*T + (j-1)*N*T +
(k-1)*T + i , -Y

                End do
        End do
End do

! 12o set periorismwn
noumero1=M*N*T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i , -1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-1)*N*T + (k-
1)*T + i , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 3*M*N*T + (j-1)*N*T +
(k-1)*T + i , -G

                End do
        End do
End do

! 31o set periorismwn
noumero1=2*M*N*T
Do j=1,M
        Do k=1,N

                noumero1=noumero1+1
                i=1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i , 1.

        End do
End do

! 32o set periorismwn
noumero1=2*M*N*T + M*N
Do j=1,M
        Do k=1,N

                noumero1=noumero1+1
                i=1
```

```
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1) + (j-1)*N*(T+1) + (k-1)*(T+1) + i, 1.

        End do
End do

! 33o set periorismwn
noumero1=2*M*N*T + 2*M*N
Do j=1,M
        Do k=1,N

                noumero1=noumero1+1
                i=1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1) + (k-1)*(T+1) + i, 1.

        End do
End do


!elegxos gia midenika RHS sta set 31,32,33
midenika31=0
midenika32=0
midenika33=0

Do j=1,M
        Do k=1,N

                If(A1(j,k)==0.) Then
                        midenika31=midenika31+1
                End If

                If(Y1(j,k)==0.) Then
                        midenika32=midenika32+1
                End If

                If(G1(j,k)==0.) Then
                        midenika33=midenika33+1
                End If

        End do
End do

write(30,'(I8)') 3*M*N-(midenika31+midenika32+midenika33)



!eq.RHS




! RHS eq 31o set
noumero1=2*M*N*T
Do j=1,M
        Do k=1,N

                noumero1=noumero1+1

                If(A1(j,k)/=0.) Then
                        Write(30,'(I5,f21.6)') noumero1 , A1(j,k)
```

```
                End If

        End do
End do

write(*,*) midenika31,'midenika EQ.RHS apo to 31o set'



! RHS eq 32o set
noumero1=2*M*N*T + M*N
Do j=1,M
        Do k=1,N

                noumero1=noumero1+1

                If(Y1(j,k)/=0.) Then
                        Write(30,'(I5,f21.6)') noumero1 , Y1(j,k)
                End If

        End do
End do

write(*,*) midenika32,'midenika EQ.RHS apo to 32o set'



! RHS eq 33o set
noumero1=2*M*N*T + 2*M*N
Do j=1,M
        Do k=1,N
                noumero1=noumero1+1

                If(G1(j,k)/=0.) Then
                        Write(30,'(I5,f21.6)') noumero1 , G1(j,k)
                End If

        End do
End do

write(*,*) midenika33,'midenika EQ.RHS apo to 33o set'
write(*,*)
write(*,'(A8,I4,A40)') '!synolo' , midenika31 + midenika32 + midenika33 , 'midenika RHS apo
tous periorismous EQ!'


write(30,'(I8)') 16*M*N*T


!ge.LHS


! 10o set periorismwn
noumero1=0
Do j=1,M
        Do k=1,N
                Do i=1,T
```

```
                    noumero1=noumero1+1

                    Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 2*M*N*T + (j-1)*N*T +
(k-1)*T + i, 1.
                    Write(30,'(I5,I5,f17.6)') noumero1,        4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i
          +1 , -1.
             Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i , 1.

                    End do
          End do
End do

! 13o set periorismwn
noumero1=M*N*T
Do j=1,M
          Do k=1,N
                    Do i=1,T

                    noumero1=noumero1+1

                    Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + 3*M*N*T+ (j-1)*N*T +
(k-1)*T + i , 1.
                    Write(30,'(I5,I5,f17.6)') noumero1,        4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i , -
1.
             Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i    + 1 , 1.

                    End do
          End do
End do

! 15o set(a) periorismwn
noumero1=2*M*N*T
Do j=1,M
          Do i=1,T

                    noumero1=noumero1+1

                    Do k=1,N

                    Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + (j-1)*N*T + (k-1)*T + i ,
1.

                    End do
          End do
End do

! 17o set periorismwn
noumero1=2*M*N*T + M*T
Do i=1,T

          noumero1=noumero1+1

          Do j=1,M
                    Do k=1,N

                    Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , 1.

                    End do
          End do
End do
```

```
! 23o set periorismwn
noumero1=2*M*N*T + M*T + T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , 1.
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i , BIG
                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 3*M*N*(T+1) + M*N*T + (j-1)*N*T + (k-
1)*T + i , -BIG
                Write(30,'(I5,I5,f17.6)') noumero1,      4 + 3*M*N*(T+1) + 5*M*N*T + T + (j-
1)*N*T + (k-1)*T + i , BIG

                End do
        End do
End do

! 27o set periorismwn
noumero1=3*M*N*T + M*T + T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + M*N*(T+1) + (j-1)*N*(T+1) + (k-
1)*(T+1) + i + 1 , 1.
                write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 , -
Ymin

