ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ

ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία με θέμα:

Τετραγωνικός Προγραμματισμός σε Αρνητικά Ορισμένες Μήτρες

Αναστασία Ν. Βουλγαράκη

<u>Επιβλέπων Καθηγητής:</u>

Νίκος Μαράτος

Αθήνα, Ιούλιος 2004

....

Αναστασία Ν. Βουλγαράκη

Περιεχόμενα

Περίληψη

Σκοπός της διπλωματικής εργασίας ήταν η επίλυση ενός προβλήματος τετραγωνικού προγραμματισμού για αρνητικά ορισμένη μήτρα. Συγκεκριμένα ασχοληθήκαμε με τη λύση του παρακάτω προβλήματος:

$$min\{f(x) = \frac{1}{2}x^T Qx + c^T x : Ax \leq b, -1 \leq x_j \leq 1, j = 1...n\}$$

, όπου η μήτρα Q είναι αρνητικά ορισμένη. Λόγω της μορφής της μήτρας Q το πρόβλημα που εξετάσαμε ανάγεται σε ένα πρόβλημα ελαχιστοποίησης κοίλης συνάρτησης με κυρτούς περιορισμούς. Είναι γνωστό ότι τα προβλήματα αυτά μπορεί να έχουν πολλά τοπικά ελάχιστα, τα οποία μπορεί να μην είναι ολικά. Ωστόσο παρουσιάζουν ορισμένες ειδικές μαθηματικές ιδιότητες, γεγονός που τα κάνει πολύ ελκυστικά. Για την εύρεση του ολικού ελαχίστου εκμεταλλευτήκαμε τη μορφή των περιορισμών και ορισμένες βασικές ιδιότητες της κοίλης ελαχιστοποίησης για να καταλήξουμε ότι το πρόβλημα που μας απασχολεί έχει ολικό ελάχιστο και αυτό βρίσκεται σε μια από τις πεπερασμένες σε αριθμό κορυφές του συνόλου $\{x : -1 \leq x_j \leq 1, j = 1...n\}$.

Συγκεκριμένα για τη λύση του προβλήματος προσαρμόσαμε κατάλληλα τον αλγόριθμο των $Horst$ και Tuy. Ο αλγόριθμος αυτός ανήκει στις cone covering προσεγγίσεις, δηλαδή καλύπτει το χώρο με κώνους και στη συνέχεια χρησιμοποιεί τις τομές Tuy για να αποκλείσει διαδοχικά περιοχές του συνόλου D που δε χρειάζονται να εξεταστούν περαιτέρω, μέχρι να βρει την κορυφή που είναι η ολικά βέλτιστη λύση. Στην περίπτωση που οι μήτρες A και b δεν είναι κενές τότε εισάγονται κάποιοι επιπλέον περιορισμοί στο αρχικό πρόβλημα.

Λέξεις Κλειδιά

Τετραγωνικός προγραμματισμός, Αρνητικά ορισμένες μήτρες, Κοίλες συναρτήσεις, Κοίλη ελαχιστοποίηση, Τομές Tuy, Αλγόριθμος $Horst$ και Tuy.

Abstract

The scope of this thesis was to solve the following quadratic programming problem for a negative definite matrix:

$$min\{f(x) = \frac{1}{2}x^T Qx + c^T x : \ Ax \leq b, \ -1 \leq x_j \leq 1, \ j = 1...n\}$$

, where Q is a negative definite matrix and A and b, when they are nonempty matrices, introduce some extra linear inequality constraints. The problem described above is a concave minimization problem, where $D = \{x : Ax \leq b, \ -1 \leq x_j \leq 1, \ j = 1...n\}$ is a nonempty, closed convex set. Since concave minimization involves the minimization of concave functions, concave minimization problems generally will possess many solutions that are local, but not global, minima. Nevertheless they display some interesting and attractive mathematical properties that make them more tractable than general global optimization problems. In order to find a global optimal solution we made use of certain mathematical properties, which establish that, due to the fact that D is a polyhedron, the problem has a global optimal solution, which is an extreme point of D.

In order to solve the problem we made use of the Horst and Tuy algorithm. This algorithm is a cone covering method: It uses conical covers of D to search out and enumerate extreme points of D until a global optimal solution has been found. In order to do so it uses the famous Tuy cuts.

Key words

Quadratic programming, Negative definite matrix, Concave functions, Concave minimization, Tuy cuts, Horst and Tuy algorithm.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πρωτίστως τον Καθηγητή της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών και επιβλέποντα της εργασίας κο Νίκο Μαράτο για την πολύτιμη βοήθεια του στην από μέρους μου καλύτερη κατανόηση του προβλήματος και για την αμέριστη κατανόηση και υπομονή που επέδειξε καθόλη τη διάρκεια περάτωσης της παρούσας.

Θα ήθελα επίσης να ευχαριστήσω το κο Νίκο Ξηρό, Λέκτορα της σχολής Ναυπηγών Μηχανολόγων Μηχανικών για τις πολύτιμες υποδείξεις που μου έδωσε σε μια δύσκολη στιγμή κατά την εκπόνηση αυτής της διπλωματικής εργασίας, καθώς επίσης και τον κο Τρύφωνα Κουσιουρή, Καθηγητή της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Τέλος θα ήθελα να ευχαριστήσω το ευρύτερο συγγενικό και φιλικό μου περιβάλλον για τη βοήθεια και τη συμπαράσταση που μου επέδειξε. Ιδιαίτερα θα ήθελα να αναφέρω τους γονείς μου και την αδερφή μου, Δέσποινα, τον κο Κωνσταντίνο Σπηλιόπουλο, διπλωματούχο της σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών και τον ξάδερφο μου κο Ιωάννη Χατζηγεωργίου, Λέκτορα της σχολής Ναυπηγών Μηχανολόγων Μηχανικών.

Κεφάλαιο 1

Γενικά περί Τετραγωνικού Προγραμματισμού

Στο κεφάλαιο αυτό θα μιλήσουμε γενικά για τα προβλήματα τετραγωνικού προγραμματισμού και θα δοθούν ορισμένοι ορισμοί απαραίτητοι για τη συνέχεια.

Σκοπός της εργασίας αυτής είναι να λυθεί το πρόβλημα τετραγωνικού προγραμματισμού:

$$min\{f(x) = \frac{1}{2}x^T Qx + c^T x : Ax \leq b, -1 \leq x_j \leq 1, j = 1...n\} \tag{1.1}$$

όπου η μήτρα Q είναι αρνητικά ορισμένη. Υπενθυμίζουμε τους εξής ορισμούς [1]:

Ορισμός 1

Η συνάρτηση $F(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij}x_i x_j = x^T Qx$ λέγεται **τετραγωνική μορφή**.

Ορισμός 2

Η τετραγωνική μορφή $x^T Qx$ και η αντίστοιχη μήτρα Q, όπου $Q = Q^T$, λέγεται **αρνητικά ορισμένη** αν $x^T Qx < 0$, $\forall x \neq 0$ ή ισοδύναμα όταν $\lambda_i < 0$, $i = 1, 2...n$, όπου λ_i είναι οι ιδιοτιμές του Q.

Ορισμός 3

Η τετραγωνική μορφή x^TQx και η αντίστοιχη μήτρα Q, όπου $Q = Q^T$, λέγεται **αρνητικά ημιορισμένη** αν $x^TQx \leq 0$, $\forall x \in \mathbb{R}^n$ και υπάρχει $\tilde{\mathrm{x}} \neq 0$ τέτοιο ώστε $\tilde{\mathrm{x}}\mathrm{Q}\tilde{\mathrm{x}}$=0 ή ισοδύναμα όταν $\lambda_i \leq 0$, $i = 1, 2...n$ και υπάρχει $\lambda_j = 0$, όπου λ_i είναι οι ιδιοτιμές του Q.

Αντίστοιχα ορίζονται οι έννοιες **θετικά ορισμένη** και **θετικά ημιορισμένη**.

Ο τετραγωνικός προγραμματισμός αποτελεί μια από τις πιο σημαντικές περιοχές του μη γραμμικού προγραμματισμού. Πολυάριθμα προβλήματα που προκύπτουν σε εφαρμογές του ελέγχου, των οικονομικών, της θεωρίας παιγνίων και του μηχανολογικού σχεδίου μπορούν να εκφρασθούν ως προβλήματα τετραγωνικού προγραμματισμού. Π.χ ορισμένες πλευρές της σχεδίασης VLSI κυκλωμάτων μπορούν να εκφραστούν ως γραμμικά, με τετραγωνικούς περιορισμούς, τετραγωνικά μοντέλα.

Συνεπώς είναι φανερό ότι το πρόβλημα τετραγωνικού προγραμματισμού είναι μεγάλης σημασίας τόσο από μαθηματικής πλευράς όσο και από πλευράς εφαρμογών. Αυτό αντικαθρεφτίζεται στο μεγάλο αριθμό προσεγγίσεων που έχουν προταθεί για τη λύση αυτών των προβλημάτων. Παραδοσιακά οι μέθοδοι για τη λύση αυτών των προβλημάτων έχουν βασιστεί σε τεχνικές εύρεσης τοπικού ελαχίστου. Ωστόσο τα τελευταία χρόνια έχουν γίνει προσπάθειες για την κατασκευή αλγορίθμων εύρεσης ολικού ελαχίστου για το συγκεκριμένο πρόβλημα.

Οι προσεγγίσεις για εύρεση ολικού ελαχίστου για το γενικό μη κυρτό, μη γραμμικό πρόβλημα (συμπεριλαμβανομένου και της βελτιστοποίησης τετραγωνικών συναρτήσεων) μπορούν να καταταχθούν σε δυο κατηγορίες: Τις ντετερμινιστικές και τις στοχαστικές. Για την επίλυση του 1.1 θα χρησιμοποιηθούν ντετερμινιστικές μέθοδοι και συγκεκριμένα θα γίνει εφαρμογή του περίφημου αλγορίθμου των Horst καιTuy.

Το γενικό τετραγωνικό πρόβλημα αποτελείται από μια τετραγωνική αντικειμενική συνάρτηση και ένα σύνολο γραμμικών ανισοτικών περιορισμών της μορφής:

$$min\{F(x) = \frac{1}{2}x^TQx + c^Tx : Ax \leq b,\ x \geq 0\} \tag{1.2}$$

,όπου c είναι ένα $n \times 1$ διάνυσμα, A είναι ένας $m \times n$ πίνακας και Q είναι ένας $n \times n$ πίνακας. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι ο Q είναι συμμετρικός. Αν αυτό δεν ισχύει τότε μπορεί να μετατραπεί σε συμμετρική μορφή αν αντικαταστήσουμε το Q από το $(Q+Q^T)/2$, μια ενέργεια που αποδεικνύεται ότι δεν αλλάζει την τιμή της συνάρτησης F(x). Αντίστοιχα οποιοδήποτε πρόβλημα, στο οποίο οι μεταβλητές δεν είναι απαραίτητα μη αρνητικές, μπορεί να μετατραπεί με ένα γραμμικό μετασχηματισμό στη μορφή 1.2.

Αν ο πίνακας Q είναι θετικά ημιορισμένος ή θετικά ορισμένος τότε το 1.2 είναι σε ένα κυρτό πρόβλημα προγραμματισμού. Είναι γνωστό ότι σε αυτή την κατηγορία προβλημάτων κάθε τοπικό ελάχιστο είναι και ολικό ελάχιστο και συνεπώς η 1.2 μπορεί να λυθεί από οποιονδήποτε γνωστό αλγόριθμο για κυρτό τετραγωνικό προγραμματισμό. Λόγω της ύπαρξης ενός μεγάλου αριθμού αλγορίθμων για κυρτό, μη γραμμικό προγραμματισμό οι περιπτώσεις που παρουσιάζουν σήμερα ενδιαφέρον από την πλευρά της εύρεσης ολικού ελαχίστου είναι οι περιπτώσεις στις οποίες η μήτρα Q δεν είναι θετικά ημιορισμένη. Τότε το 1.2 είναι ένα μη κυρτό πρόβλημα και η εφαρμογή τεχνικών εύρεσης τοπικού ελαχίστου για αυτό το πρόβλημα δεν μπορούν πλέον να εγγυηθούν και εύρεση του γενικού ελαχίστου. Ειδικά στην περίπτωση που ο πίνακας Q είναι αρνητικά ημιορισμένος ή αρνητικά ορισμένος τότε βρισκόμαστε στην περιοχή των κοίλων προβλημάτων.

Κεφάλαιο 2

Ελαχιστοποίηση Κοίλων Συναρτήσεων

2.1 Εισαγωγή

Το πρόβλημα ελαχιστοποίησης κοίλων συναρτήσεων αποτελεί ένα από τα πιο βασικά και περισσότερο μελετημένα προβλήματα εύρεσης ολικού ελαχίστου. Στην εξίσωση 1.1 η μήτρα Q είναι αρνητικά ορισμένη, πράγμα το οποίο σημαίνει ότι το πρόβλημα που μας ενδιαφέρει είναι ένα πρόβλημα κοίλης ελαχιστοποίησης, και συνεπώς κρίνεται αναγκαίο να παρουσιαστούν εν συντομία κάποιες βασικές ιδιότητες, οι πιο συνήθεις εφαρμογές αλλά και οι γενικές μέθοδοι αντιμετώπισης αυτού του προβλήματος. Επίσης στο κεφάλαιο αυτό συμπεριλαμβάνεται μια ιστορική ανασκόπηση, όπου θα παρουσιαστούν τα πιο σημαντικά ερευνητικά αποτελέσματα.

2.2 Γενικά

Υπενθυμίζουμε τους παρακάτω ορισμούς:

Ορισμός 1

Το σύνολο $A \subseteq \mathbb{R}^n$ λέγεται **κυρτό** αν για κάθε ζευγάρι σημείων $x \in A$, $y \in A$ και

$\forall \lambda \in [0,1]$ ισχύει $\lambda x + (1-\lambda)y \in A$.

Ορισμός 2

Η συνάρτηση $f : A \subseteq \mathbb{R}^n$, όπου το A είναι κυρτό σύνολο, λέγεται **κυρτή** αν για κάθε ζευγάρι σημείων $x \in A$, $y \in A$ και $\forall \lambda \in [0,1]$ ισχύει $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$.

Ορισμός 3

Μια συνάρτηση f λέγεται **κοίλη** αν η $-f$ είναι κυρτή .

Στη συνέχεια παρουσιάζονται μερικές βασικές ιδιότητες των κυρτών συναρτήσεων και των κυρτών συνόλων:

Ιδιότητα 1

Αν $f_1(x)$ και $f_2(x)$ είναι κυρτές συναρτήσεις τότε η συνάρτηση $\alpha_1 f_1(x) + \alpha_2 f_2(x)$ είναι κυρτή εφ' όσον $\alpha_1 \geq 0$ και $\alpha_2 \geq 0$.
Η ιδιότητα αυτή ισχύει και όταν οι συναρτήσεις $f_1(x)$ και $f_2(x)$ είναι κοίλες.

Ιδιότητα 2

Αν $f(x)$ είναι κυρτή συνάρτηση τότε το σύνολο $\{x \in \mathbb{R}^n : f(x) \leq c\}$ είναι κυρτό για κάθε $c \in \mathbb{R}$.

Ιδιότητα 3

Αν τα σύνολα B_1 και B_2 είναι κυρτά τότε και το σύνολο $B_1 \cap B_2$ είναι κυρτό.

Επίσης παραθέτουμε το πολύ σημαντικό θεώρημα $Weierstrass$ που δίνει τις ικανές συνθήκες για την ύπαρξη γενικού ελαχίστου.

Θεώρημα $Weierstrass$

Αν η $f : \mathbb{R}^n \to \mathbb{R}$ είναι συνεχής και το $A \subset \mathbb{R}^n$ είναι κλειστό και φραγμένο, τότε η f έχει τουλάχιστον ένα ελάχιστο πάνω στο A.

Σκοπός των αλγορίθμων ελαχιστοποίησης κοίλων συναρτήσεων είναι να βρουν το ολικό ελάχιστο μιας πραγματικής (και διαφορίσιμης) κοίλης συνάρτησης πάνω σε κλειστά κυρτά σύνολα. Γενικά, όπως και σε άλλα προβλήματα ελαχιστοποίησης, μπορεί το πρόβλημα να έχει πολλά τοπικά ελάχιστα, που όμως δεν είναι ολικά. Ακόμα και φαινομενικά απλά προβλήματα κοίλων συναρτήσεων μπορεί να έχουν εκθετικό αριθμό τοπικών ελαχίστων. Ωστόσο αυτό που τα κάνει τόσο ελκυστικά είναι το γεγονός ότι οι κοίλες συναρτήσεις και η ελαχιστοποίηση κοίλων συναρτήσεων παρουσιάζουν ορισμένες ειδικές μαθηματικές ιδιότητες. Επίσης έχουν ένα πολύ ευρύ πεδίο άμεσων και έμμεσων εφαρμογών.