                End do
        End do
End do

! 28o set periorismwn
noumero1=4*M*N*T + M*T + T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,I5,f17.6)') noumero1, 4 + 2*M*N*(T+1) + M*N*T + (j-1)*N*(T+1)
+ (k-1)*(T+1) + i + 1 , 1.
                write(30,'(I5,I5,f17.6)') noumero1, 4 + (j-1)*N*(T+1) + (k-1)*(T+1) + i + 1 ,
Gmin

                End do
        End do
End do


!elegxos gia midenika RHS sto set 17
Do i=1,T

        If(M*N-C==0.) Then
                midenika17=midenika17+1
        End If
```

```
End do

write(30,'(I8)') 2*M*N*T + M*T + T - midenika17


!ge.RHS


! RHS ge 15o set (a)
noumero1=2*M*N*T
Do j=1,M
        Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , L*S(j,i)

        End do
End do


midenika17=0

! RHS 17o set
noumero1=2*M*N*T + M*T
Do i=1,T

                noumero1=noumero1+1

                If(M*N-C/=0.) Then
                        Write(30,'(I5,f21.6)') noumero1 , M*N-C
                Else
                        midenika17=midenika17+1

                End If

End do

write(*,*)
write(*,*) midenika17,'midenika GE.RHS apo to 17o set'


! RHS ge 23o set
noumero1=2*M*N*T + M*T + T
Do j=1,M
        Do k=1,N
                Do i=1,T

                noumero1=noumero1+1

                Write(30,'(I5,f21.6)') noumero1 , 1.

                End do
        End do
End do

! RHS ge 28o set
noumero1=4*M*N*T + M*T + T
Do j=1,M
        Do k=1,N
```

```
                    Do i=1,T

                    noumero1=noumero1+1

                    Write(30,'(I5,f21.6)') noumero1 , Gmin

                    End do
          End do
End do


write(30,'(I8)') 4


!obj.LHS


Write(30,'(I5,I5,f17.6)') 1, 1 , 1.
Write(30,'(I5,I5,f17.6)') 2, 2 , 1.
Write(30,'(I5,I5,f17.6)') 3, 3 , 1.
Write(30,'(I5,I5,f17.6)') 4, 4 , 1.


write(30,'(I8)') 0


Deallocate(B,S,A1,Y1,G1)

Close(20)

Close(30)


!elegxos arnitikotitas RHS 17ou set periorismwn (GE)
If(M*N-C < 0) Then
          Write(*,*)
          Write(*,'(A31,I5,A3,I5,A28)') '!!!PROSOXI : oi periorismoi GE' , 2*M*N*T + M*T + 1 ,
'ws' , 2*M*N*T + M*T + T , 'prepei na metatrapoun se LE'
          Write(*,'(A36)')                    'giati to RHS tous einai arnitiko!!!'
End if

End
```

Ο κώδικας της C++ είναι ο εξής:

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>


main()
{
  int nomb, nobjs, n1, ik, ie, is, ifase0, ngrays;
        int  i, j, k, t;
        long double x;
        FILE *fptr1, *fptr2;

        fptr1 = fopen("fmp_fortran.ii", "r");
        fptr2 = fopen("fmp.ii", "w");

        fscanf(fptr1, "%d%d%d\n", &i, &j, &k);
        fprintf(fptr2, "FMP Problem with M=%d , N=%d , T=%d \n", i, j, k);
        fscanf(fptr1, "%8d%8d%8d%8d%8d%8d%8d%8d\n", &nomb, &nobjs, &n1, &ik, &ie,
&is, &ifase0, &ngrays);
        fprintf(fptr2, "%8d%8d%8d%8d%8d%8d%8d%8d\n", nomb, nobjs, n1, ik, ie, is,
ifase0, ngrays);

        fscanf(fptr1, "%8d\n", &k);
  fprintf(fptr2, "%8d  .LE.LHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%5d%17Lf\n", &j, &t, &x);
                fprintf(fptr2, "%4d%4d%17Lf", j, t, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
  fprintf(fptr2, "%8d  .LE.RHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%21Lf\n", &j, &x);
                fprintf(fptr2, "%4d%21Lf", j, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
  fprintf(fptr2, "%8d  .EQ.LHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%5d%17Lf\n", &j, &t, &x);
                fprintf(fptr2, "%4d%4d%17Lf", j, t, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
  fprintf(fptr2, "%8d  .EQ.RHS.COEFFICIENTS\n", k);
        i = 1;
```

```
        while (i <= k){
                fscanf(fptr1, "%5d%21Lf\n", &j, &x);
                fprintf(fptr2, "%4d%21Lf", j, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
    fprintf(fptr2, "%8d  .GE.LHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%5d%17Lf\n", &j, &t, &x);
                fprintf(fptr2, "%4d%4d%17Lf", j, t, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
    fprintf(fptr2, "%8d  .GE.RHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%21Lf\n", &j, &x);
                fprintf(fptr2, "%4d%21Lf", j, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fscanf(fptr1, "%8d\n", &k);
        fprintf(fptr2, "%8d  .OBJ.LHS.COEFFICIENTS\n", k);
        i = 1;
        while (i <= k){
                fscanf(fptr1, "%5d%5d%17Lf\n", &j, &t, &x);
                fprintf(fptr2, "%4d%4d%17Lf", j, t, x);
                if((i % 4) == 0) fprintf(fptr2,"\n");
                i = i + 1;
        }

        if(((i-1) % 4) > 0) fprintf(fptr2,"\n");
        fprintf(fptr2, "       0  .OBJ.LHS.CONSTANTS\n");

    fclose(fptr1);
    fclose(fptr2);
}
```

Άλλα λυμένα προβλήματα:

Σε αυτό το κομμάτι παρατίθενται διάφορα λυμένα προβλήματα FMP που παρουσιάζουν ενδιαφέρον ως προς τις τιμές των παραμέτρων L, U και φυσικά ως προς τα αποτελέσματα τους.

Πρόβλημα με $M=1$, $N=1$, $T=1$ και με επιτρεπόμενες αποκλίσεις 5%

M=1
N=1
T=1
G=320
Y=300
C=1
L=0.95
U=1.05
X_{max}=170
Y_{min}=0.1
G_{min}=0.1
K=496

B_1=495

S_{11}=131

$A1_{11}$=1

$Y1_{11}$=250

$G1_{11}$=0

Λύσεις ADBASE

FMP Problem with M=1 , N=1 , T=1

EXTREME POINT	CRITERION VALUES		NONZERO STRUCTURAL BASIC VARIABLE VALUES	
1	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000

		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
2	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(16) = 0.496982
3	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(17) = 1.000000
6	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 124.449997
		X(13) = 0.581500
7	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(15) = 0.004024
14	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(16) = 0.496982

		X(17) = 1.000000
17	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 124.449997
		X(13) = 0.581500
		X(16) = 0.496982
18	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(15) = 0.004024
		X(16) = 0.496982
25	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 124.449997
		X(13) = 0.581500
		X(17) = 1.000000
26	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 137.549988
		X(13) = 0.625167
		X(15) = 0.004024
		X(17) = 1.000000
46	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 250.000000
		X(8) = 300.000000
		X(9) = 124.449997
		X(13) = 0.581500

			X(15) =	0.004024
61	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000
			X(8) =	300.000000
			X(9) =	124.449997
			X(13) =	0.581500
			X(16) =	0.496982
			X(17) =	1.000000
63	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000
			X(8) =	300.000000
			X(9) =	137.549988
			X(13) =	0.625167
			X(15) =	0.004024
			X(16) =	0.496982
			X(17) =	1.000000
83	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000
			X(8) =	300.000000
			X(9) =	124.449997
			X(13) =	0.581500
			X(15) =	0.004024
			X(16) =	0.496982
97	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000
			X(8) =	300.000000
			X(9) =	124.449997
			X(13) =	0.581500
			X(15) =	0.004024
			X(17) =	1.000000
141	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	250.000000

```
X( 8) =  300.000000
X( 9) =  124.449997
X( 13) =   0.581500
X( 15) =   0.004024
X( 16) =   0.496982
X( 17) =   1.000000
```

Λύσεις AMPL (γραμμικό πρόβλημα)

Input = 0.004
Solve = 0
Output = 0.001

Optimal solution: objective 450

z1 = 1
z2 = 300
z3 = 1
z4 = 300

a :=
m1 n1 1 1
m1 n1 2 1

y :=
m1 n1 1 250
m1 n1 2 300

x :=
m1 n1 1 124.45

g :=
m1 n1 1 0
m1 n1 2 0

h :=
m1 n1 1 0

d :=
m1 n1 1 0
m1 n1 2 0.5815

f :=
m1 n1 1 0
m1 n1 2 0

q [*] :=
1 0

p :=
m1 n1 1   0

r :=
m1 n1 1   1

Λύσεις AMPL (ακέραιο πρόβλημα)

Input = 0.005999
Solve = 0
Output = 0.000999

Optimal integer solution: objective 275.55

z1 = 1
z2 = 125.55
z3 = 1
z4 = 125.55

a :=
m1 n1 1   1
m1 n1 2   1

y :=
m1 n1 1   250
m1 n1 2   125.55

x :=
m1 n1 1   124.45

g :=
m1 n1 1   0
m1 n1 2   0

h :=
m1 n1 1   0

d :=
m1 n1 1   0
m1 n1 2   0

f :=
m1 n1 1   0
m1 n1 2   0

q [*] :=
1 0

p :=

m1 n1 1 0

r :=
m1 n1 1 0

Πρόβλημα με *M=1*, *N=1*, *T=1* χωρίς επιτρεπόμενες αποκλίσεις (*L=U=1*)

M=1
N=1
T=1
G=320
Y=300
C=1
L=1
U=1
X_{max}=170
Y_{min}=0.1
G_{min}=0.1
K=585

B_1=585

S_{11}=89

$A1_{11}$=1

$Y1_{11}$=281

$G1_{11}$=0

Λύσεις ADBASE

FMP Problem with M=1 , N=1 , T=1

EXTREME POINT	CRITERION VALUES		NONZERO STRUCTURAL BASIC VARIABLE VALUES	
1	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	281.000000
			X(8) =	300.000000
			X(9) =	89.000000
			X(13) =	0.360000