Το γενικό πρόβλημα που μας απασχολεί είναι της μορφής:

$$glob\ minf(x)\ \acute{o}\tau\alpha\nu\ x \in D \tag{P}$$

,όπου το $D \subseteq \mathbb{R}^n$ είναι ένα μη κενό, κυρτό σύνολο και f είναι μια πραγματική, κοίλη συνάρτηση ορισμένη σε ένα κατάλληλο κυρτό σύνολο $A \subseteq \mathbb{R}^n$ που περιέχει το D. Σκοπός είναι να βρεθεί η ελάχιστη τιμή της f πάνω στο D και αν αυτή η τιμή δεν είναι ίση με το $-\infty$ να βρεθεί, αν υπάρχει, τουλάχιστον ένα σημείο στο D που επιτυγχάνει αυτή την τιμή. Στις περισσότερες εφαρμογές το A είναι ίσο είτε με το $\mathbb{R}^n$ είτε τουλάχιστον με ένα ανοικτό κυρτό σύνολο που περιέχει το D και το D είναι συμπαγές. Κάτω από αυτούς τους περιορισμούς η συνάρτηση f είναι συνεχής στο A και σύμφωνα με το θεώρημα $Weierstrss$ το ολικό ελάχιστο είναι πεπερασμένο και υπάρχει τουλάχιστον ένα σημείο στο D που επιτυγχάνει αυτή την τιμή. Ωστόσο υπάρχουν σημαντικές εφαρμογές όπου το A δεν ισούται με το $\mathbb{R}^n$ ή με ένα ανοικτό κυρτό σύνολο που περιέχει το D, ή το D δεν είναι συμπαγές.

Το σύνολο D μπορεί να καθοριστεί πλήρως από ένα σύστημα γραμμικών ή/και μη γραμμικών ισοτικών και ανισοτικών περιορισμών. Συνεπώς μπορούμε να υποθέσουμε ότι γενικά το D είναι της μορφής:

$$D = \{x \in A|\ g_i(x) \geq 0,\ i = 1, 2, ..., m\},$$

όπου για κάθε $i = 1, 2, ..., m : A \rightarrow \mathbb{R}$.

Όπως αναφέραμε παραπάνω τα προβλήματα κοίλης ελαχιστοποίησης αφορούν την ελαχιστοποίηση κοίλων, και όχι κυρτών, συναρτήσεων πάνω σε κυρτά σύνολα και γι΄ αυτό έχουν γενικά πολλές λύσεις x_* που είναι τοπικά αλλά όχι και ολικά ελάχιστα. Αυτές οι λύσεις ανήκουν στο D και παίρνουν τιμές $f(x_*)$, που είναι οι ελάχιστες τιμές που μπορεί να πάρει η συνάρτηση f πάνω σε διάφορες υποπεριοχές $(N_{x_*} \cap D)$ του D, όπου N_{x_*} είναι ένα ανοικτό σύνολο που περιέχει το x_*, αλλά οι τιμές της $f(x_*)$ δεν είναι και το ολικό ελάχιστο της f πάνω στο D. Γι' αυτό το λόγο τα προβλήματα κοίλης ελαχιστοποίησης καλούνται και *multiextremal* προβλήματα εύρεσης ολικού ελαχίστου. Η εφαρμογή των κοινών αλγορίθμων που έχουν σχεδιαστεί για την εύρεση τοπικών ελαχίστων κυρτών προβλημάτων με περιορισμούς θα αποτύχουν γενικά να λύσουν το πρόβλημα (P). Στο σημείο αυτό θα πρέπει να σημειωθεί ότι αφού μια κυρτή συνάρτηση g μπορεί να μεγιστοποιηθεί πάνω σε ένα κυρτό σύνολο Y ελαχιστοποιώντας την $-g$ πάνω στο Y τα κυρτά προβλήματα μεγιστοποίησης είναι ουσιαστικά ισοδύναμα με τα προβλημάτων κοίλης ελαχιστοποίησης.

Συνοψίζοντας μπορούμε να πούμε ότι τα προβλήματα ελαχιστοποίησης κοίλων συναρτήσεων πάνω σε κυρτά σύνολα κατέχουν τέσσερα σημαντικά χαρακτηριστικά που δικαιολογούν το αυξανόμενο ενδιαφέρον για αυτή την κατηγορία της θεωρίας βελτιστοποίησης:

1. Παρουσιάζουν ορισμένες ενδιαφέρουσες και πολύ ελκυστικές μαθηματικές ιδιότητες που τα κάνουν πιο εύκολα προς λύση από άλλα γενικά προβλήματα εύρεσης ολικού ελαχίστου.

2. Έχουν ένα ευρύ πεδίο απευθείας εφαρμογών.

3. Πολλά μοντέλα, αν και δεν είναι αρχικά προβλήματα κοίλης ελαχιστοποίησης, μπορούν να μετασχηματιστούν σε τέτοια προβλήματα ή μπορούν να λυθούν μέσω κοίλης ελαχιστοποίησης.

4. Πολλές τεχνικές που χρησιμοποιούνται σε αλγόριθμους που λύνουν προβλήματα κοίλης ελαχιστοποίησης παίζουν ένα πολύ σημαντικό ρόλο για τη λύση και άλλων κατηγοριών προβλημάτων εύρεσης ολικού ελαχίστου.

2.3 Ανασκόπηση Βιβλιογραφίας

Τα προβλήματα ελαχιστοποίησης κοίλων συναρτήσεων προκάλεσαν από πολύ νωρίς το ενδιαφέρον, αποδείξεις του οποίου μπορούν να βρεθούν από πολλές πηγές. Για παράδειγμα εγχειρίδια και εξειδικευμένα βιβλία που επικεντρώνονται κυρίως στο κυρτό προγραμματισμό και στην ανάλυση κυρτών προβλημάτων αφιερώνουν επίσης κάποιο κομμάτι στην ανάλυση μη κυρτών προβλημάτων. Στην περίπτωση αυτή η ελαχιστοποίηση κοίλων συναρτήσεων κατείχε πάντα εξέχουσα θέση (π.χ. [6], [7], [8], [9]).

Μια δυσκολία κατά την ελαχιστοποίηση κοίλων συναρτήσεων είναι ότι δεν έχει βρεθεί κάποιο απλό τοπικό κριτήριο για το γενικό πρόβλημα (P) που να διευκρινίζει κατά πόσο ένα τοπικό ελάχιστο είναι και ολικό ελάχιστο. Έτσι ακόμα και φαινομενικά απλές περιπτώσεις κοίλης ελαχιστοποίησης πάνω σε συμπαγή πολύεδρα μπορεί να περιπλεχτούν από την ύπαρξη ενός εκθετικού αριθμού ακραίων σημείων του επιτρεπτού συνόλου που είναι τοπικά ελάχιστα. Επιπλέον σε κάθε πρόβλημα κοίλης ελαχιστοποίησης ακριβώς ένα, μερικά αλλά όχι όλα, ή και όλα τα τοπικά ελάχιστα μπορεί να είναι και ολικά ελάχιστα. Για αυτό το λόγο έγιναν προσπάθειες για την εύρεση κάποιου κριτηρίου που θα διευκολύνει τη λύση για ορισμένες ειδικές περιπτώσεις προβλημάτων αυτής της κατηγορίας. Για παράδειγμα στο [10] παρουσιάζονται κάποια επαρκή κριτήρια που αποδεικνύουν ότι η λύση ενός διαχωρίσιμου προβλήματος κοίλης ελαχιστοποίησης πάνω σε ένα φραγμένο πολύεδρο, η οποία είναι ακραίο σημείο, είναι επίσης και ολικό ελάχιστο, ενώ στα [11] και [12] αναλύονται οι αναγκαίες και ικανές συνθήκες καθώς επίσης και οι ικανές συνθήκες έτσι ώστε ένα τοπικό ελάχιστο ενός κοίλου τετραγωνικού προβλήματος ελαχιστοποίησης να είναι επίσης και ολικό ελάχιστο.

Ταυτόχρονα από πολύ νωρίς έγιναν δημοσιεύσεις για απευθείας εφαρμογές της ελαχιστοποίησης κοίλων συναρτήσεων. Για παράδειγμα ήδη από τα μέσα της δεκαετίας του '50 ερευνητές διατύπωσαν προβλήματα που αφορούσαν τη βέλτιστη επιλογή θέσης (π.χ. [13], [14]), τη διαχείριση απογραφής (π.χ. [15], [16]), την αξιολόγηση προσφορών (π.χ. [17]), τις πάγιες επιβαρύνσεις (π.χ. [18], [19], [20]) ως προβλήματα ελαχιστοποίησης κοίλων συναρτήσεων.

Οι πρώτες προσπάθειες για να λυθεί αλγοριθμικά το πρόβλημα ελαχιστοποίησης κοίλων συναρτήσεων χρονολογούνται από τα μέσα της δεκαετίας του '60. Ενδεικτικά αναφέρουμε τα [21], [22], [23]. Από τότε μέχρι σήμερα έχουν παρουσιαστεί μια σειρά αλγορίθμων για την επίλυση κοίλων προβλημάτων και έχουν προταθεί πολλές πιθανές εφαρμογές. Το ενδιαφέρον των ερευνητών για ειδικές κατηγορίες της κοίλης ελαχιστοποίησης παραμένει ισχυρό μέχρι σήμερα.

Οι αλγοριθμικές μέθοδοι που μπορούν να χρησιμοποιηθούν για την επίλυση αυτών των προβλημάτων διακρίνονται στις ντετερμινιστικές και στις στοχαστικές. Στην εργασία αυτή χρησιμοποιήθηκαν μόνο ντετερμινιστικές μέθοδοι ωστόσο υπάρχουν πολλά ενδιαφέροντα άρθρα για εύρεση γενικού ελαχίστου με μεθόδους στοχαστικής βελτιστοποίησης κατάλληλες για γενικά προβλήματα ελαχιστοποίησης κοίλων συναρτήσεων π.χ. από τους *Aarts*, *Boender* και *Romeijn* καθώς επίσης και από τους *Ratschek* και *Roekne* στο [2]. Επίσης στα βιβλία [24], [25], [26]. Τέλος στο [2] μπορούν να βρεθούν άρθρα από τους *Floudas* και *Visweswaran*, *Du*, *Guisewite*, και *Konno* και *Kuno* αντίστοιχα που περιγράφουν εξειδικευμένες αλγοριθμικές προσπάθειες για τη λύση ορισμένων υποκατηγοριών των προβλημάτων κοίλης ελαχιστοποίησης, όπως για παράδειγμα προβλήματα ελαχιστοποίησης κοίλων τετραγωνικών συναρτήσεων, ορισμένα $min - max$ προβλήματα και κοίλα προβλήματα δικτύων.

2.4 Ιδιότητες κοίλων συναρτήσεων και κοίλων προβλημάτων

Στην ενότητα αυτή θα εξεταστούν ορισμένες βασικές ιδιότητες των κοίλων συναρτήσεων που είναι ιδιαίτερα χρήσιμες στη συνέχεια [2]. Έτσι θεωρούμε την πραγματική και κοίλη συνάρτηση $f : A \to \mathbb{R}$ του προβλήματος (P) που ορίζεται στο κυρτό υποσύνολο A του $\mathbb{R}^n$.

Ιδιότητα 1

Αν $A = \mathbb{R}^n$ ή το A είναι ένα ανοιχτό κυρτό υποσύνολο του $\mathbb{R}^n$, τότε η f είναι συνεχής συνάρτηση στο A.

Κάποιες φορές στην κοίλη ανάλυση πρέπει να γίνουν διάφορες πράξεις πάνω σε σύνολα κοίλων συναρτήσεων που ορίζονται πάνω σε ένα κοινό σύνολο. Στην κοίλη ελαχιστοποίηση είναι κάποιες φορές αναγκαίο να βρεθεί το ελάχιστο ενός συνόλου κοίλων συναρτήσεων. Σε αυτές τις περιπτώσεις η ακόλουθη ιδιότητα είναι ιδιαίτερα σημαντική.

Ιδιότητα 2

Έστω h_i, $i = 1, 2, ..., q$ ένα σύνολο πραγματικών, κοίλων συναρτήσεων, κάθε μια από τις οποίες ορίζεται στο κυρτό υποσύνολο A του $\mathbb{R}^n$. Τότε η συνάρτηση $u : A \to \mathbb{R}$ ορίζεται για κάθε $x \in A$ ως

$$u(x) = min\{h_i(x)|\, i = 1, 2, ..., q\}$$

και είναι μια πραγματική κοίλη συνάρτηση στο A.

Στο πρόβλημα (P) έχουμε θεωρήσει ότι το σύνολο $D \subseteq \mathbb{R}^n$ είναι ένα μη κενό, κλειστό κυρτό σύνολο και ότι το κυρτό σύνολο $A \subseteq \mathbb{R}^n$ περιέχει το D. Υπενθυμίζουμε επίσης ότι για την κοίλη συνάρτηση f ισχύει ότι $f : A \to \mathbb{R}$. Εν συντομία θα αναφερόμαστε στα παραπάνω ως τις *βασικές υποθέσεις για το πρόβλημα* (P).

Αφού το σύνολο D είναι μη κενό τότε υπάρχουν τρεις δυνατές περιπτώσεις για τη λύση του προβλήματος (P):

1. Υπάρχει τουλάχιστον ένα ολικό ελάχιστο.

2. Παρουσιάζει πεπερασμένη ελάχιστη τιμή υ̃ αλλά δεν υπάρχει $x \in D$ τέτοιο ώστε $f(x) =$υ̃

3. Το πρόβλημα δεν είναι φραγμένο, δηλαδή υ̃$= -\infty$

Η δεύτερη περίπτωση, αν και δεν είναι συνηθισμένη, μπορεί να προκύψει για παράδειγμα όταν το A δεν είναι ένα ανοικτό σύνολο και η f είναι ασυνεχής σε ένα ή περισσότερα σημεία στο σύνορο του D, ή όταν το D δεν είναι φραγμένο. Πρακτικά στις περισσότερες περιπτώσεις αν υπάρχει ολικό ελάχιστο τότε υπάρχει και σημείο που να ανήκει στο D που να το επιτυγχάνει.

Σε ορισμένες περιπτώσεις προβλημάτων εύρεσης ολικού ελαχίστου το ολικό ελάχιστο, όταν υπάρχει, μπορεί να περιοριστεί στο σύνορο του επιτρεπτού συνόλου. Αυτό το πλεονέκτημα έχει χρησιμοποιηθεί από διάφορους αλγόριθμους εύρεσης ολικού ελαχίστου. Ειδικά τα προβλήματα ελαχιστοποίησης κοίλων συναρτήσεων έχουν μια ακόμα πιο σημαντική ιδιότητα:

Ιδιότητα 3

Έστω ότι οι βασικές υποθέσεις για το πρόβλημα (P) ισχύουν και επιπλέον το σύνολο D περιέχει τουλάχιστον ένα ακραίο σημείο. Εάν το πρόβλημα (P) έχει ολικό ελάχιστο τότε το σημείο αυτό είναι ένα ακραίο σημείο του D.

Γενικά αυτή η ιδιότητα θεωρείται η πιο εντυπωσιακή ιδιότητα των προβλημάτων κοίλης ελαχιστοποίησης. Συνεπώς η έρευνα για την εύρεση του ολικού ελαχίστου για το πρόβλημα (P) μπορεί να περιοριστεί στο σύνολο των ακραίων σημείων του D, με τον περιορισμό βέβαια ότι αυτό το σύνολο δεν είναι κενό. Επίσης αξίζει να σημειωθεί ότι η ιδιότητα 3 είναι αρκετά γενική και ισχύει ακόμα και όταν το D δεν είναι φραγμένο ή όταν η συνάρτηση f είναι ασυνεχής στο σύνορο του D.

Η ιδιότητα 3 είναι ιδιαίτερα χρήσιμη στην περίπτωση που το σύνολο D είναι κυρτό πολύεδρο. Αυτό συμβαίνει επειδή σύνολα αυτής της μορφής έχουν ένα πεπερασμένο αριθμό ακραίων σημείων, ακόμα και όταν δεν είναι φραγμένα.

Αλλά ακόμα και όταν το D δεν είναι πολύεδρο πολλοί αλγόριθμοι για τη λύση του προβλήματος (P) βασίζονται σε αυτή την ιδιότητα. Αυτοί οι αλγόριθμοι μπορεί να χρησιμοποιήσουν την ιδιότητα 3 με διάφορους τρόπους. Για παράδειγμα πολλοί αλγόριθμοι βασίζονται σε αυτήν για να επιταχύνουν την ελαχιστοποίηση της f πάνω σε διάφορα σύνολα που κατασκευάζουν και είναι πολύεδρα και περιέχουν το D ή περιέχονται μέσα στο D.