2	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 281.000000
		X(8) = 300.000000
		X(9) = 89.000000
		X(13) = 0.360000
		X(16) = 0.520478
3	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 281.000000
		X(8) = 300.000000
		X(9) = 89.000000
		X(13) = 0.360000
		X(17) = 1.000000
7	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 281.000000
		X(8) = 300.000000
		X(9) = 89.000000
		X(13) = 0.360000
		X(15) = 0.001706
14	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 281.000000
		X(8) = 300.000000
		X(9) = 89.000000
		X(13) = 0.360000
		X(16) = 0.520478
		X(17) = 1.000000
18	Z(1) = 1.000000	X(1) = 1.000000
	Z(2) = 300.000000	X(2) = 300.000000
	Z(3) = 1.000000	X(3) = 1.000000
	Z(4) = 300.000000	X(4) = 300.000000
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 281.000000
		X(8) = 300.000000
		X(9) = 89.000000
		X(13) = 0.360000
		X(15) = 0.001706
		X(16) = 0.520478

26	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	281.000000
			X(8) =	300.000000
			X(9) =	89.000000
			X(13) =	0.360000
			X(15) =	0.001706
			X(17) =	1.000000
63	Z(1) =	1.000000	X(1) =	1.000000
	Z(2) =	300.000000	X(2) =	300.000000
	Z(3) =	1.000000	X(3) =	1.000000
	Z(4) =	300.000000	X(4) =	300.000000
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	281.000000
			X(8) =	300.000000
			X(9) =	89.000000
			X(13) =	0.360000
			X(15) =	0.001706
			X(16) =	0.520478
			X(17) =	1.000000

Λύσεις AMPL (γραμμικό πρόβλημα)

Input = 0.003999 ίδιο με 1 από ss
Solve = 0.001
Output = 0.001

Optimal solution: objective 450

z1 = 1
z2 = 300
z3 = 1
z4 = 300

a :=
m1 n1 1 1
m1 n1 2 1

y :=
m1 n1 1 281
m1 n1 2 300

x :=
m1 n1 1 89

g :=
m1 n1 1 0
m1 n1 2 0

h :=
m1 n1 1   0

d :=
m1 n1 1   0
m1 n1 2   0.36

f :=
m1 n1 1   0
m1 n1 2   0

q [*] :=
1 0

p :=
m1 n1 1   0

r :=
m1 n1 1   1

Λύσεις AMPL (ακέραιο πρόβλημα)

Input = 0.003999
Solve = 0.001
Output = 0

Optimal integer solution: objective 342

z1 = 1
z2 = 192
z3 = 1
z4 = 192

a :=
m1 n1 1   1
m1 n1 2   1

y :=
m1 n1 1   281
m1 n1 2   192

x :=
m1 n1 1   89

g :=
m1 n1 1   0
m1 n1 2   0

h :=

m1 n1 1 0

d :=
m1 n1 1 0
m1 n1 2 0

f :=
m1 n1 1 0
m1 n1 2 0

q [*] :=
1 0

p :=
m1 n1 1 0

r :=
m1 n1 1 0

Πρόβλημα με *M*=2, *N*=2, *T*=2 και με επιτρεπόμενες αποκλίσεις 5%

Σε αυτό το παράδειγμα οι επιτρεπόμενες αποκλίσεις είναι μικρότερες από το προηγούμενο (5%) ενώ και ο σταθμός συντήρησης έχει μεγαλύτερη χωρητικότητα (*C*=*3*).

M=2
N=2
T=2
G=320
Y=300
C=3
L=0.95
U=1.05
X_{max}=50
Y_{min}=0.1
G_{min}=0.1
K=961

B_1=400
B_2=500

S_{11}=35 S_{12}=35
S_{21}=40 S_{22}=30

$A1_{11}$=0 $A1_{12}$=1

$A1_{21}=1$ $A1_{22}=1$

$Y1_{11}=0$ $Y1_{12}=200$
$Y1_{21}=100$ $Y1_{22}=150$

$G1_{11}=300$ $G1_{12}=0$
$G1_{21}=0$ $G1_{22}=0$

Λύσεις ADBASE

FMP Problem with M=2 , N=2 , T=2

EXTREME POINT	CRITERION VALUES		NONZERO STRUCTURAL BASIC VARIABLE VALUES	
1	Z(1) =	3.999996	X(1) =	3.999996
	Z(2) =	1199.993501	X(2) =	1199.993501
	Z(3) =	1.999996	X(3) =	1.999996
	Z(4) =	599.994636	X(4) =	599.994636
			X(6) =	1.000000
			X(7) =	1.000000
			X(8) =	1.000000
			X(9) =	1.000000
			X(10) =	1.000000
			X(11) =	1.000000
			X(12) =	0.999998
			X(13) =	1.000000
			X(14) =	1.000000
			X(15) =	0.999999
			X(16) =	0.999996
			X(18) =	299.999965
			X(19) =	299.994636
			X(20) =	200.000000
			X(21) =	299.995294
			X(22) =	300.000000
			X(23) =	100.000000
			X(24) =	299.999294
			X(25) =	299.999946
			X(26) =	150.000000
			X(27) =	299.998943
			X(28) =	299.998919
			X(30) =	36.750000
			X(31) =	36.750000
			X(33) =	42.000000
			X(34) =	31.499998
			X(37) =	300.000000
			X(38) =	0.000002
			X(39) =	0.005721
			X(44) =	0.000756
			X(47) =	0.001134
			X(48) =	0.000002
			X(49) =	300.000000
			X(50) =	0.000002
			X(54) =	0.000756
			X(56) =	0.001129
			X(57) =	1.000000
			X(58) =	0.122482

X(59) =	0.455818
X(60) =	0.000016
X(61) =	0.806664
X(62) =	0.105002
X(63) =	0.499996
X(64) =	0.000005
X(66) =	0.000018
X(69) =	0.000002
X(71) =	0.000001
X(73) =	0.000001
X(74) =	0.000002
X(75) =	0.001042

NUMBER OF EFFICIENT BASES VISITED	=	1
NUMBER OF EFFICIENT EXTREME POINTS	=	1
NUMBER OF UNBOUNDED EFFICIENT EDGES	=	0

Λύσεις AMPL (γραμμικό πρόβλημα)

Input = 0.004999
Solve = 0.002999
Output = 0.001

Optimal solution: objective 1125

z1 = 4
z2 = 1200
z3 = 2
z4 = 600

a :=
m1 n1 1 0
m1 n1 2 1
m1 n1 3 1
m1 n2 1 1
m1 n2 2 1
m1 n2 3 1
m2 n1 1 1
m2 n1 2 1
m2 n1 3 1
m2 n2 1 1
m2 n2 2 1
m2 n2 3 1

y :=
m1 n1 1 0
m1 n1 2 300
m1 n1 3 300
m1 n2 1 200
m1 n2 2 300
m1 n2 3 300
m2 n1 1 100
m2 n1 2 300

m2 n1 3 300
m2 n2 1 150
m2 n2 2 300
m2 n2 3 300

x :=
m1 n1 1 0
m1 n1 2 0
m1 n2 1 33.25
m1 n2 2 33.25
m2 n1 1 38
m2 n1 2 28.5
m2 n2 1 0
m2 n2 2 0

g :=
m1 n1 1 300
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

h :=
m1 n1 1 300
m1 n1 2 0
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 0
m2 n2 2 0

d :=
m1 n1 1 0
m1 n1 2 1
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0.444167
m1 n2 3 0.110833
m2 n1 1 0
m2 n1 2 0.793333
m2 n1 3 0.095
m2 n2 1 0
m2 n2 2 0.5

m2 n2 3 0

f :=
m1 n1 1 0
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

q [*] :=
1 0.895833
2 0

p :=
m1 n1 1 1
m1 n1 2 0.6875
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 0
m2 n2 2 0

r :=
m1 n1 1 0.6875
m1 n1 2 1
m1 n2 1 1
m1 n2 2 1
m2 n1 1 1
m2 n1 2 1
m2 n2 1 1
m2 n2 2 1

Σε αυτό το πρόβλημα παρατηρούμε πως ενώ αρχικά οι λύσεις ταυτίζονται ως και τη μεταβλητή X(29) δηλαδή την x_{111}, από την επόμενη μεταβλητή (X(30) ή x_{112}) τα αποτελέσματα αρχίζουν να εμφανίζουν διαφορές. Μάλιστα βλέπουμε παρακάτω πως η ADBASE δουλεύει με μεγαλύτερη ακρίβεια, εφόσον κάποιες μεταβλητές (όπως η X(38), X(39), X(44), X(47), X(48)) στην AMPL εμφανίζονται ως μηδέν ενώ στην ADBASE έχουν κάποια τιμή, έστω και πού μικρή.

Λύσεις AMPL (ακέραιο πρόβλημα)

Input = 0.004
Solve = 0.001
Output = 0.001

Optimal integer solution: objective 666.875

z1 = 4
z2 = 617
z3 = 2
z4 = 183.5

a :=

m1	n1	1	0
m1	n1	2	1
m1	n1	3	1
m1	n2	1	1
m1	n2	2	1
m1	n2	3	1
m2	n1	1	1
m2	n1	2	1
m2	n1	3	1
m2	n2	1	1
m2	n2	2	1
m2	n2	3	1

y :=

m1	n1	1	0
m1	n1	2	300
m1	n1	3	266.75
m1	n2	1	200
m1	n2	2	166.75
m1	n2	3	166.75
m2	n1	1	100
m2	n1	2	100
m2	n1	3	71.5
m2	n2	1	150
m2	n2	2	112
m2	n2	3	112