Την ιδιότητα αυτή θα εκμεταλλευτούμε τόσο κατά την εφαρμογή του αλγορίθμου των $Horst$ και Tuy για την εύρεση του ολικού ελαχίστου του προβλήματος 1.1, όσο και για τη δημιουργία του προγράμματος για την επαλήθευση του αποτελέσματος.

Φυσικά για να χρησιμοποιήσει κανείς την ιδιότητα 3 θα πρέπει να επιβεβαιώσει πρώτα ότι το πρόβλημα (P) έχει τουλάχιστον μια ολικά βέλτιστη λύση. Η επαλήθευση προκύπτει με εφαρμογή του θεωρήματος $Weierstrass$. Επειδή ένα μη κενό, συμπαγές σύνολο πρέπει να έχει ένα ακραίο σημείο, από την ιδιότητα 1 αυτής της παραγράφου παίρνουμε την ακόλουθη ιδιότητα, που είναι μια εξειδικευμένη μορφή της ιδιότητας 3.

Ιδιότητα 4

Έστω ότι οι βασικές υποθέσεις για το πρόβλημα (P) ισχύουν και επιπλέον το σύνολο A ισούται είτε με το $\mathbb{R}^n$ είτε με ένα ανοικτό κυρτό σύνολο στο $\mathbb{R}^n$ και ότι το D είναι φραγμένο. Τότε το πρόβλημα (P) έχει οπωσδήποτε ένα ολικό ελάχιστο που είναι ακραίο σημείο του D.

Από την ιδιότητα 4 συνεπάγεται ότι το πρόβλημα (P) έχει ολικό ελάχιστο που είναι ακραίο σημείο εφ΄ όσον η f είναι κοίλη συνάρτηση σε κάποιο ανοιχτό κυρτό σύνολο που περιέχει το D και το D είναι ένα μη κενό, συμπαγές κυρτό σύνολο. Αυτό είναι ίσως το πιο γνωστό θεωρητικό συμπέρασμα στην ελαχιστοποίηση κοίλων συναρτήσεων και είναι ένας από τους λόγους που κάνουν αυτή την κατηγορία προβλημάτων τόσο ελκυστική για τους ερευνητές. Η ιδιότητα 4 χρησιμοποιείται με παρόμοιο τρόπο με την ιδιότητα 3, δηλαδή για να δικαιολογήσει και να επιταχύνει πολλούς αλγορίθμους και διαδικασίες που ψάχνουν λύση για το πρόβλημα (P) σε ακραίο σημείο.

Ωστόσο σε ένα μεγάλο αριθμό εφαρμογών του προβλήματος (P) είτε η συνάρτηση f δεν είναι συνεχής σε κάθε ανοιχτό σύνολο που περιέχει το D, είτε το D είναι μη φραγμένο, είτε συμβαίνουν και τα δυο. Σε αυτές τις περιπτώσεις η ιδιότητα 4 δεν μπορεί να εφαρμοστεί. Ωστόσο η ιδιότητα 3, ως πιο γενική στη διατύπωση της, μπορεί ακόμα να χρησιμοποιηθεί και να δείξει έτσι ότι ακόμα και αυτά τα προβλήματα έχουν ολικό ελάχιστο, που είναι ακραίο σημείο. Για να εφαρμοστεί η ιδιότητα 3 σε αυτές τις περιπτώσεις μπορούν να χρησιμοποιηθούν κάποιες επιπλέον ιδιότητες του προβλήματος (P), που εγγυούνται την ύπαρξη ολικού ελαχίστου. Μια από τις πιο αξιομνημόνευτες ιδιότητες αυτού του τύπου παρουσιάζεται στη συνέχεια.

Ιδιότητα 5

Έστω ότι το D είναι ένα μη κενό πολύεδρο, πιθανώς μη φραγμένο. Τότε αν η συνάρτηση f είναι φραγμένη από κάτω πάνω στο σύνολο D τότε το πρόβλημα (P) έχει τουλάχιστον ένα ολικό ελάχιστο.

Είναι γνωστό ότι μια κοίλη συνάρτηση που είναι φραγμένη από κάτω πάνω σε ένα αυθαίρετο κυρτό σύνολο δε χρειάζεται να αποκτήσει την ελάχιστη τιμή της πάνω στο σύνολο, ακόμα και αν το σύνολο είναι επίσης συμπαγές. Αλλά βάση της ιδιότητας 5, μια κοίλη συνάρτηση φραγμένη από κάτω πάνω σε ένα πολύεδρο θα απόκτα πάντα την ελάχιστη τιμή της πάνω στο πολύεδρο, σε κάποιο σημείο μέσα στο πολύεδρο. Αυτό ισχύει ακόμα και όταν η συνάρτηση είναι ασυνεχής στο σύνορο του πολύεδρου, ή όταν το πολύεδρο είναι μη φραγμένο, ή όταν ισχύουν και τα δυο. Το αποτέλεσμα είναι ότι όταν το σύνολο D είναι ένα μη κενό πολύεδρο τότε αν το πρόβλημα (P) είναι μη φραγμένο θα πρέπει να έχει μια ολικά βέλτιστη λύση.

Επίσης το πρόβλημα (P) έχει κάποιες ειδικές ιδιότητες όταν το πρόβλημα δεν είναι φραγμένο, δηλαδή όταν $\tilde{\upsilon}= -\infty$. Για παράδειγμα μπορεί να αποδειχθεί ότι όταν το σύνολο D περιέχει τουλάχιστον ένα ακραίο σημείο, τότε αν το πρόβλημα (P) είναι μη φραγμένο η συνάρτηση f πρέπει να είναι μη φραγμένη από κάτω είτε πάνω σε κάποια ημιευθεία στο D είτε πάνω σε κάποιο σύνολο ακραίων σημείων του D (Για περισσότερες πληροφορίες δείτε Θεώρημα 32.3, $Rockafellar$ (1970)). Το τελευταίο ενδεχόμενο σε αυτή την ιδιότητα μπορεί να εξαλειφθεί κάτω από ορισμένες ισχυρότερες προϋποθέσεις.

Για παράδειγμα αφού ένα πολύεδρο έχει το πολύ ένα πεπερασμένο αριθμό ακραίων σημείων η ακόλουθη εξειδίκευση της ιδιότητας 5 ισχύει:

Ιδιότητα 6

Έστω ότι το D είναι ένα πολύεδρο που έχει τουλάχιστον ένα ακραίο σημείο. Τότε αν το πρόβλημα (P) είναι μη φραγμένο η συνάρτηση f πρέπει να είναι μη φραγμένη από κάτω πάνω σε τουλάχιστον μια ημιευθεία που περιέχεται στο D.

Η ιδιότητα 6 εφαρμόζεται ακόμα και όταν η συνάρτηση f είναι ασυνεχής σε σημεία πάνω στο σύνορο του D. Όταν είτε $A = \mathbb{R}^n$, είτε το A είναι ένα ανοικτό κυρτό σύνολο, τότε από την ιδιότητα 1 αυτής της παραγράφου η f είναι συνεχής στο $A \supseteq D$. Σε αυτή την περίπτωση η ιδιότητα 6 μπορεί να γίνει πιο ισχυρή. Υπενθυμίζουμε ότι μια *ακραία κατεύθυνση* ενός κυρτού συνόλου είναι μια έκφραση του συνόλου που είναι ημιευθεία.

Ιδιότητα 7

Έστω ότι το D είναι ένα πολύεδρο που έχει τουλάχιστον ένα ακραίο σημείο, και ότι είτε $A = \mathbb{R}^n$, είτε το A είναι ένα ανοικτό κυρτό σύνολο που περιέχει το D. Τότε αν το πρόβλημα (P) είναι μη φραγμένο η συνάρτηση f πρέπει να είναι μη φραγμένη από κάτω πάνω σε τουλάχιστον μια ακραία κατεύθυνση του D.

Οι ιδιότητες 6 και 7, ειδικά η τελευταία, είναι ιδιαίτερα χρήσιμες κατά την επίλυση του προβλήματος (P) όταν το D είναι ένα μη φραγμένο πολύεδρο. Κάτω από τις προϋποθέσεις της ιδιότητας 7 όταν το D είναι ένα πολύεδρο με τουλάχιστον ένα ακραίο σημείο και εάν το πρόβλημα (P) είναι μη φραγμένο τότε η συνάρτηση f πρέπει να είναι μη φραγμένη πάνω σε τουλάχιστον μια ακραία κατεύθυνση του D.

Συνοψίζοντας παραθέτουμε τις πιο σημαντικές ιδιότητες του προβλήματος (P) καθώς επίσης και ιδιότητες που προκύπτουν εύκολα αν συνδυάσουμε τις ανάλογες ιδιότητες και τα συμπεράσματα που αναφέρθηκαν σε αυτή την παράγραφο. Για κάθε ιδιότητα υποθέτουμε ότι οι βασικές υποθέσεις για το πρόβλημα (P) ισχύουν.

1. Το πρόβλημα (P) είτε έχει τουλάχιστον μια ολικά βέλτιστη λύση, είτε παρουσιάζει πεπερασμένη ελάχιστη τιμή $\tilde{\upsilon}$ αλλά δεν υπάρχει $x \in D$ τέτοιο ώστε $f(x) = \tilde{\upsilon}$, είτε είναι μη φραγμένο.

2. Όταν το σύνολο D περιέχει τουλάχιστον ένα ακραίο σημείο, εάν το πρόβλημα

(P) έχει τουλάχιστον ένα ολικό ελάχιστο, τότε πρέπει να έχει ένα ολικό ελάχιστο που να είναι ακραίο σημείο του D.

3. Όταν το σύνολο D είναι συμπαγές, εάν η συνάρτηση f ορίζεται σε κάποιο ανοικτό κυρτό σύνολο που περιέχει το D, τότε το πρόβλημα (P) πρέπει να έχει τουλάχιστον ένα ολικό ελάχιστο που να είναι ακραίο σημείο του D.

4. Όταν το D περιέχει τουλάχιστον ένα ακραίο σημείο, εάν το πρόβλημα (P) είναι μη φραγμένο, τότε η συνάρτηση f πρέπει να είναι μη φραγμένη από κάτω είτε πάνω σε κάποια ημιευθεία στο D, είτε πάνω σε κάποιο σύνολο ακραίων σημείων του D.

5. Όταν το σύνολο D είναι ένα πολύεδρο που περιέχει τουλάχιστον ένα ακραίο σημείο, τότε είτε το πρόβλημα (P) πρέπει να έχει ένα ολικό ελάχιστο που να είναι ακραίο σημείο του D, είτε το πρόβλημα (P) πρέπει να είναι μη φραγμένο και η συνάρτηση f πρέπει να είναι μη φραγμένη από κάτω πάνω σε κάποια ημιευθεία στο D.

6. Όταν το σύνολο D είναι πολύεδρο που περιέχει τουλάχιστον ένα ακραίο σημείο και η συνάρτηση f ορίζεται σε κάποιο ανοικτό κυρτό σύνολο που περιέχει το D, τότε, εάν το πρόβλημα (P) είναι μη φραγμένο, η f πρέπει να είναι μη φραγμένη από κάτω πάνω σε κάποια ακραία κατεύθυνση του D.

Οι θεμελιώδεις ιδιότητες του προβλήματος (P) που παρουσιάστηκαν σε αυτή την ενότητα θεωρούνται απαραίτητες για την ανάλυση, τη διατύπωση ή τη δικαιολόγηση κάθε διαδικασίας που προσπαθεί να λύσει ένα πρόβλημα ελαχιστοποίησης κοίλης συνάρτησης. Φυσικά υπάρχουν και άλλες, περισσότερο εξειδικευμένες ιδιότητες. Ωστόσο οι ιδιότητες που παραθέτονται εδώ χρησιμεύουν ως ένα βασικό σύνολο αναφοράς, που μπορεί να βοηθήσει τον αναγνώστη να καταλάβει τη διαδικασία που χρησιμοποιείται για την επίλυση του προβλήματος τετραγωνικού προγραμματισμού 1.1.

2.5 Τρόποι επίλυσης κοίλων προβλημάτων

Οι ειδικές μαθηματικές ιδιότητες και η ποικιλία των άμεσων και έμμεσων εφαρμογών των προβλημάτων κοίλης ελαχιστοποίησης υποκινήσαν την ανάπτυξη πολλών και διαφορετικών προσεγγίσεων για την επίλυση τους. Παρ' όλο που πολλοί αλγόριθμοι ελαχιστοποίησης κοίλων συναρτήσεων χρησιμοποιούν ένα συνδυασμό μεθόδων, οι περισσότεροι από αυτούς ανήκουν σε μια από τις τρεις βασικές κατηγορίες. Αυτές είναι η απαριθμητική μέθοδος (enumeration approach), η διαδοχική προσέγγιση (successive approximation) και ο διαδοχικός διαμερισμός (successive partitioning ή branch and bound).

Η απαριθμητική μέθοδος για την επίλυση του προβλήματος (P) εφαρμόζεται όταν το σύνολο D είναι ένα μη κενό πολύεδρο. Για παράδειγμα από την ιδιότητα 4 της προηγούμενης παραγράφου είδαμε ότι εάν το D είναι ένα μη κενό, φραγμένο πολύεδρο και η f ορίζεται σε κάποιο ανοιχτό κυρτό σύνολο που περιέχει το D, τότε ένα από τα (πεπερασμένα σε αριθμό) ακραία σημεία του D πρέπει να είναι ολικό ελάχιστο για το πρόβλημα (P). Αυτή η παρατήρηση είναι εξαιρετικά χρήσιμη γιατί για πολυεδρικές περιοχές D, το πρόβλημα (P) μπορεί να επιλυθεί με κατάλληλη απαρίθμηση των ακραίων σημείων του D εώς ότου βρεθεί το ολικό ελάχιστο. Αυτό ακριβώς είναι το επίκεντρο των απαριθμητικών μεθόδων για προβλήματα κοίλης ελαχιστοποίησης πάνω σε πολύεδρα. Σε προβλήματα αυτού του τύπου οι μέθοδοι αυτές ψάχνουν το ακριβές ακραίο σημείο που βρίσκει το ολικό ελάχιστο μέσα σε πεπερασμένο αριθμό επαναλήψεων.

Οι άλλες δυο προσεγγίσεις για τη λύση του προβλήματος (P), η διαδοχική προσέγγιση και ο διαδοχικός διαμερισμός, εφαρμόζονται πιο συχνά από την απαριθμητική μέθοδο. Αυτό συμβαίνει γιατί και οι δυο μπορούν να χρησιμοποιηθούν όταν το D είναι πολύεδρο αλλά και όταν το D είναι γενικά ένα κλειστό κυρτό σύνολο.

Η successive approximation προσέγγιση για το πρόβλημα (P) κατασκευάζει μια ακολουθία βελτιωμένων προσεγγίσεων του αρχικού προβλήματος. Αυτά τα προσεγγιστικά προβλήματα είναι, εν γένει, πιο απλό να λυθούν από το αρχικό πρόβλημα. Συνήθως λύνονται διαδοχικά μέχρι να βρεθεί η ακριβής ολικά βέλτιστη λύση του αρχικού προβλήματος, ή μια προσέγγιση αυτής.

Τέλος, η μέθοδος του διαδοχικού διαμερισμού είναι πιθανόν η πιο δημοφιλής προσέγγιση για τη λύση του προβλήματος (P). Η μέθοδος αυτή ξεκινάει δημιουργώντας ένα διαμερισμό είτε του D, είτε ενός συνόλου M_o που περιέχει το D. Αυτή η διχοτόμηση υποδιαιρεί το D ή το M_o σε μικρότερα κομμάτια. Στη συνέχεια από κάθε κομμάτι εξάγονται με έμμεσους τρόπους πληροφορίες σχετικά με την πιθανότητα να περιέχεται σε αυτό μια ολικά βέλτιστη λύση. Ορισμένα κομμάτια με βάση αυτές τις πληροφορίες μπορούν να αποκλειστούν από περαιτέρω έρευνα. Μετά το τέλος της εξέτασης όλων των κομματιών και της αφαίρεσης ορισμένων από αυτά, κατασκευάζεται μια βελτιωμένη υποδιαίρεση του D ή του M_o, και η διαδικασία επαναλαμβάνεται.

Κεφάλαιο 3

Τομές Tuy και αλγόριθμος Horst και Tuy

3.1 Εισαγωγή

Στο κεφάλαιο αυτό θα παρουσιάσουμε τον αλγόριθμο Horst και Tuy, ο οποίος ανήκει στις απαριθμητικές μεθόδους λύσης του προβλήματος (P). Αρχικά στην παράγραφο 3.2 εισάγεται η έννοια των Tuy cuts. Εάν εν γένει $x \in D$ τότε οι τομές Tuy είναι μια διαδικασία που προσπαθεί να απαλείψει υποπεριοχές D_1 του D οι οποίες δε χρειάζεται να εξεταστούν περαιτέρω. Στην παράγραφο 3.3 αναλύονται διεξοδικά τα βήματα που αποτελούν τον αλγόριθμο. Ο αλγόριθμος Horst και Tuy ανήκει στις cone covering προσεγγίσεις και, όπως λέει και το όνομα του, καλύπτει το σύνολο D με κώνους και στη συνέχεια χρησιμοποιώντας τομές Tuy προσπαθεί να βρει ένα ολικό ελάχιστο για το πρόβλημα (P).

3.2 Τομές Tuy

Μια από τις πιο συχνές ' εργασίες ' που περιλαμβάνονται στους αλγορίθμους ελαχιστοποίησης κοίλων συναρτήσεων με σκοπό την επίλυση του προβλήματος (P) που εξετάσαμε στο πρώτο κεφάλαιο είναι η προσπάθεια να απαλειφθούν τμήματα από το πραγματοποιήσιμο σύνολο D που δε χρειάζεται να διερευνηθούν περαιτέρω [2], [3]. Για παράδειγμα, εάν έχουμε ένα δεδομένο σημείο $\bar{x} \in D$ ή αν μπορέσουμε να βρούμε ένα τέτοιο σημείο από ένα δοσμένο αλγόριθμο, τότε κάθε τμήμα D_1 του D τέτοιο ώστε $f(x) \geq f(\bar{x})$ για όλα τα $x \in D_1$ μπορεί να απαλειφθεί χωρίς να εξεταστεί. Για να επιτύχουν αυτή την απαλοιφή, κάποιοι αλγόριθμοι χρησιμοποιούν διαδικασίες οι οποίες περιλαμβάνουν την κατασκευή ειδικών τομών. Η γενική ιδέα πίσω από την κατασκευή αυτών των τομών δόθηκε αρχικά από τον Tuy το 1964. Οι κατασκευές αυτών των τομών στηρίζονται σε μια πολύ σημαντική ιδιότητα των κοίλων συναρτήσεων, που θα παρουσιαστεί στη συνέχεια. Αρχικά θα δώσουμε τον εξής πολύ σημαντικό ορισμό:

Ορισμός 1

Ένα σύνολο K λέγεται κώνος εάν για $x \in K$ τότε και το $\alpha x \in K$ για κάθε $\alpha > 0$.

Ένα σύνολο K λέγεται κυρτός κώνος εάν επιπλέον για κάθε ζευγάρι σημείων $x \in K$, $y \in K$ και $\forall \lambda \in [0,1]$ ισχύει $\lambda x + (1-\lambda)y \in K$.

Για απλότητα υποθέτουμε ότι $A = \mathbb{R}^n$ και ότι η συνάρτηση f έχει φραγμένα ισοϋψή σύνολα, δηλαδή για κάθε $\alpha \in \mathbb{R}$, το σύνολο $\{x \in \mathbb{R}^n | \, f(x) \geq \alpha\}$ είναι φραγμένο. Με μικρές αλλαγές η κατασκευή τομών που περιγράφεται στη συνέχεια ισχύει και όταν το σύνολο A είναι ένα ανοικτό κυρτό υποσύνολο του $\mathbb{R}^n$ και ένα ή περισσότερα από τα ισοϋψή σύνολα της f στο A είναι μη φραγμένα.

Έστω M ένα συμπαγές, φραγμένο σύνολο στο $\mathbb{R}^n$. Υποθέτουμε ότι υπάρχουν $x^o \in M$ και $\bar{x} \in M$ που ικανοποιούν τη σχέση $f(\bar{x}) \leq f(x^o)$. Για παράδειγμα το x^o μπορεί να είναι ένα εύκολα υπολογίσιμο σημείο του M και το $\bar{x}$ μπορεί να είναι η καλύτερη δυνατή λύση που βρίσκεται μετά από κάποιες επαναλήψεις ενός αλγορίθμου ελαχιστοποίησης της f πάνω στο M. Θα δείξουμε πως μπορούν να κατασκευαστούν υπερεπίπεδα που μπορούν να χρησιμοποιηθούν για να ορίσουν υποσύνολα M_1 του M που μπορούν να απαλειφθούν από περαιτέρω έλεγχο από τους αλγόριθμους που ψάχνουν για ολικά ελάχιστα της f πάνω στο M.

Υποθέτουμε ότι το K είναι ένας πολυεδρικός κώνος με κορυφή x^o και γραμμικά ανεξάρτητους γεννήτορες $(v^i - x^o)$, $i \in I$, όπου $\{v^i | \, i \in I\}$ είναι ένα σύνολο σημείων πάνω στο σύνορο του M. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι $I = \{1, 2, ..., n\}$. Ορίζουμε την ποσότητα γ ως:

$$\gamma = min\{f(\bar{x}), f(v^1), f(v^2), ..., f(v^n)\}.$$

Για κάθε $i \in I$ θεωρούμε ότι το z^i ορίζει κάποιο σημείο, διάφορο του x^o πάνω στην i πλευρά $\{x^o + \theta(v^i - x^o) | \; \theta \geq 0\}$ του Κ τέτοιο ώστε $f(z^i \geq \gamma)$. Παρατηρούμε ότι αφού για κάθε $i \in I$, $f(v^i) \geq \gamma$ τα σημεία z^i, $i \in I$ υπάρχουν. Επιπλέον τα σημεία αυτά είναι αφινικά ανεξάρτητα, έτσι ώστε να υπάρχει ένα μοναδικό υπερεπίπεδο, το οποίο περνάει ανάμεσα τους. Για την καλύτερη κατανόηση του όρου ' αφινικά ανεξάρτητα ' παραθέτουμε τους εξής ορισμούς [4]:

Ορισμός 2

Το $A \subseteq \mathbb{R}^n$ είναι ένα **αφινικό σύνολο** αν και μόνο αν $\lambda A + \mu A \subseteq A$ για κάθε λ, $\mu \in \mathbb{R}$ με $\lambda + \mu = 1$.

Ορισμός 3

Το σύνολο $A \subseteq \mathbb{R}^n$ είναι **αφινικά εξαρτημένο** αν και μόνο αν υπάρχουν διαφορετικά σημεία $\alpha_1 ... \alpha_m \in A$ και $\lambda_1 ... \lambda_m \in \mathbb{R}$ όχι όλα ίσα με το μηδέν τέτοια ώστε $\sum_{k=1}^{m} \lambda_k \alpha_k = 0$ και $\sum_{k=1}^{m} \lambda_k = 0$.

Πρόταση 1

Το υπερεπιπέδο H_z δίνεται από την εξίσωση:

$$< e^T Z^{-1}, x - x^o >= 1, \tag{3.1}$$

όπου $e^T \in \mathbb{R}^n$ είναι ένα διάνυσμα γραμμής με στοιχεία μονάδες και Z είναι ο $n \times n$ πίνακας που έχει ως στήλες τα $(z^i - x^o, i \in I)$.

Απόδειξη

Γνωρίζουμε ότι υπερεπίπεδο H στον $\mathbb{R}^n$ λέγεται το σύνολο

$$H = \{x \in \mathbb{R}^n : \; c^T x = z\}$$

, όπου $c, z \in \mathbb{R}^n$ και $c \neq 0$.

Συνεπώς θεωρούμε ότι το υπερεπιπέδο H_z δίνεται από τη γενική εξίσωση

$$H_z = \{x \in \mathbb{R}^n : \; c^T (x - x^o) + q = 0\}$$

Τα σημεία z_i, $i = 1, 2, ..., n$ είναι σημεία του υπερεπιπέδου H_z οπότε πρέπει να ισχύει:

$c^T (z_i - x^o) + q = 0, \; i = 1, 2, ..., n \Leftrightarrow$

$\Leftrightarrow (z_i - x^o)^T c + q = 0, \; i = 1, 2, ..., n \Leftrightarrow$

$\Leftrightarrow Z^T c + qe = 0 \Leftrightarrow$
$\Leftrightarrow c = -qZ^{-T}e$
όπου $Z = [z^1 - x^o, z^2 - x^o, ..., z^n - x^o]$ και $e^T = [1 \quad 1 \ ... \ 1]$.

Οπότε το υπερεπιπέδου H_z έχει εξίσωση

$$-qe^T Z^{-1}(x - x^o) + q = 0 \Leftrightarrow e^T Z^{-1}(x - x^0) = 1$$

$Q.E.D.$

Εύκολα μπορεί να παρατηρήσει κανείς ότι παίρνουμε ένα διαφορετικό υπερεπίπεδο H_z για κάθε διαφορετικό σύνολο σημείων z^i, $i \in I$.

Ιδιότητα 1

Θεωρούμε ότι $A = \mathbb{R}^n$ και ορίζουμε τα M, γ και K σύμφωνα με όσα αναφέραμε παραπάνω. Έστω επίσης H_z ένα οποιοδήποτε υπερεπίπεδο που ικανοποιεί την εξίσωση 3.1, όπου τα x^o και Z ορίζονται με βάση όσα έχουμε προαναφέρει. Ορίζουμε το σύνολο M_1 ως:

$$M_1 = (M \cap K) \cap \{x \in \mathbb{R}^n | < e^T Z^{-1}, x - x^o > \leq 1\}.$$

Τότε $f(x) \geq \gamma$ για όλα τα $x \in M_1$.

Η γραμμική ανισότητα $< e^T Z^{-1}, x - x^o > \geq 1$ καλείται ***γ-valid cut*** για τη συνάρτηση f και το $(M \cap K)$, αφού κάθε σημείο x στο $(M \cap K)$ που ικανοποιεί την ανισότητα $f(x) < \gamma$ θα πρέπει να ικανοποιεί και αυτή.

Στην πράξη προκύπτουν ορισμένες παραλλαγές στον ορισμό του πολυεδρικού κώνου K που χρησιμοποιείται για την κατασκευή των γ-valid cuts. Μια από τις πιο γνωστές παραλλαγές συναντάται όταν το M είναι πολυεδρικό και το x^o είναι ένα ακραίο σημείο του M.

Από την ιδιότητα 1 παρατηρούμε ότι όσο μακρύτερα τα σημεία z^i, $i \in I$ είναι από το x^o, τόσο μεγαλύτερο γίνεται το σύνολο M_1. Καθώς το M_1 γίνεται μεγαλύτερο τότε, λόγω της ιδιότητας 1, ολοένα και μεγαλύτερα σύνολα σημείων x βρίσκονται που ικανοποιούν την ανισότητα $f(x) \geq \gamma$. Επειδή σκοπός είναι να ελαχιστοποιηθεί η συνάρτηση

f πάνω στο σύνολο M είναι επιθυμητό να είναι αυτά τα σύνολα όσο μεγαλύτερα γίνεται, αφού σύμφωνα με τη θεωρία που αναπτύχθηκε παραπάνω τα σύνολα αυτά μπορούν να απαλειφθούν και να μην εξεταστούν περαιτέρω. Είναι συνεπώς ωφέλιμο οι αλγόριθμοι που στοχεύουν στην ελαχιστοποίηση κοίλων συναρτήσεων να επιλέγουν σημεία z^i, $i \in I$ κατά την κατασκευή του H_z, τέτοια ώστε να είναι όσο το δυνατόν πιο μακριά από το x^o κατά μήκος των πλευρών του κώνου Κ. Για κάθε $i \in I$ αυτό μπορεί να επιτευχθεί αν θέσουμε $z^i = x^o + \theta_i(\upsilon^i - x^o)$, όπου $\theta_i = max\{\theta \mid f[x^o + \theta(\upsilon^i - x^o)] \geq \gamma\}$. Είναι εύκολο να παρατηρήσει κανείς ότι τα σημεία z^i που προκύπτουν με αυτό τον τρόπο μπορεί να είναι έξω από το σύνολο M. Τότε τα σημεία αυτά λέγονται ***γ-extensions*** των υ^i, $i \in I$. Η γ-valid cut που αποκτάται με τον τρόπο αυτό είναι η βαθύτερη (μεγαλύτερη) δυνατή και καλείται ***concavity cut*** ή και ***Tuy cut*** προς τιμή του Hoang Tuy που την ανακάλυψε.

3.3 Αλγόριθμος Horst και Tuy

Στην παράγραφο αυτή θα παρουσιάσουμε τον περίφημο αλγόριθμο των Horst και Tuy [2], [3]. Ο αλγόριθμος αυτός επιχειρεί, καλύπτοντας το χώρο με κώνους και χρησιμοποιώντας τις τομές *Tuy* που περιγράφηκαν στην προηγούμενη παράγραφο, να ψάξει με μια επαναληπτική διαδικασία το χώρο D για ακραία σημεία, για τα οποία η συνάρτηση f του προβλήματος (P) δίνει τιμές που μειώνονται αυστηρά από επανάληψη σε επανάληψη. Διάφορες παραλλαγές του αρχικού αλγορίθμου του *Tuy* προτάθηκαν από τους Bali (1973), Zwart (1974), Jacobsen (1981) και άλλους. Οι περισσότεροι από αυτούς τους αλγόριθμους ψάχνουν ολικά ε-ελάχιστα, δηλαδή ελάχιστα στα οποία η συνάρτηση f παίρνει τιμές οι οποίες δεν ξεπερνούν την τιμή ῦ+ε, όπου ῦ το ολικό ελάχιστο της f και $\varepsilon \geq 0$ είναι μια σταθερά που καθορίζεται από πριν.

Χωρίς βλάβη της γενικότητας υποθέτουμε σε αυτή την παράγραφο ότι το σύνολο D δίνεται από την παρακάτω σχέση:

$$D = \{x \in \mathbb{R}^n | \, Cx \leq b, \; x \geq 0\}$$

. Επιπλέον θεωρούμε ότι το εσωτερικό του D δεν είναι κενό, δηλαδή ότι $(intD) \neq \emptyset$.

Στη συνέχεια θα περιγράψουμε τις δυο βασικές διαδικασίες, τις οποίες συναντάμε σε όλους τους αλγόριθμους που ανήκουν στις cone covering προσεγγίσεις για την επίλυση του προβλήματος ελαχιστοποίησης κοίλων συναρτήσεων. Εν γένει στην πρώτη διαδικασία, δοσμένου ενός κώνου K, προσπαθούμε να κατασκευάσουμε υπερεπίπεδα, ανάλογα των Tuy cuts που περιγράφηκαν στην προηγούμενη παράγραφο. Στη δεύτερη διαδικασία ένας πολυεδρικός κώνος $K \subseteq \mathbb{R}^n$ που έχει την κορυφή του σε ένα ακραίο σημείο του D και έχει n γραμμικούς γεννήτορες υποδιαιρείται σε n ή λιγότερους υποκώνους του ίδιου τύπου. Οι δυο αυτές διαδικασίες συνδυάζονται κατάλληλα ώστε είτε να βρεθεί ένα βελτιωμένο σημείο του D που περιέχεται στο κομμάτι $(D \cap K)$ του D που καλύπτεται από τον κώνο K και που δίνει μικρότερη τιμή στη συνάρτηση f είτε να δειχτεί ότι δεν υπάρχει τέτοιο σημείο.

Αναλυτικότερα, στο πρώτο βήμα των αλγορίθμων που ανήκουν στις cone covering προσεγγίσεις, θεωρούμε έναν πολυεδρικό κώνο K, του οποίου η κορυφή x^o είναι ακραίο σημείο του συνόλου D και ταυτόχρονα το $f(x^o)$ είναι τοπικό ελάχιστο της συνάρτησης f πάνω στο D. Υποθέτουμε ότι ο K έχει n γραμμικά ανεξάρτητους γεννήτορες $(y^i - x^o)$, $i = 1, 2, ..., n$ όπου για κάθε $i = 1, 2, ..., n$ το y^i είναι ένα στοιχείο του D για το οποίο ισχύει $f(y^i) \geq f(x^o)$. Συνεπώς, δοσμένου ενός τέτοιου κώνου K, η διαδικασία αναζήτησης που ακολουθείται είναι ως έχει:

Αρχικά για κάθε $i = 1, 2, ...n$ βρίσκουμε την α - επέκταση z^i του y^i. Για κάθε i αυτό το σημείο z^i δίνεται από τη σχέση $z^i = x^o + \theta_i(y^i - x^o)$, όπου

$$\theta_i = sup\{\theta > 0 \mid [x^o + \theta(y^i - x^o)] \in A,\ \ f[x^o + \theta(y^i - x^o)] \geq \alpha\}$$

και $\alpha = f(x^o) - \varepsilon$. Με βάση την παράγραφο 3.2 τα σημεία z^i, $i = 1, 2, ..., n$ θα υπάρχουν γιατί το σύνολο $\{x \in A \mid f(x^o) \geq \alpha\}$ έχουμε υποθέσει ότι είναι φραγμένο. Επιπλέον θα ισχύει ότι η γραμμική ανισότητα

$$< e^T Z^{-1}, x - x^o > \ \geq 1$$

, όπου Z είναι ο $(n \times n)$ πίνακας με στήλες τα $(z^i - x^o)$, $i = 1, 2, ...n$, είναι ένα Tuy cut. Κάθε $x \in (D \cap K)$ για το οποίο $f(x) < \alpha$ πρέπει να ικανοποιεί την παραπάνω ανισότητα αυστηρά.