x :=

m1	n1	1	0
m1	n1	2	33.25
m1	n2	1	33.25
m1	n2	2	0
m2	n1	1	0
m2	n1	2	28.5
m2	n2	1	38
m2	n2	2	0

g :=
m1 n1 1 300
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

h :=
m1 n1 1 300
m1 n1 2 0
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 0
m2 n2 2 0

d :=
m1 n1 1 0
m1 n1 2 1
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

f :=
m1 n1 1 0
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

q [*] :=
1 0
2 0

p :=

m1	n1	1	1
m1	n1	2	0
m1	n2	1	0
m1	n2	2	0
m2	n1	1	0
m2	n1	2	0
m2	n2	1	0
m2	n2	2	0

r :=

m1	n1	1	0
m1	n1	2	0
m1	n2	1	0
m1	n2	2	1
m2	n1	1	0
m2	n1	2	1
m2	n2	1	0
m2	n2	2	1

Πρόβλημα με $M=2$, $N=2$, $T=2$ χωρίς επιτρεπόμενες αποκλίσεις ($L=U=1$)

M=2
N=2
T=2
G=320
Y=300
C=3
L=1
U=1
$X_{max}=150$
$Y_{min}=0.1$
$G_{min}=0.1$
K=448

$B_1=407$
$B_2=447$

$S_{11}=58$ $S_{12}=39$
$S_{21}=51$ $S_{22}=68$

$A1_{11}=1$ $A1_{12}=1$
$A1_{21}=1$ $A1_{22}=0$

$Y1_{11}=220$ $Y1_{12}=200$
$Y1_{21}=237$ $Y1_{22}=0$

$G1_{11}=0$ $G1_{12}=0$
$G1_{21}=0$ $G1_{22}=276$

Λύσεις ADBASE

FMP Problem with M=2 , N=2 , T=2

EXTREME POINT	CRITERION VALUES	NONZERO STRUCTURAL BASIC VARIABLE VALUES
1	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227

2	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(29) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.460000
		X(59) = 0.333333
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
3	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000

		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(30) = 38.999998
		X(31) = 58.000000
		X(32) = 0.000002
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(58) = 0.130000
		X(59) = 0.526667
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
12	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(75) = 0.510022
		X(81) = 0.002227
13	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998

	Z(3) =	2.000000	X(3) =	2.000000
	Z(4) =	599.999998	X(4) =	599.999998
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	1.000000
			X(8) =	1.000000
			X(9) =	1.000000
			X(10) =	1.000000
			X(11) =	1.000000
			X(12) =	1.000000
			X(13) =	1.000000
			X(15) =	1.000000
			X(16) =	1.000000
			X(17) =	220.000000
			X(18) =	300.000000
			X(19) =	300.000000
			X(20) =	200.000000
			X(21) =	300.000000
			X(22) =	299.999998
			X(23) =	237.000000
			X(24) =	300.000000
			X(25) =	300.000000
			X(27) =	299.999998
			X(28) =	300.000000
			X(31) =	58.000000
			X(32) =	39.000000
			X(33) =	51.000000
			X(36) =	68.000000
			X(46) =	276.000000
			X(55) =	276.000000
			X(57) =	0.266667
			X(59) =	0.526667
			X(60) =	0.130000
			X(61) =	0.380000
			X(63) =	1.000000
			X(64) =	0.226667
			X(76) =	0.331849
			X(81) =	0.002227
14	Z(1) =	4.000000	X(1) =	4.000000
	Z(2) =	1199.999998	X(2) =	1199.999998
	Z(3) =	2.000000	X(3) =	2.000000
	Z(4) =	599.999998	X(4) =	599.999998
			X(5) =	1.000000
			X(6) =	1.000000
			X(7) =	1.000000
			X(8) =	1.000000
			X(9) =	1.000000
			X(10) =	1.000000
			X(11) =	1.000000
			X(12) =	1.000000
			X(13) =	1.000000
			X(15) =	1.000000
			X(16) =	1.000000
			X(17) =	220.000000
			X(18) =	300.000000
			X(19) =	300.000000
			X(20) =	200.000000
			X(21) =	300.000000
			X(22) =	299.999998
			X(23) =	237.000000
			X(24) =	300.000000

		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(77) = 0.554566
		X(81) = 0.002227
15	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(78) = 0.331849
		X(81) = 0.002227
16	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000

	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(79) = 0.472160
		X(81) = 0.002227
17	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000

		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(80) = 0.331849
		X(81) = 0.002227
18	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
		X(82) = 0.331849
19	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998

		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
		X(85) = 1.000000
21	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998