Ωστόσο στη συνέχεια για να βρεθεί σημείο x στο $(D \cap K)$ που να ικανοποιεί τη σχέση $f(x) < \alpha$ θα πρέπει να λυθεί το πρόβλημα γραμμικού προγραμματισμού $LP(Z, D)$:

$$\max_{x \in D} < e^T Z^{-1}, x - x^o >$$

Έστω $\hat{w}$ η βέλτιστη λύση του παραπάνω προβλήματος. Παρατηρούμε ότι το $\hat{w}$ είναι ένα ακραίο σημείο του D. Είναι επίσης σημαντικό να συμπεριλάβουμε τον επιπλέον περιορισμό στη λύση του προβλήματος γραμμικού προγραμματισμού $LP(Z,D)$:

$$Z^{-1}(x - x^o) \geq 0 \tag{3.2}$$

Αυτός ο περιορισμός ουσιαστικά εκφράζει τη συνθήκη ότι $x \in K$.

Πράγματι:

$$\begin{aligned} x \in K &\Leftrightarrow x = x^o + \sum_{i=1}^{n} \mu_i (z^i - x^o),\ \mu_i \geq 0 \\ &\Leftrightarrow x - x^o = Z\mu,\ \mu \geq 0 \\ &\Leftrightarrow \mu = Z^{-1}(x - x^o),\ \mu \geq 0 \\ &\Leftrightarrow Z^{-1}(x - x^o) \geq 0. \end{aligned}$$

□

Στο σημείο αυτό υπάρχουν οι ακόλουθες εκδοχές:

- Αν $< e^T Z^{-1}, x - x^o > \ \leq 1$ τότε η διεργασία αναζήτησης καταλήγει στο ότι δεν υπάρχουν σημεία στο $(D \cap K)$ τέτοια ώστε η τιμή της συνάρτησης f στα σημεία αυτά να δίνει τιμές μικρότερες από α και συνεπώς σταματάει.
- Αν $< e^T Z^{-1}, x - x^o > \ > 1$ και $f(\hat{w}) < \alpha$ τότε η διεργασία έρευνας καταλήγει στο βελτιωμένο σημείο $\hat{w} \in (D \cap K)$.
- Τέλος αν $< e^T Z^{-1}, x - x^o > \ > 1$ αλλά $f(\hat{w}) \geq \alpha$ τότε η διεργασία αναζήτησης δεν μπορεί να βγάλει κάποιο συμπέρασμα για την ύπαρξη ενός σημείου w στο σύνολο $(D \cap K)$ που να ικανοποιεί τη σχέση $f(w) < \alpha$ και πρέπει συνεπώς να συνεχίσει.

Στην τελευταία περίπτωση για να συνεχίσουμε να ψάχνουμε τον κώνο K η διεργασία αναζήτησης χρησιμοποιεί τη δεύτερη διαδικασία, που ονομάζεται cone partitioning process, και ουσιαστικά υποδιαιρεί τον κώνο K σε n ή λιγότερους υποκώνους. Ειδικότερα η διαδικασία τεμαχισμού κώνων που χρησιμοποιείται στους cone covering αλγορίθμους καλείται radial subdivision και είναι ως έχει:

Ορίζουμε έναν πολυεδρικό κώνο $K \subseteq \mathbb{R}^n$ με κορυφή στο x^o και n γραμμικούς γεννήτορες $z^i - x^o$, $i = 1, 2, ..., n$, όπου x^o είναι ένα ακραίο σημείο του D, σύμφωνα με τη διαδικασία που αναφέρθηκε παραπάνω. Ορίζουμε S το *simplex* διάστασης $n-1$, που αποτελείται από το σύνολο όλων των κυρτών συνδυασμών των διανυσμάτων z^i, $i = 1, 2, ...n$. Δηλαδή

$$S = \{x \in \mathbb{R}^n \,:\, x = \sum_{i=1}^{n} \rho_i z_i,\ \sum_{i=1}^{n} \rho_i = 1,\ \rho_i \geq 0,\ i = 1, 2, ..., n\}.$$

Τότε δοσμένου οποιουδήποτε σημείου $\hat{u}$ στο K, που δεν βρίσκεται σε κάποια πλευρά του K, ο κώνος K υποδιαιρείται σε n ή λιγότερους υποκώνους ως εξής: Αρχικά βρίσκουμε μοναδικές τιμές $\hat{\lambda}_i$, $i = 1, 2, ..., n$ που να ικανοποιούν τη σχέση:

$$\hat{u} = x^o + \sum_{i=1}^{n} \hat{\lambda}_i (z^i - x^o),\ \textit{όπου}\ \hat{\lambda}_i \geq 0,\ i = 1, 2, ..., n. \tag{3.3}$$

Παρατηρούμε ότι αφού $\hat{u} \neq x^o$ τότε και $\{i \mid \hat{\lambda}_i > 0\} \neq \emptyset$. Στη συνέχεια βρίσκουμε μοναδικό σημείο u που ανήκει στο S, το οποίο κείται στη γραμμή που διέρχεται από τα $\hat{u}$ και x^o. Το σημείο αυτό πρέπει να ικανοποιεί τη σχέση:

$$u = x^o + \sum_{i=1}^{n} (\hat{\lambda}_i / L)(z^i - x^o),\ \ \textit{όπου}\ \ L = \sum_{i=1}^{n} \hat{\lambda}_i > 0 \tag{3.4}$$

Η εξίσωση 3.4 είναι πολύ εύκολο να αποδειχθεί. Πράγματι για το σημείο u γνωρίζουμε ότι:

- Ανήκει στη γραμμή που διέρχεται από τα $\hat{u}$ και x^o. Συνεπώς πρέπει $u = x^o + \kappa(\hat{u} - x^o)$.
- Ανήκει στο simplex S. Δηλαδή $u = \sum_{i=1}^{n} \rho_i z_i,\ \sum_{i=1}^{n} \rho_i = 1,\ \rho_i \geq 0,\ i = 1, 2, ..., n.$

Συνδυάζοντας τις παραπάνω σχέσεις παίρνουμε ότι:

$$x^o + \kappa(\hat{u} - x^o) = \sum_{i=1}^{n} \rho_i z_i,\ \sum_{i=1}^{n} \rho_i = 1,\ \rho_i \geq 0,\ i = 1, 2, ..., n \Leftrightarrow$$

$$\Leftrightarrow \kappa[x^o + \sum_{i=1}^{n} \hat{\lambda}_i (z^i - x^o) - x^o] = \sum_{i=1}^{n} \rho_i z^i - x^o,\ i = 1, 2, ..., n \Leftrightarrow$$

$$\Leftrightarrow \kappa \sum_{i=1}^{n} \hat{\lambda}_i (z^i - x^o) = \sum_{i=1}^{n} \rho_i z^i - \sum_{i=1}^{n} \rho_i x^o,\ i = 1, 2, ..., n \Leftrightarrow$$

$$\Leftrightarrow \sum_{i=1}^{n}(\kappa\hat{\lambda}_i - \rho_i)(z^i - x^o) = 0,\ i = 1,2,...,n \Leftrightarrow$$

$$\Leftrightarrow \kappa\hat{\lambda}_i - \rho_i = 0,\ i = 1,2,...,n \Leftrightarrow$$

$$\Leftrightarrow \rho_i = \kappa\hat{\lambda}_i,\ i = 1,2,...,n.$$

Όμως:

$$\sum_{i=1}^{n}\rho_i = 1 \Leftrightarrow \kappa = \frac{1}{\sum_{i=1}^{n}\hat{\lambda}} > 0$$

Επομένως:

$$\begin{aligned} u &= x^o + \frac{1}{\sum_{i=1}^{n}\hat{\lambda}}(x^o + \sum_{i-1}^{n}\hat{\lambda}_i(z^i - x^o) - x^o) \\ &= x^o + \sum_{i=1}^{n}\frac{\hat{\lambda}_i}{L}(z^i - x^o),\ \textit{όπου}\ L = \sum_{i=1}^{n}\hat{\lambda}_i \end{aligned}$$

□

Ορίζουμε $\lambda_i = \hat{\lambda}_i/L,\ i = 1,2,...,n$ και $I = \{i \mid \lambda_i > 0\}$. Ο κώνος K τότε υποδιαιρείται στους κώνους $\{K_i \mid i \in I\}$, όπου, για κάθε $i \in I$, K_i είναι ο πολυεδρικός κώνος με κορυφή x^o και γεννήτορες $z^1 - x^o, z^2 - x^o, ..., z^{i-1} - x^o, \tilde{u} - x^o, z^{i+1} - x^o, ..., z^n - x^o$, όπου $\tilde{u}$ είναι η α - επέκταση του u.

Είναι εύκολο να δει κανείς ότι για κάθε $i \in I$, οι n γεννήτορες του κώνου K_i είναι γραμμικά ανεξάρτητοι. Μπορεί επίσης να δειχτεί ότι το εσωτερικό των κώνων $K_i,\ i \in I$, είναι αντίστοιχα κλειστό, και ότι η ένωση των $K_i,\ i \in I$ μας δίνει τον αρχικό κώνο K. Συνεπώς η διαδικασία υποδιαίρεσης κώνου δημιουργεί ένα τεμαχισμό του K σε $q \leq n$ πολυεδρικούς κώνους, όπου q είναι ο αριθμός των στοιχείων του I, κάθε ένα από τα οποία, όπως και ο κώνος K, έχουν την κορυφή τους στο x^o και έχουν n γραμμικά ανεξάρτητους γεννήτορες.

Όπως αναφέρθηκε προηγουμένως, όταν στο τέλος της πρώτης διαδικασίας βρεθεί ότι $< e^T Z^{-1}, x - x^o > > 1$ αλλά $f(\hat{w}) \geq \alpha$ τότε η διεργασία αναζήτησης δεν μπορεί να βγάλει κάποιο συμπέρασμα για την ύπαρξη ενός σημείου w στο σύνολο $(D \cap K)$ που να ικανοποιεί τη σχέση $f(w) < \alpha$ και πρέπει συνεπώς να συνεχίσει. Στην τελευταία περίπτωση για να συνεχίσουμε να ψάχνουμε τον κώνο K η διεργασία αναζήτησης χρησιμοποιεί τη δεύτερη διαδικασία και υποδιαιρεί τον κώνο K σε n ή λιγότερους υποκώνους. Με αρχικό σημείο το $\hat{u} = \hat{w} \in K$, ο κώνος K υποδιαιρείται σε μέχρι n

πολυεδρικούς κώνους, κάθε ένας από τους οποίους, όπως και ο K, έχει το x^o ως κορυφή και n γραμμικά ανεξάρτητους γεννήτορες. Σε κάθε υποκώνο, ακριβώς $(n-1)$ από τους γεννήτορες είναι ίδιοι με αυτούς του κώνου K, αλλά ένας από τους γεννήτορες δίνεται από το διάνυσμα $(w - x^o)$, όπου το w υπολογίζεται από το $\hat{w}$ με τη βοήθεια της εξίσωσης 3.4 (δηλαδή το w είναι το αντίστοιχο του u στη διαδικασία υποδιαίρεσης κώνου που περιγράψαμε παραπάνω). Στη συνέχεια η διαδικασία αναζήτησης επαναλαμβάνεται σε κάθε υποκώνο. Αν δε βρεθεί κανένα βελτιωμένο σημείο x^o, τότε κάθε υποκώνος που πιθανόν μπορεί να περιέχει κάποιο βελτιωμένο σημείο υποδιαιρείται ξανά και η διαδικασία επαναλαμβάνεται έως ότου είτε να βρεθεί ένα σημείο $w \in (D \cap K)$ τέτοιο ώστε $f(w) > \alpha$ είτε να αποδειχθεί ότι ένα τέτοιο σημείο δεν υπάρχει.

Η αλγοριθμική διαδικασία που περιγράφηκε παραπάνω έχει χρησιμοποιηθεί στη δημιουργία πολλών αλγορίθμων με σκοπό την επίλυση του προβλήματος (P). Ένας από τους πιο γνωστούς αλγόριθμους είναι ο αλγόριθμος των $Horst$ και Tuy, που περιγράφεται στη συνέχεια:

Φάση I

Επιλέξτε σταθερά $\varepsilon \geq 0$ και κάποιο σημείο $z \in D$. Ξεκινώντας από το z, βρείτε ένα ακραίο σημείο $x = x^0$ του D που να είναι τοπικό ελάχιστο της $f(x)$ πάνω στο D.

Φάση II

Βήμα 1: Έστω $\alpha = f(x^o) - \varepsilon$. Κατασκευάστε έναν πολυεδρικό κώνο K^o με κορυφή το x^o και n γραμμικά ανεξάρτητους γεννήτορες $(z^i - x^o),\ i = 1, 2, ..., n$ έτσι ώστε $K \supseteq D$. Έστω Z^o ο $(n \times n)$ πίνακας του οποίου οι στήλες είναι οι γεννήτορες του K^o. Θέστε $CR = \{K^o\}$.

Βήμα 2: Για κάθε $K \in CR$ λύστε το πρόβλημα γραμμικού προγραμματισμού $LP(Z, D)$, όπου Z είναι ο $(n \times n)$ πίνακας του οποίου οι στήλες είναι οι γεννήτορες του K, λαμβάνοντας υπ' όψη και τη συνθήκη 3.2. Έστω $\hat{w}(K)$ η λύση.

Αν $f[\hat{w}(K)] < \alpha$ για κάποιο K, θέστε:

$$D = D \cap \{x \in \mathbb{R}^n \mid\ < e^T (Z^o)^{-1}, x - x^o > \ \geq 1\}$$

και, ξεκινώντας από το $\hat{w}(K)$, βρείτε ένα ακραίο σημείο x^o του D που να είναι τοπικό ελάχιστο της f πάνω στο D. Επιστροφή στο βήμα 1.

Αν $f[\hat{w}(K)] \geq \alpha$ για όλα τα K τότε συνεχίστε στο Βήμα 2.

<u>Βήμα 3:</u> Αν $< e^T Z^{-1}, \hat{w}(K) - x^o > \leq 1$ για κάθε $K \in CR$ τότε σταματήστε: Το x^o είναι ολικό ε - ελάχιστο. Αλλιώς, αφαιρέστε από το CR κάθε κώνο $K \in CR$ για τον οποίο ισχύει $< e^T Z^{-1}, \hat{w}(K) - x^o >> 1$ και στη συνέχεια θέτοντας $\hat{u} = \hat{w}(K)$ υποδιαιρέστε τον κώνο K σε δυο ή περισσότερους υποκώνους, σύμφωνα με τη διαδικασία που περιγράφηκε παραπάνω, και προσθέστε αυτούς τους υποκώνους στο CR. Επιστρέψτε στο Βήμα 2.

Παρατηρούμε ότι, στο βήμα 2, κάθε φορά που για κάποιο K συμβαίνει να ισχύει ότι $f[\hat{w}(K)] < \alpha$ βρίσκουμε ένα βελτιωμένο σημείο $\hat{w}(K) \in D$. Το σημείο αυτό με βάση την προηγούμενη παράγραφο και την εξίσωση 3.1 ικανοποιεί αυστηρά την ανισότητα $< e^T (Z^o)^{-1}, x - x^o > \geq 1\}$, η οποία προσδιορίζει μια τομή Tuy, εφ' όσον $K^o \supseteq D$. Αυτή η τομή Tuy χρησιμοποιείται στη συνέχεια για να μειώσει το μέγεθος του D και για να βρεθεί ένα βελτιωμένο ακραίο σημείο του D που να ικανοποιεί επίσης την τομή αυστηρά. Αυτό υπονοεί ότι ότι, αν και κατά τη διάρκεια του αλγορίθμου το D μειώνεται σταδιακά σε μέγεθος από τις τομές Tuy, τα βελτιωμένα ακραία σημεία δεν κείνται ποτέ πάνω σε αυτές τις τομές.