		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
		X(86) = 1.000000
24	Z(1) = 4.000000	X(1) = 4.000000
	Z(2) = 1199.999998	X(2) = 1199.999998
	Z(3) = 2.000000	X(3) = 2.000000
	Z(4) = 599.999998	X(4) = 599.999998
		X(5) = 1.000000
		X(6) = 1.000000
		X(7) = 1.000000
		X(8) = 1.000000
		X(9) = 1.000000
		X(10) = 1.000000
		X(11) = 1.000000
		X(12) = 1.000000
		X(13) = 1.000000
		X(15) = 1.000000
		X(16) = 1.000000
		X(17) = 220.000000
		X(18) = 300.000000
		X(19) = 300.000000
		X(20) = 200.000000
		X(21) = 300.000000
		X(22) = 299.999998
		X(23) = 237.000000
		X(24) = 300.000000
		X(25) = 300.000000
		X(27) = 299.999998
		X(28) = 300.000000
		X(31) = 58.000000
		X(32) = 39.000000
		X(33) = 51.000000
		X(36) = 68.000000
		X(46) = 276.000000
		X(55) = 276.000000
		X(57) = 0.266667
		X(59) = 0.526667
		X(60) = 0.130000
		X(61) = 0.380000
		X(63) = 1.000000
		X(64) = 0.226667
		X(81) = 0.002227
		X(89) = 0.385301

Λύσεις AMPL (γραμμικό πρόβλημα)

Input = 0.004
Solve = 0.002
Output = 0.001

Optimal solution: objective 1125

z1 = 4
z2 = 1200
z3 = 2
z4 = 600

a :=

m1	n1	1	1
m1	n1	2	1
m1	n1	3	1
m1	n2	1	1
m1	n2	2	1
m1	n2	3	1
m2	n1	1	1
m2	n1	2	1
m2	n1	3	1
m2	n2	1	0
m2	n2	2	1
m2	n2	3	1

y :=

m1	n1	1	220
m1	n1	2	300
m1	n1	3	300
m1	n2	1	200
m1	n2	2	300
m1	n2	3	300
m2	n1	1	237
m2	n1	2	300
m2	n1	3	300
m2	n2	1	0
m2	n2	2	300
m2	n2	3	300

x :=

m1	n1	1	58
m1	n1	2	39
m1	n2	1	0
m1	n2	2	0
m2	n1	1	51
m2	n1	2	68
m2	n2	1	0
m2	n2	2	0

g :=

m1	n1	1	0
m1	n1	2	0
m1	n1	3	0
m1	n2	1	0
m1	n2	2	0
m1	n2	3	0
m2	n1	1	0
m2	n1	2	0
m2	n1	3	0
m2	n2	1	276
m2	n2	2	0
m2	n2	3	0

h :=

m1	n1	1	0
m1	n1	2	0
m1	n2	1	0
m1	n2	2	0
m2	n1	1	0
m2	n1	2	0
m2	n2	1	276
m2	n2	2	0

d :=

m1	n1	1	0
m1	n1	2	0.46
m1	n1	3	0.13
m1	n2	1	0
m1	n2	2	0.333333
m1	n2	3	0
m2	n1	1	0
m2	n1	2	0.38
m2	n1	3	0.226667
m2	n2	1	0
m2	n2	2	1
m2	n2	3	0

f :=

m1	n1	1	0
m1	n1	2	0
m1	n1	3	0
m1	n2	1	0
m1	n2	2	0
m1	n2	3	0
m2	n1	1	0
m2	n1	2	0
m2	n1	3	0
m2	n2	1	0
m2	n2	2	0
m2	n2	3	0

q [*] :=
1 0.707589
2 0

p :=
m1 n1 1 0
m1 n1 2 0
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 1
m2 n2 2 0.330357

r :=
m1 n1 1 1
m1 n1 2 1
m1 n2 1 1
m1 n2 2 1
m2 n1 1 1
m2 n1 2 1
m2 n2 1 0.383929
m2 n2 2 1

Λύσεις AMPL (ακέραιο πρόβλημα)

Input = 0.004
Solve = 0
Output = 0.002

Optimal integer solution; objective 802.5

z1 = 4
z2 = 741
z3 = 2
z4 = 323

a :=
m1 n1 1 1
m1 n1 2 1
m1 n1 3 1
m1 n2 1 1
m1 n2 2 1
m1 n2 3 1
m2 n1 1 1
m2 n1 2 1
m2 n1 3 1
m2 n2 1 0
m2 n2 2 1
m2 n2 3 1

y :=
m1 n1 1 220
m1 n1 2 162
m1 n1 3 123
m1 n2 1 200
m1 n2 2 200
m1 n2 3 200
m2 n1 1 237
m2 n1 2 186
m2 n1 3 118
m2 n2 1 0
m2 n2 2 300
m2 n2 3 300

x :=
m1 n1 1 58
m1 n1 2 39
m1 n2 1 0
m1 n2 2 0
m2 n1 1 51
m2 n1 2 68
m2 n2 1 0
m2 n2 2 0

g :=
m1 n1 1 0
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 276
m2 n2 2 0
m2 n2 3 0

h :=
m1 n1 1 0
m1 n1 2 0
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 276
m2 n2 2 0

d :=
m1 n1 1 0
m1 n1 2 0

m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 1
m2 n2 3 0

f :=
m1 n1 1 0
m1 n1 2 0
m1 n1 3 0
m1 n2 1 0
m1 n2 2 0
m1 n2 3 0
m2 n1 1 0
m2 n1 2 0
m2 n1 3 0
m2 n2 1 0
m2 n2 2 0
m2 n2 3 0

q [*] :=
1 0
2 0

p :=
m1 n1 1 0
m1 n1 2 0
m1 n2 1 0
m1 n2 2 0
m2 n1 1 0
m2 n1 2 0
m2 n2 1 1
m2 n2 2 0

r :=
m1 n1 1 0
m1 n1 2 1
m1 n2 1 0
m1 n2 2 1
m2 n1 1 0
m2 n1 2 1
m2 n2 1 0
m2 n2 2 0