Ο αλγόριθμος των $Horst$ και Tuy αποτελεί επέκταση του αρχικού αλγορίθμου που είχε δημοσιευτεί από τον Tuy το 1964. Η διαφορά βρισκόταν στο ότι ο αρχικός αλγόριθμος του Tuy δεν περιείχε το πρόβλημα γραμμικού προγραμματισμού $LP(Z, D)$, ενώ επίσης λάμβανε ότι ε=0. Αυτό είχε ως αποτέλεσμα, στο βήμα 2, κάθε σημείο $\hat{w}(K)$ να είναι ένα ακραίο σημείο του D και συνεπώς δεν είχε νόημα η εύρεση ενός νέου ακραίου σημείου x^o ξεκινώντας από το $\hat{w}(K)$. Ο αλγόριθμος αυτός, όπως απέδειξε ο $Zwart$ το 1973 έχει το μειονέκτημα ότι μπορεί να κάνει κύκλους απείρως ανάμεσα σε ακραία σημεία του D που ελαχιστοποιούν την f τοπικά αλλά όχι και ολικά. Αυτό είναι προφανές αν σκεφτεί κανείς ότι χωρίς τον περιορισμό 3.2 το $\hat{w}(K)$ μπορεί να βρίσκεται έξω από τον κώνο K.

Ο αλγόριθμος των $Horst$ και Tuy είναι μια από τις πιο πρόσφατες προσπάθειες λύσης αυτού του προβλήματος. Ωστόσο και αυτός υπάρχει περίπτωση να μην τερματίσει ποτέ [3], αλλά άμα τερματίσει είναι εγγυημένο ότι θα έχει βρει το ακριβές ακραίο σημείο που θα είναιτο ολικό ελάχιστο της f.

Κεφάλαιο 4

Εφαρμογή του αλγορίθμου Horst και Tuy για την επίλυση του προβλήματος 1.1

4.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται η προγραμματιστική υλοποίηση του αλγορίθμου των Horst και Tuy για την επίλυση του προβλήματος 1.1. Στην παράγραφο 4.2 εξηγείται βήμα προς βήμα πως προσαρμόζεται ο αλγόριθμος στο πρόβλημα 1.1 ενώ στην παράγραφο 4.3 παραθέτεται ο ψευδοκώδικας με βάση τον οποίο υλοποιήθηκε το πρόγραμμα *generaltuy*. Αναλυτικά ο κώδικας όλων των προγραμμάτων που υλοποιήθηκαν μπορεί να βρεθεί στο κεφάλαιο 7.

4.2 Προσαρμογή του αλγορίθμου των Horst και Tuy στο πρόβλημα 1.1

Όπως αναφέρθηκε και στο πρώτο κεφάλαιο, σκοπός της εργασίας αυτής είναι η κατασκευή ενός αλγορίθμου που να επιλύει το πρόγραμμα τετραγωνικού προγραμματισμού:

$$min\{f(x) = \frac{1}{2}x^TQx + c^Tx : Ax \leq b, -1 \leq x_j \leq 1, \ j = 1...n\} \tag{4.1}$$

όταν η μήτρα Q είναι αρνητικά ορισμένη.

Παρατηρούμε ότι το πρόβλημα 4.1 είναι της μορφής (P), όπου $f = \frac{1}{2}x^TQx + c^Tx$ είναι μια κοίλη συνάρτηση και $D = \{x : Ax \leq b, -1 \leq x_j \leq 1, \ j = 1...n\}$ είναι ένα μη κενό κυρτό σύνολο. Συνεπώς πραγματευόμαστε ένα πρόβλημα κοίλης ελαχιστοποίησης, το οποίο έχει όλες τις ιδιότητες που αναφέρθηκαν στο κεφάλαιο 2. Για την επίλυση του θα προσαρμόσουμε κατάλληλα τον αλγόριθμο των Horst και Tuy.

Κλειδί σε αυτή την διαδικασία είναι η παρατήρηση ότι, επειδή $f : A \to \mathbb{R}$ και $A = \mathbb{R}^n$, το πρόβλημα 4.1 ικανοποιεί τις ιδιότητες 3 και 4 του κεφαλαίου 2 και επομένως θα έχει μια ολικά βέλτιστη λύση που θα είναι ακραίο σημείο του D. Συνεπώς μπορούμε να εκμεταλλευτούμε κατάλληλα τις τομές Tuy για να αποκλείσουμε σταδιακά τα ακραία σημεία του D και μετά από πεπερασμένο αριθμό βημάτων να βρούμε την ολικά βέλτιστη λύση.

Στη συνέχεια ακολουθούν τα βασικά βήματα του αλγορίθμου των Horst και Tuy, στα οποία επισημαίνονται οι αλλαγές που πρέπει να γίνουν για να προσαρμοστούν στα δεδομένα του προβλήματος 4.1:

Φάση *I*

Έστω n η διάσταση του προβλήματος ή αλλιώς ο αριθμός των γραμμών της μήτρας Q. Επιλέγουμε σταθερά $\varepsilon \geq 0$ και κάποιο αυστηρά επιτρεπτό τυχαίο σημείο y. Ξεκινώντας από το y, χρησιμοποιούμε το πρόγραμμα *localmin* για να βρούμε ένα ακραίο σημείο $x = x^0$ του D που να είναι τοπικό ελάχιστο της $f(x)$ πάνω στο D. Περισσότερες πληροφορίες για τον τρόπο λειτουργίας του προγράμματος αυτού μπορούν να βρεθούν στο παράρτημα.

Φάση *II*

Βήμα 1: Έστω $\alpha = f(x^o) - \varepsilon$. Λόγω της μορφής του συνόλου D, για να κατασκευάσουμε πολυεδρικό κώνο K^o με κορυφή το x^o και n γραμμικά ανεξάρτητους γεννήτορες $Y_i = (y^i - x^o), \ i = 1, 2, ..., n$ έτσι ώστε $K^o \supseteq D$ επιλέγουμε τα διανύσματα $Y_i, \ i = 1, 2, ..., n$ να βρίσκονται πάνω στα ευθύγραμμα τμήματα που ενώνουν το σημείο x^0 με τα γειτονικά του ακραία σημεία. Πρέπει να θέσουμε $CR = \{K^o\}$. Το CR όμως ουσιαστικά υποδηλώνει τον αριθμό των κώνων που έχουμε. Συνεπώς θα είναι $CR = 1$, αφού ο μοναδικός κώνος που έχουμε είναι ο K^o. Έστω Z^o ο $(n \times n)$ πίνακας του οποίου οι στήλες είναι οι γεννήτορες του K^o, δηλαδή τα σημεία $(y^i - x^o),\ i = 1, 2, ..., n$. Τότε $Z^o = -F^{-1}$, όπου F είναι ένας πίνακας τέτοιος ώστε $Fx^o = g$, δηλαδή εκφράζει το σύνολο των ενεργών περιορισμών.

Απόδειξη

Πέρα από τους περιορισμούς που εισάγει το σύνολο D, κατά τη διάρκεια της επίλυσης του προβλήματος 4.1, εισάγονται και κάποιοι επιπλέον περιορισμοί που εκφράζουν τις τομές Tuy. Οι περιορισμοί αυτοί αποθηκεύονται στις μήτρες A και b με τη μορφή επιπλέον γραμμών.

Θεωρούμε το σύνολο των ενεργών περιορισμών:
$x_i^o = 1,\ i \in J^+$
$x_i^o = -1,\ i \in J^-$
$\alpha_i^T x^o = b_i,\ i \in I'$, όπου $I' = \{1, 2, ..., la\}$ και la ο αριθμός των γραμμών της μήτρας A.

Συνοπτικά οι περιορισμοί αυτοί γράφονται $\begin{bmatrix} E_{J^+} \\ -E_{J^-} \\ A_{I'} \end{bmatrix} x^o = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ b_{I'} \end{bmatrix} \Leftrightarrow Fx^o = g$,

όπου E_{J^+} είναι μια μήτρα που σε κάθε γραμμή της έχει σε όλες τις θέσεις μηδέν εκτός από την i - θέση που έχει μονάδα. Ανάλογα ορίζεται η E_{J^-}. Οι μήτρες $A_{I'}$ και $b_{I'}$ απαρτίζονται από τις γραμμές των A και b, για τις οποίες ισχύει η σχέση $\alpha_i^T x^o = b_i,\ i \in I'$.

Ανάλογα ορίζουμε το σύνολο των μη ενεργών περιορισμών:
$x_j^o = 1,\ j \notin J^+$
$x_j^o = -1,\ j \notin J^-$
$\alpha_j^T x^o = b_j,\ j \notin I'$.

Συνοπτικά οι περιορισμοί αυτοί γράφονται $Hx^o < p$, όπου αν θεωρήσουμε ότι I είναι ο $n \times n$ μοναδιαίος πίνακας τότε $H = \begin{bmatrix} I\, εκτός\, E_{J^+} \\ -I\, εκτός\, E_{J^-} \\ A\, εκτός\, A_{I'} \end{bmatrix}$ και $p = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ b\, εκτός\, b_{I'} \end{bmatrix}$

Τότε, αν $e_i = [1\,1\,...\,1]^T$ θα ισχύει:

$$F(x^o + Y_i) = g + \epsilon e_i \Leftrightarrow Fx^o + FY_i = g + \epsilon e_i \Leftrightarrow FY_i = \epsilon e_i \Leftrightarrow Y_i = \epsilon F^{-1} e_i$$

$$H(x^o+Y_i)=Hx^o+HY_i=Hx^o+\epsilon HF^{-1}e_i<p+\epsilon HF^{-1}e_i$$

Η σχέση αυτή ισχύει όταν ϵ αρκετά μικρό. Επίσης θα πρέπει $\epsilon<0$ έτσι ώστε να ισχύει η σχέση $F(x^o+Y_i)=g+\epsilon e_i\leq g$. Συνεπώς προκύπτει ότι οι γεννήτορες του κώνου Y_i δίνονται από τη σχέση:

$$Y_i=-F^{-1}e_i,\ i=1,2,...,n$$

Q.E.D.

Στη συνέχεια βρίσκουμε την α - επέκταση $(z^i-x^o),\ i=1,2,...,n$ των σημείων $(y^i-x^o),\ i=1,2,...,n$. Συγκεκριμένα θα είναι:

$$(z^i-x^o)=-\frac{[(Qx^o+c)^TY_i+\sqrt{Qx^o+c)^TY_i-2Y_i^TQY_i\varepsilon}]}{Y_i^TQY_i}(y^i-x^o),\ i=1,2,...,n$$

Πράγματι από την παράγραφο 3.3 γνωρίζουμε ότι η α - επέκταση z^i του σημείου y^i δίνεται από τη σχέση:

$$\theta_i=sup\{\theta>0\mid[x^o+\theta(y^i-x^o)]\in A,\ \ f[x^o+\theta(y^i-x^o)]\geq\alpha\}$$

Όμως

$$\begin{aligned}
f(x^o+\theta Y_i)\geq\alpha,\ &\Leftrightarrow\ \frac{1}{2}(x^o+\theta Y_i)^TQ(x^o+\theta Y_i)+c^T(x^o+\theta Y_i)\geq\alpha\\
&\Leftrightarrow\ \frac{1}{2}x^{oT}Qx^0+c^Tx^o+\frac{1}{2}\theta^2Y_i^TQY_i+\theta(x^oQY_i+c^TY_i)\geq\alpha\\
&\Leftrightarrow\ \frac{1}{2}\theta^2Y_i^TQY_i+\theta(Qx^o+c)^TY_i\geq\alpha-f(x^o)\\
&\Leftrightarrow\ \frac{1}{2}\theta^2Y_i^TQY_i+\theta(Qx^o+c)^TY_i\geq-\varepsilon.
\end{aligned}$$

Συνεπώς καταλήγουμε στο ότι:

$$\theta_i=sup\{\frac{1}{2}\theta^2\alpha_1+\theta\alpha_2+\varepsilon\geq0,\ \theta>0\}\tag{4.2}$$

όπου $\alpha_1=Y_i^TQY_i<0$, γιατί η μήτρα Q είναι αρνητικά ορισμένη, και $\alpha_2=(Qx^o+c)^TY_i$. Παρατηρούμε ότι η ανισότητα ισχύει και για $\theta=0$. Η εξίσωση $\frac{1}{2}\theta^2\alpha_1+\theta\alpha_2+\varepsilon=0$ έχει γινόμενο ριζών $\rho_1\rho_2=\frac{2\varepsilon}{\alpha_1}<0$ και συνεπώς οι ρίζες είναι ετερόσημες. Επιπλέον η διακρίνουσα $\Delta=\sqrt{\alpha_2^2-2\alpha_1\varepsilon}$ είναι αυστηρά μεγαλύτερη του 0, πράγμα το οποίο σημαίνει ότι η εξίσωση θα είναι ομόσημη του α_1 μεταξύ των ριζών της, δηλαδή θα παίρνει θετικές τιμές σε αυτό το διάστημα. Η 4.2 ικανοποιείται όταν το θ_i ταυτίζεται με τη θετική ρίζα της εξίσωσης $\frac{1}{2}\theta^2\alpha_1 + \theta\alpha_2 + \varepsilon = 0$, δηλαδή όταν

$$\theta_i = -\frac{1}{2}[\alpha_2 + \sqrt{\alpha_2^2 - 2\alpha_1\varepsilon}] \Leftrightarrow \theta_i = -\frac{[(Qx^o + c)^T Y_i + \sqrt{Qx^o + c)^T Y_i - 2Y_i^T Q Y_i \varepsilon}]}{Y_i^T Q Y_i}$$

$\square$

Επίσης είναι σημαντικό να αναφέρουμε ότι στο βήμα 1 ορίζουμε επίσης μια μήτρα II, η οποία έχει διάσταση $n \times n \times CR$ και στην οποία αποθηκεύεται η Z^o και γενικά όλοι οι $(n \times n)$ πίνακες των οποίων οι στήλες είναι οι γεννήτορες του εκάστοτε κώνου.

Βήμα 2: Για κάθε κώνο λύνουμε το πρόβλημα γραμμικού προγραμματισμού $LP(Z, D)$, όπου Z είναι ο $(n \times n)$ πίνακας του οποίου οι στήλες είναι οι γεννήτορες του εκάστοτε κώνου K, λαμβάνοντας υπ΄ όψη τη συνθήκη 3.2 και τους περιορισμούς που ενδέχεται να εισάγουν οι μήτρες A και b, αν δεν είναι κενές. Έστω $\hat{w}(K)$ η λύση.

Αν $f[\hat{w}(K)] < \alpha$ για κάποιο κώνο K, θέτουμε:

$$D = D \cap \{x \in \mathbb{R}^n \mid < e^T (Z^o)^{-1}, x - x^o > \ \geq 1\}$$

. Η συνθήκη αυτή ικανοποιείται με την επικόλληση της γραμμής $-e^T Z^o$ στη μήτρα A και του στοιχείου $-e^T Z^o x^o - 1$ στη μήτρα b. Στη συνέχεια, ξεκινώντας από το $\hat{w}(K)$, βρίσκουμε με τη βοήθεια του προγράμματος *localmin* ένα καινούργιο ακραίο σημείο x^o του D που να είναι τοπικό ελάχιστο της f πάνω στο D. Επιστρέφουμε στο βήμα 1.

Αν $f[\hat{w}(K)] \geq \alpha$ για όλα τα K τότε συνεχίζουμε στο Βήμα 2.

Βήμα 3: Αν $< e^T Z^{-1}, \hat{w}(K) - x^o > \leq 1$ για κάθε κώνο τότε τερματισμός: Το x^o είναι ολικό ε - ελάχιστο. Αλλιώς, αφαιρούμε από το CR κάθε κώνο $K \in CR$ για τον οποίο ισχύει $< e^T Z^{-1}, \hat{w}(K) - x^o >> 1$. Δηλαδή το CR μειώνεται κατά ένα. Ταυτόχρονα αφαιρούμε και τον ανάλογο πίνακα Z από τη μήτρα II. Στη συνέχεια θέτοντας $\hat{u} = \hat{w}(K)$ υποδιαιρούμε, με τη βοήθεια των εξισώσεων 3.3, 3.4, τον κώνο K σε δυο ή περισσότερους υποκώνους, αυξάνουμε κατάλληλα την τιμή του CR και επικολλάμε τις μήτρες Z, που οι στήλες τους είναι οι γεννήτορες των υποκώνων, στο τέλος της μήτρας II . Επιστρέφουμε στο Βήμα 2.

4.3 Προγραμματιστική υλοποίηση του αλγορίθμου των Horst και Tuy

Στην ενότητα αυτή παρουσιάζεται ο **ψευδοκώδικας** που χρησιμοποιήθηκε για την προγραμματιστική υλοποίηση του αλγορίθμου των Horst και Tuy. Το ολοκληρωμένο πρόγραμμα, γραμμένο σε *Matlab*, παρατίθεται στην παράγραφο 7.2.