8.Βιβλιογραφία

1. Arguello, M. F., Bard J. F., & Yu G. (1997) "*Models and Methods for Managing Airline Irregular Operations Aircraft Routing*" Operations Research in the Airline Industry ed. Yu G., pp 1-45, Kluwer Academic Publishers, Boston, MA.

2. Barnhart C., Boland N. L., Clarke L. W., Johnson E. L., Nemhauser G. L. & Shenoi R. G. (1998) "*Flight String Models for Aircraft Fleeting and Routing*" Transportation Science, 32(3), 208-220.

3. van Buskirk C., Dawant B., Karsai G., Sprinkle J., Szokoli G., Suwanmongkol K. & Currer R. (2002) "*Computer-Aided Aircraft Maintenance Scheduling*" ISIS/Vanderbilt University, Technical Report.

4. Clarke L. W., Hane C. A., Johnson E. L. & Nemhauser G. L. (1996) "*Maintenance and Crew Considerations in Fleet Assignment*" Transportation Science, 30(3), 249-260.

5. Clarke L. W., Johnson E. L., Nemhauser G. L. & Zhongxi Z. (1997) "*The Aircraft Rotation Problem*" Annals of Operations Research, (69), 33-46.

6. Ehrgott M. (2000) "*Multicriteria Optimization*" Lecture notes in Economics and Mathematical Systems, Springer-Verlag, Berlin.

7. Feo T. A. & Bard J. F. (1989) "*Flight Scheduling and Maintenance Base Planning*" Management Science, (35), 1415-1432.

8. Gopalan R. & Talluri K. T. (1998a) "*The Aircraft Maintenance Routing Problem*" Operations Research, 46(2), 260-271.

9. Gopalan R. & Talluri K. T. (1998b) "*Mathematical Models in Airline Schedule Planning*" Annals if Operations Research, (76), 155-185.

10. International Society on Multiple Criteria Decision Making http://www.terry.uga.edu/mcdm/.

11. Klabjan D., Johnson E. L., Nemhauser G. L., Gelman E. & Ramaswamy S. (2002) "*Airline Crew Scheduling with Time Windows and Plane-Count Constraints*" Transportation Science, 36(3), 337-348.

12. Kozanidis G., Liberopoulos G. & Pitsilkas C. (2005) "*Flight and Maintenance Planning of Military Aircraft for Maximum Fleet Availability*" under review.

13. Kozanidis G. (2007) "*A Multiobjective Model for Maximizing Fleet Availability of a Unit under the Presence of Flight and Maintenance Requirements*" Technical Report, Department of Industrial And Mechanical Engineering, University of Thessaly, Volos.

14. Kozanidis G. (2008) "*Solving the Linear Multiple Choice Knapsack Problem with Two Objectives: Profit and Equity. Computational Optimization and Applications*" in press.

15. Mousseau V., Slovinski R. & Zielniewicz P. (2000) "*Presentation of the Electre tri 2.0a software*" European Working Group "Multicriteria Aid for Decisions" Series 3, n#2, Fall.

16. Rushmeier R. A. & Kontogiorgis S. A. (1997) "*Advances in the Optimization of Airline Fleet Assignment*" Transportation Science, 31(2), 159-169.

17. Steuer R. E. (2004) "*Multiple Criteria Optimization and Computation*" Terry College of Business, University of Georgia, Athens, Georgia, 30602-6253 USA.

18. Steuer R. E. (2006) "*ADBASE: A Multiple Objective Linear Programming Solver for Efficient Extreme Points and Unbounded Efficient Edges – Part 1: ADBASE 06 Users Manual*" Terry College of Business, University of Georgia, Athens, Georgia, 30602-6253 USA.

19. Steuer R. E. (2006) "*ADBASE: A Multiple Objective Linear Programming Solver for Efficient Extreme Points and Unbounded Efficient Edges – Part 2: Repertoire of MOLP Test Problems*" Terry College of Business, University of Georgia, Athens, Georgia, 30602-6253 USA.

20. Talluri K. T. (1998) "*The Four Day Aircraft Maintenance Problem*" Transportation Science, 32(1), 43-53.

21. The Association of European Operational Research Societies – Euro Working Group – Multicriteria Decision Aiding http://inescc.pt/~ewgmcda/index.html.

22. Weistroffer R. H. & Narula S. C. (1997) "*The State of Multiple Criteria Decision Support Software*" Annals of Operational Research, (72), 299-313.