Φάση *I*

Επιλέγουμε σταθερά $\varepsilon \geq 0$ και κάποιο σημείο y αυστηρά επιτρεπτό. Ξεκινώντας από το y, βρίσκουμε ένα ακραίο σημείο x^0 του D που να είναι τοπικό ελάχιστο της $f(x)$. Έστω n και la ο αριθμός των γραμμών των πινάκων Q και A αντίστοιχα.

Φάση *II*

Βήμα 1:

Έστω $\alpha = f(x^o) - \varepsilon$, $CR = 1$, $II = \emptyset$, $Z^o = \emptyset$, $X1 = \emptyset$, $U1 = \emptyset$, $e = [1\,1\ldots 1]^T$.
For i=1,2,...n: if $|x_i^o| = 1$ τότε επικόλληση στη μήτρα Z^o της γραμμής $sgn(x_i^o)e_i^T$.
For i=1,2,...la: if $\alpha_i^T x^o - b_i = 1$ τότε επικόλληση στη μήτρα Z^o της γραμμής α_i^T.
$Z^o = -(Z^o)^{-1}$
for i=1,2,...n
$Y_i = Z^o e_i$ η i - στήλη της μήτρας Z^o
$\alpha_1 = Y_i^T Q Y_i$, $\alpha_2 = (Qx^o + c)^T Y_i$
$\theta_i = -\frac{1}{2}(\alpha_2 + \sqrt{\alpha_2^2 - 2\alpha_1\varepsilon})$
$Z_i^o = \theta_i Y_i$, όπου Z_i^o η i - στήλη της μήτρας Z^o
$u00 = (Z^o)^{-T} e$
end

Βήμα 2:

For i=1,2,...CR
$Z = II(n,n,i)$
$ZZ = Z^{-1}$
$xoo = -ZZx^o$
$u1 = ZZ^T e$

Επίλυση του προβλήματος γραμμικού προγραμματισμού :

$$\min_{x\in\mathbb{R}^n}\{-u1^Tx:\,-1\leq x_j\leq 1,\,j=1,2,...,n,\,-ZZx\leq xoo,\,Ax\leq b\}$$

. Έστω $x1$ η λύση. Επικολλάμε στον πίνακα $X1$ τη στήλη $x1$ και στον πίνακα $U1$ τη στήλη $u1$. Στη συνέχεια υπάρχουν οι ακόλουθες δυνατές περιπτώσεις:

- Αν για κάποιο $x1$ ισχύει $f(x1)<\alpha$ τότε επικόλληση στη μήτρα A της γραμμής $-u00^T$ και στον πίνακα b του στοιχείου $-u00^Tx^o-1$. Χρησιμοποιώντας το πρόγραμμα *localmin* με αρχικό σημείο το $x1$ βρίσκουμε ένα τοπικό ελάχιστο του προβλήματος:

  $$\min_{x}\{\frac{1}{2}x^TQx+c^Tx:\,-1\leq x_j\leq 1,\,j=1,2,...,n,\,Ax\leq b\}$$

  . Επιστροφή στο βήμα 1.

- Διαφορετικά συνεχίζουμε στο βήμα 3.

end for

Βήμα 3:

Αν $u1^T(x1-x^o)\leq 1$ για κάθε $i=1,2,...,CR$ τότε τερματισμός: Το x^o είναι ένα ολικό ε - ελάχιστο.

Αλλιώς,

$CRold=CR$

$ii=1$

$while\quad ii\leq CRold$

$if\quad u1^T(x1-x^o)>1$ τότε

%Αφαίρεση του Κ από το CR

$CR=CR-1$

$Z=II(n,n,ii)$

$Z_j=Z_{j+1},\,j=ii,ii+1,...,CR,$ όπου $Z_j=II(n,n,j)$

%Υποδιαίρεση του Κ σε υποκώνους

Εύρεση $l1$ τέτοιου ώστε $Z\cdot l1=x1-x^o$

$L=\sum_{i=1}^{n}l1_i$

$l=\frac{l1}{L}$

$v=Z\cdot l$

%Βρίσκουμε την α - επέκταση του v

$\alpha_1 = v^T Q v$

$\alpha_2 = (Qx^o + c)^T \cdot v$

$thita1 = -\frac{1}{\alpha_1}(\alpha_2 + \sqrt{\alpha_2^2 - 2\alpha_1 \varepsilon})$

$v = thita1 \cdot v$

$for \quad j = 1, 2, ..., n$

$if \quad l_j > 0$ τότε

$CR = CR + 1$

Έστω Z_{CR} είναι η μήτρα που στήλες της είναι οι γεννήτορες του νέου υποκώνου CR. Τότε η Z_{CR} προκύπτει από το Z αν αντικαταστήσουμε τη j - στήλη με το διάνυσμα v.

$end \quad if$

$end \quad for$

$end \quad if$

$ii = ii + 1$

$end \quad while$

$X1 = \emptyset$, $U1 = \emptyset$

Επιστροφή στο βήμα 2.

Κεφάλαιο 5

Παραδείγματα

5.1 Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζονται τα αποτελέσματα του προγράμματος *generaltuy* για διάφορες τυχαίες μήτρες Q και c και τυχαίο αρχικό σημείο. Στην παράγραφο 5.2 αναλύεται βήμα προς βήμα ο αλγόριθμος των *Horst* και *Tuy* για μήτρα $Q : 2 \times 2$ και με τις μήτρες A και b αρχικά κενές, ενώ στη επόμενη παράγραφο παρουσιάζονται τα τελικά αποτελέσματα του προγράμματος *generaltuy* όταν η μήτρα Q είναι πολύ μεγαλύτερης διάστασης. Τέλος στην παράγραφο 5.4 παρατίθεται ένα παράδειγμα όπου οι μήτρες A και b εισάγουν κάποιους αρχικούς περιορισμούς. Αξίζει να σημειώσουμε ότι σε όλα τα παραδείγματα οι μήτρες Q και c, καθώς και το αρχικό σημείο y δημιουργήθηκαν με τη βοήθεια της συνάρτησης *rand* του *matlab* για να είναι το αποτέλεσμα όσο το δυνατόν πιο αξιόπιστο.

5.2 Γραφική εξήγηση του αλγορίθμου για μήτρα $Q : 2 \times 2$

Στην παράγραφο αυτή παρουσιάζεται μια γραφική εξήγηση του αλγορίθμου των *Horst* και *Tuy*. Με τη βοήθεια των σχημάτων ο αναγνώστης μπορεί να δει εποπτικά τα διαδοχικά βήματα του αλγορίθμου, να καταλάβει την έννοια των τομών *Tuy* και πως, με τη βοήθεια τους, καταλήγουμε στο ολικό ελάχιστο του προβλήματος 4.1.

Για το σκοπό αυτό θεωρούμε τις μήτρες:

```
Q =
    -5   -14
   -14   -52


c =
    -9
     6
```

και το αρχικό διάνυσμα y:

```
y =
    0.3891
    0.2426
```

Επίσης $A = []$ και $b = []$. Με αφετηρία το σημείο αυτό το πρόγραμμα *localmin* καταλήγει στο τοπικό ελάχιστο

```
x0 =
    1.0000
    1.0000
```

Στη συνέχεια προχωράμε στο βήμα 1. Με ε=1.9 ορίζουμε $\alpha = f(x0) - \varepsilon = -47.4$. Η μήτρα $Z0$ που περιέχει τους γεννήτορες του κώνου CR είναι

```
Z0 =
    -1     0
     0    -1
```

και μετά την εύρεση της α - επέκτασης για κάθε γεννήτορα ξεχωριστά παίρνουμε τελικά ότι:

```
Z0 =
  -11.2675         0
         0   -2.3389
```

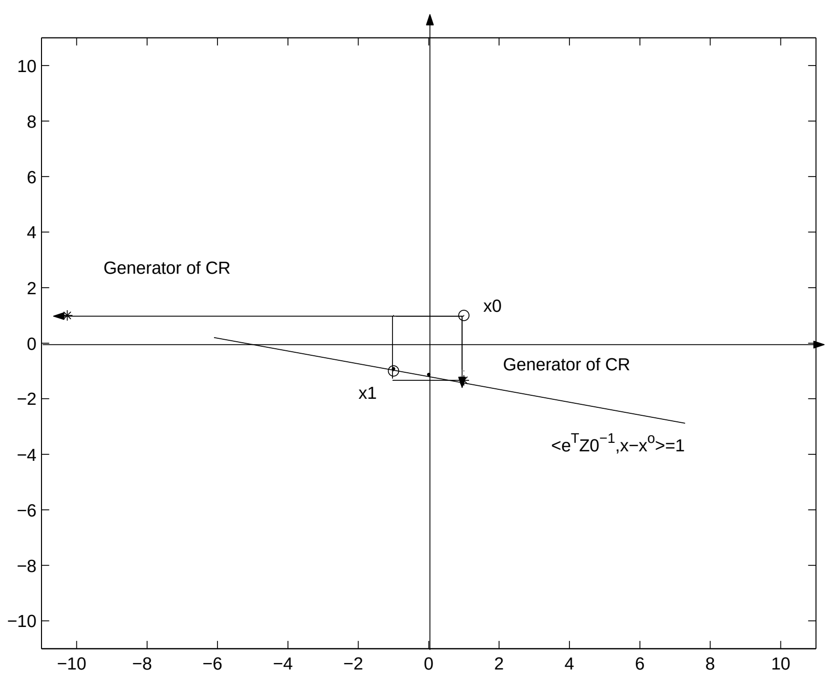
Στη συνέχεια υπολογίζεται η ανίσωση $< e^T Z0^{-1}, x - x^o > \geq 1$, η οποία αποτελεί μια τομή Tuy. Αντικαθιστώντας τα μέχρι τώρα δεδομένα βρίσκουμε ότι κάθε νέο σημείο $x0 = [x01 \quad x02]'$ που υπολογίζουμε και για το οποίο ισχύει $f(x0) < \alpha$ πρέπει επίσης να ικανοποιεί τη σχέση:

$$-0.0888x01 - 0.4276x02 \geq 0.4836$$

Στη συνέχεια λύνουμε το πρόβλημα γραμμικού προγραμματισμού και βρίσκουμε σημείο $x1$ που να ικανοποιεί την παραπάνω ανισότητα. Το σημείο αυτό χρησιμεύει ως αφετηρία στην ερευνά μας για ένα νέο σημείο $x0$. Από το *output* του προγράμματος προέκυψε ότι

```
x1 =
    -1.0000
    -1.0000
```

Τα μέχρι τώρα βήματα φαίνονται στο παρακάτω σχήμα:



Σχήμα 1

Επειδή ισχύει $f(x1) > \alpha$ συνεχίζουμε στο βήμα 3. Όμως το $x1$ δεν ικανοποιεί τη συνθήκη τερματισμού και συνεπώς πρέπει να χωρίσουμε τον κώνο CR σε υποκώνους με τη βοήθεια των εξισώσεων 3.3 και 3.4 αφού θέσουμε $\hat{u} = x1$. Αφού βρήκαμε το u υπολογίσαμε την α - επέκταση του v:

```
v =
    -2.0920
    -2.0920
```

Στη συνέχεια με τη βοήθεια του v χωρίζουμε τον κώνο CR σε δυο υποκώνους. Οι γεννήτορες του πρώτου και του δεύτερου υποκώνου είναι αντίστοιχα:

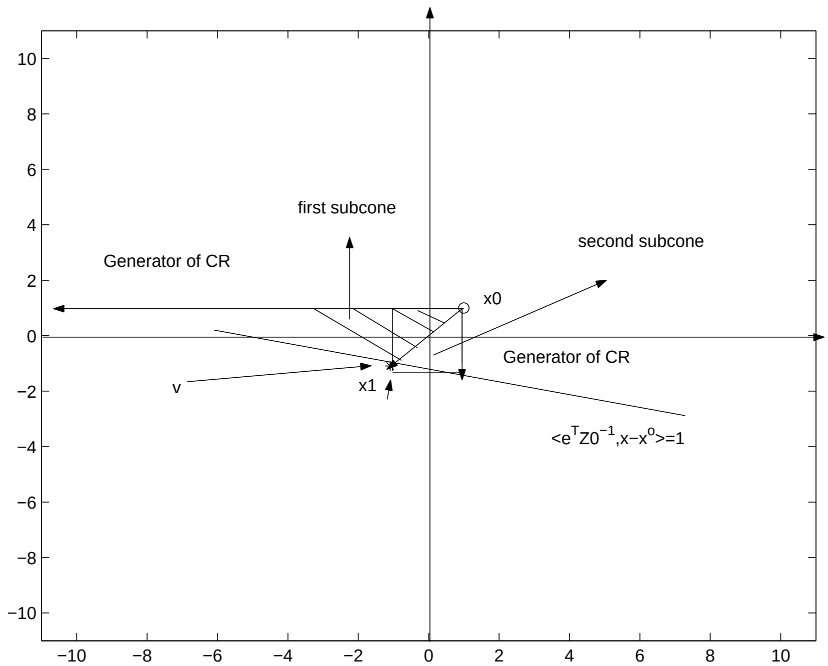
```
I(:,:,1) =
    -2.0920         0
    -2.0920   -2.3389


I(:,:,2) =
   -11.2675   -2.0920
          0   -2.0920
```

Επιστρέφουμε στο βήμα 2.

Στο βήμα 2 λύνουμε το πρόγραμμα γραμμικού προγραμματισμού για κάθε υποκώνο ξεχωριστά. Και στις δυο περιπτώσεις προκύπτει όμως ότι $x1 = [-1 \ -1]'$. Επειδή $f(x1) > \alpha$ συνεχίζουμε στο βήμα 3, όπου και το πρόγραμμα τερματίζει γιατί η συνθήκη τερματισμού ικανοποιείται. Συνεπώς η συνάρτηση

$$f(x) = \frac{1}{2}x^T Qx + c^T x : -1 \leq x_j \leq 1, \, j = 1, 2$$

παρουσιάζει ολικό ελάχιστο στο σημείο $x0 = [1 \quad 1]'$ Τα παραπάνω παρουσιάζονται γραφικά στο ακόλουθο σχήμα:



Σχήμα 2

5.3 Γενικά παραδείγματα εφαρμογής του προγράμματος *generaltuy* όταν οι μήτρες A και b είναι κενές

Παράδειγμα 1

Στην παράγραφο αυτή παραθέτουμε το αποτέλεσμα του προγράμματος *generaltuy* για μήτρα Q διάστασης 8×8. Το πρόγραμμα κατέληξε στο σωστό ολικό ελάχιστο μέσα σε 10 επαναλήψεις. Πρέπει να σημειώσουμε ότι η τιμή της θετικής σταθεράς ε ήταν ίση με 1.5. Οι μήτρες Q, c που χρησιμοποιήθηκαν ήταν:

```
Q =

  Columns 1 through 7

-59.3766    7.8335   23.0066   39.1484   -1.9757   -1.1642    8.8052
  7.8335  -39.1637  -16.2008  -11.2549   -4.3094    5.4046   -7.9216
 23.0066  -16.2008  -52.6618  -11.6051   -9.5478   28.1552   -1.8292
```

```
  39.1484  -11.2549  -11.6051  -79.3608  -22.4938  -19.4904    3.3586
  -1.9757   -4.3094   -9.5478  -22.4938  -34.5086   -6.7199   16.1832
  -1.1642    5.4046   28.1552  -19.4904   -6.7199  -38.5210   12.7858
   8.8052   -7.9216   -1.8292    3.3586   16.1832   12.7858  -50.4131
  19.5434    3.6486  -29.0676  -25.0699   -2.8680   26.5906   -3.4234

 Column 8

  19.5434
   3.6486
 -29.0676
 -25.0699
  -2.8680
  26.5906
  -3.4234
 -48.6475

 c = [-1.3463  -3.5996  0.6677  3.2301  1.7395  4.9945  4.6164
 -4.4114]'
```

και το αρχικό διάνυσμα y ήταν:

```
y = [0.1068  -0.4160  0.7159  -0.3285  0.3604  -0.8931  -0.2867
-0.0034]'
```

Τέλος το σημείο τερματισμού του προγράμματος που συμπίπτει με το ολικό ελάχιστο είναι:

```
x0 = [-1  1  1  1  1  -1  1  1]'
```

Παράδειγμα 2

Στην παράγραφο αυτή παραθέτουμε το αποτέλεσμα του προγράμματος *generaltuy* για μήτρα Q διάστασης 30×30. Το πρόγραμμα κατέληξε στο σωστό ολικό ελάχιστο μέσα σε μόλις 2 επαναλήψεις! Στη συνέχεια παρουσιάζονται οι μήτρες που χρησιμοποιήθηκαν καθώς και τα τελικά αποτελέσματα.

```
Q = 1.0e+007 *

  Columns 1 through 6

   -0.9896   -0.7744   -0.6972   -0.8213   -0.7380   -0.6592
   -0.7744   -1.0412   -0.7325   -0.8665   -0.7959   -0.7487
   -0.6972   -0.7325   -0.9008   -0.7146   -0.7631   -0.7553
   -0.8213   -0.8665   -0.7146   -1.0535   -0.7860   -0.7258
   -0.7380   -0.7959   -0.7631   -0.7860   -1.0200   -0.7859
   -0.6592   -0.7487   -0.7553   -0.7258   -0.7859   -0.9996
   -0.7067   -0.7303   -0.6903   -0.8419   -0.7657   -0.7414
   -0.7875   -0.9102   -0.7662   -0.8430   -0.7699   -0.8364
   -0.8058   -0.8128   -0.7624   -0.8196   -0.8060   -0.7361
   -0.6261   -0.7454   -0.7798   -0.7808   -0.7764   -0.7772
   -0.8017   -0.8316   -0.8043   -0.8511   -0.7751   -0.7139
   -0.6903   -0.6762   -0.6331   -0.6908   -0.6960   -0.6913
   -0.7094   -0.7443   -0.6507   -0.7170   -0.7372   -0.7636
   -0.8767   -0.8845   -0.8291   -0.8774   -0.9450   -0.9267
   -0.7787   -0.8425   -0.7719   -0.9475   -0.8011   -0.8554
   -0.8297   -0.8629   -0.6890   -0.8679   -0.7381   -0.7223
   -0.8183   -0.8750   -0.8270   -0.8651   -0.8572   -0.7924
   -0.6706   -0.7466   -0.7293   -0.6602   -0.7365   -0.7804
   -0.6151   -0.6848   -0.6027   -0.6622   -0.7428   -0.7217
   -0.8815   -0.8526   -0.8695   -0.8978   -0.9855   -0.8942
   -0.4751   -0.6129   -0.5431   -0.6137   -0.6190   -0.6208
   -0.8242   -0.7784   -0.7547   -0.9199   -0.8219   -0.7874
   -0.7387   -0.7457   -0.7787   -0.7606   -0.7851   -0.7764
   -0.7643   -0.7746   -0.7292   -0.8511   -0.7923   -0.7119
   -0.6557   -0.6381   -0.6296   -0.6559   -0.6393   -0.5768
   -0.6999   -0.8755   -0.6693   -0.8449   -0.7254   -0.8044
   -0.5585   -0.6251   -0.6309   -0.6231   -0.6200   -0.6795
   -0.6848   -0.8119   -0.6493   -0.7687   -0.7626   -0.7910
```

-0.7867	-0.8836	-0.7780	-0.8876	-0.8822	-0.8581
-0.6896	-0.7774	-0.6612	-0.7628	-0.6633	-0.6309

Columns 7 through 12

-0.7067	-0.7875	-0.8058	-0.6261	-0.8017	-0.6903
-0.7303	-0.9102	-0.8128	-0.7454	-0.8316	-0.6762
-0.6903	-0.7662	-0.7624	-0.7798	-0.8043	-0.6331
-0.8419	-0.8430	-0.8196	-0.7808	-0.8511	-0.6908
-0.7657	-0.7699	-0.8060	-0.7764	-0.7751	-0.6960
-0.7414	-0.8364	-0.7361	-0.7772	-0.7139	-0.6913
-0.9291	-0.8144	-0.7886	-0.7828	-0.7582	-0.7020
-0.8144	-1.2127	-0.8864	-0.8642	-0.8600	-0.7544
-0.7886	-0.8864	-1.0257	-0.7532	-0.8009	-0.7452
-0.7828	-0.8642	-0.7532	-0.9487	-0.8066	-0.6445
-0.7582	-0.8600	-0.8009	-0.8066	-1.1280	-0.7631
-0.7020	-0.7544	-0.7452	-0.6445	-0.7631	-0.8402
-0.6300	-0.7039	-0.6989	-0.6576	-0.7850	-0.6035
-0.8787	-1.0001	-0.9777	-0.8304	-0.9202	-0.8724
-0.9531	-0.9890	-0.8451	-0.9031	-0.9247	-0.8168
-0.6763	-0.8306	-0.7787	-0.6926	-0.7937	-0.6261
-0.7973	-0.8067	-0.8477	-0.8258	-0.9188	-0.7250
-0.6640	-0.7627	-0.6813	-0.7708	-0.8762	-0.7021
-0.6743	-0.7458	-0.6803	-0.7357	-0.7608	-0.6413
-0.9325	-0.9313	-0.9121	-0.9122	-0.8678	-0.8115
-0.6667	-0.6436	-0.6804	-0.5956	-0.6340	-0.5893
-0.8139	-0.8771	-0.8054	-0.7883	-0.9334	-0.7325
-0.7735	-0.8869	-0.7867	-0.8031	-0.9149	-0.7756
-0.7424	-0.8535	-0.7204	-0.7012	-0.8318	-0.6327
-0.6213	-0.7169	-0.7201	-0.6290	-0.7293	-0.6579
-0.7164	-0.7882	-0.7174	-0.7362	-0.8156	-0.6185
-0.6389	-0.7189	-0.6501	-0.6873	-0.6286	-0.4674

-0.7803	-0.8069	-0.7371	-0.6993	-0.8561	-0.6818
-0.8887	-0.9273	-0.8577	-0.8971	-0.8412	-0.7495
-0.6711	-0.7179	-0.6117	-0.6583	-0.7240	-0.5357

Columns 13 through 18

-0.7094	-0.8767	-0.7787	-0.8297	-0.8183	-0.6706
-0.7443	-0.8845	-0.8425	-0.8629	-0.8750	-0.7466
-0.6507	-0.8291	-0.7719	-0.6890	-0.8270	-0.7293
-0.7170	-0.8774	-0.9475	-0.8679	-0.8651	-0.6602
-0.7372	-0.9450	-0.8011	-0.7381	-0.8572	-0.7365
-0.7636	-0.9267	-0.8554	-0.7223	-0.7924	-0.7804
-0.6300	-0.8787	-0.9531	-0.6763	-0.7973	-0.6640
-0.7039	-1.0001	-0.9890	-0.8306	-0.8067	-0.7627
-0.6989	-0.9777	-0.8451	-0.7787	-0.8477	-0.6813
-0.6576	-0.8304	-0.9031	-0.6926	-0.8258	-0.7708
-0.7850	-0.9202	-0.9247	-0.7937	-0.9188	-0.8762
-0.6035	-0.8724	-0.8168	-0.6261	-0.7250	-0.7021
-0.8787	-0.8070	-0.7672	-0.7658	-0.7843	-0.7002
-0.8070	-1.2854	-1.0724	-0.8501	-0.9211	-0.8599
-0.7672	-1.0724	-1.2130	-0.8323	-0.8740	-0.8272
-0.7658	-0.8501	-0.8323	-1.0194	-0.8501	-0.6556
-0.7843	-0.9211	-0.8740	-0.8501	-1.1242	-0.8272
-0.7002	-0.8599	-0.8272	-0.6556	-0.8272	-0.9171
-0.6821	-0.8321	-0.8129	-0.7107	-0.7478	-0.7212
-0.7802	-1.0686	-1.0240	-0.7799	-0.9633	-0.8194
-0.5518	-0.7882	-0.7490	-0.5438	-0.5576	-0.5329
-0.7167	-0.9425	-0.9382	-0.8466	-0.8279	-0.7428
-0.6918	-0.9304	-0.8943	-0.7886	-0.9218	-0.8436
-0.6547	-0.8973	-0.8405	-0.7975	-0.7794	-0.6762
-0.5178	-0.7922	-0.7170	-0.5826	-0.6741	-0.6261
-0.7400	-0.8661	-0.8424	-0.8305	-0.8203	-0.7407

-0.5747	-0.7110	-0.7119	-0.6552	-0.6699	-0.6183
-0.6870	-0.9324	-0.8495	-0.6696	-0.8532	-0.7589
-0.7771	-1.0474	-1.0244	-0.8248	-0.9087	-0.8483
-0.5772	-0.7430	-0.7887	-0.6708	-0.7668	-0.6276

Columns 19 through 24

-0.6151	-0.8815	-0.4751	-0.8242	-0.7387	-0.7643
-0.6848	-0.8526	-0.6129	-0.7784	-0.7457	-0.7746
-0.6027	-0.8695	-0.5431	-0.7547	-0.7787	-0.7292
-0.6622	-0.8978	-0.6137	-0.9199	-0.7606	-0.8511
-0.7428	-0.9855	-0.6190	-0.8219	-0.7851	-0.7923
-0.7217	-0.8942	-0.6208	-0.7874	-0.7764	-0.7119
-0.6743	-0.9325	-0.6667	-0.8139	-0.7735	-0.7424
-0.7458	-0.9313	-0.6436	-0.8771	-0.8869	-0.8535
-0.6803	-0.9121	-0.6804	-0.8054	-0.7867	-0.7204
-0.7357	-0.9122	-0.5956	-0.7883	-0.8031	-0.7012
-0.7608	-0.8678	-0.6340	-0.9334	-0.9149	-0.8318
-0.6413	-0.8115	-0.5893	-0.7325	-0.7756	-0.6327
-0.6821	-0.7802	-0.5518	-0.7167	-0.6918	-0.6547
-0.8321	-1.0686	-0.7882	-0.9425	-0.9304	-0.8973
-0.8129	-1.0240	-0.7490	-0.9382	-0.8943	-0.8405
-0.7107	-0.7799	-0.5438	-0.8466	-0.7886	-0.7975
-0.7478	-0.9633	-0.5576	-0.8279	-0.9218	-0.7794
-0.7212	-0.8194	-0.5329	-0.7428	-0.8436	-0.6762
-0.8452	-0.8582	-0.5881	-0.7673	-0.8281	-0.6707
-0.8582	-1.2899	-0.6724	-0.9238	-0.9094	-0.8578
-0.5881	-0.6724	-0.7880	-0.6324	-0.6425	-0.6201
-0.7673	-0.9238	-0.6324	-1.0578	-0.8521	-0.8975
-0.8281	-0.9094	-0.6425	-0.8521	-1.0679	-0.8308
-0.6707	-0.8578	-0.6201	-0.8975	-0.8308	-1.0053
-0.5852	-0.7548	-0.5407	-0.7265	-0.7067	-0.6961

-0.7224	-0.8579	-0.6529	-0.8316	-0.7466	-0.8207
-0.6090	-0.6983	-0.5112	-0.6324	-0.6906	-0.6605
-0.6995	-0.8770	-0.5846	-0.8081	-0.7658	-0.7230
-0.8619	-1.0031	-0.7095	-0.9002	-0.9079	-0.8307
-0.5302	-0.8319	-0.5112	-0.7088	-0.6357	-0.7158

Columns 25 through 30

-0.6557	-0.6999	-0.5585	-0.6848	-0.7867	-0.6896
-0.6381	-0.8755	-0.6251	-0.8119	-0.8836	-0.7774
-0.6296	-0.6693	-0.6309	-0.6493	-0.7780	-0.6612
-0.6559	-0.8449	-0.6231	-0.7687	-0.8876	-0.7628
-0.6393	-0.7254	-0.6200	-0.7626	-0.8822	-0.6633
-0.5768	-0.8044	-0.6795	-0.7910	-0.8581	-0.6309
-0.6213	-0.7164	-0.6389	-0.7803	-0.8887	-0.6711
-0.7169	-0.7882	-0.7189	-0.8069	-0.9273	-0.7179
-0.7201	-0.7174	-0.6501	-0.7371	-0.8577	-0.6117
-0.6290	-0.7362	-0.6873	-0.6993	-0.8971	-0.6583
-0.7293	-0.8156	-0.6286	-0.8561	-0.8412	-0.7240
-0.6579	-0.6185	-0.4674	-0.6818	-0.7495	-0.5357
-0.5178	-0.7400	-0.5747	-0.6870	-0.7771	-0.5772
-0.7922	-0.8661	-0.7110	-0.9324	-1.0474	-0.7430
-0.7170	-0.8424	-0.7119	-0.8495	-1.0244	-0.7887
-0.5826	-0.8305	-0.6552	-0.6696	-0.8248	-0.6708
-0.6741	-0.8203	-0.6699	-0.8532	-0.9087	-0.7668
-0.6261	-0.7407	-0.6183	-0.7589	-0.8483	-0.6276
-0.5852	-0.7224	-0.6090	-0.6995	-0.8619	-0.5302
-0.7548	-0.8579	-0.6983	-0.8770	-1.0031	-0.8319
-0.5407	-0.6529	-0.5112	-0.5846	-0.7095	-0.5112
-0.7265	-0.8316	-0.6324	-0.8081	-0.9002	-0.7088
-0.7067	-0.7466	-0.6906	-0.7658	-0.9079	-0.6357
-0.6961	-0.8207	-0.6605	-0.7230	-0.8307	-0.7158

```
   -0.7591   -0.5804   -0.5272   -0.6583   -0.7310   -0.5133
   -0.5804   -1.0298   -0.6491   -0.7707   -0.8811   -0.7672
   -0.5272   -0.6491   -0.7510   -0.6438   -0.7553   -0.4682
   -0.6583   -0.7707   -0.6438   -1.0197   -0.8146   -0.6400
   -0.7310   -0.8811   -0.7553   -0.8146   -1.1775   -0.7043
   -0.5133   -0.7672   -0.4682   -0.6400   -0.7043   -0.8392

c = [27.1904  793.6670  999.2308  110.2385  622.6012  132.5718
310.0298  134.7877  223.3263  396.5468  135.1435  241.0587
927.5161  391.1009  511.2628  92.8961  21.6989  159.5348  844.5159
879.1535  186.9895  991.3049  712.0298  871.3646  479.6312
496.0049  287.5319  60.9415  262.4677  186.2610]'

y = [0.8342  -0.7534  -0.9731  -0.2606  0.3973  0.7787  0.1875
  -0.6866 -0.3666  -0.5332  -0.9832  -0.2062  0.2997  -0.8300
  0.5376  0.9394  0.4296  0.5639  -0.5249  -0.6085  -0.4736
  0.4276  0.9552  0.2742  0.0918  0.6961  0.6042  0.3366  0.3420
  0.6413]'

x0 = [-1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1]'
```

Παράδειγμα 3

Σε αντίθεση με το προηγούμενο παράδειγμα, που παρά τη μεγάλη του διάσταση, το πρόγραμμα *generaltuy* πέτυχε να βρει το ολικό ελάχιστο σε ελάχιστο χρόνο, στην προκειμένη περίπτωση δεν μπόρεσε να καταλήξει σε κάποιο αποτέλεσμα αν και η μήτρα Q ήταν μόλις 4×4. Μετά από 20 επαναλήψεις το πρόγραμμα δεν είχε ακόμα τερματίσει. Στη συνέχεια παρουσιάζονται οι μήτρες που χρησιμοποιήθηκαν καθώς και το αρχικό σημείο y.

```
Q =
  -104      0    -58    -94
     0    -66      7     -6
```

```
   -58      7    -70     34
   -94     -6     34   -285
```

```
c = [3  14  -8  -22]'
```

```
y = [-0.8596  -0.9619  0.0166  0.7388]'
```

5.4 Παραδείγμα εφαρμογής του προγράμματος *generaltuy* όταν οι μήτρες A και b δεν είναι κενές

Παράδειγμα 1

Στο παράδειγμα αυτό χρησιμοποιήθηκαν οι μήτρες:

```
Q =
  Columns 1 through 8

  -56.2113  -18.2485   -1.1428    6.7532  -13.3415   -4.1746  -15.1668   -3.5810
  -18.2485  -81.9986   19.8300   30.4452  -15.1214    3.8003  -36.7710   38.1003
   -1.1428   19.8300 -111.9144   -5.4252  -10.9673  -13.6773  -44.3810   -8.6741
    6.7532   30.4452   -5.4252  -67.9119  -12.2264  -13.8303   22.3174    2.4102
  -13.3415  -15.1214  -10.9673  -12.2264  -83.2738  -10.6874  -48.1641  -16.8287
   -4.1746    3.8003  -13.6773  -13.8303  -10.6874 -102.7139  -17.4954   -3.7543
  -15.1668  -36.7710  -44.3810   22.3174  -48.1641  -17.4954 -119.9470   -0.7579
   -3.5810   38.1003   -8.6741    2.4102  -16.8287   -3.7543   -0.7579  -58.2614
  -16.9797  -48.7212    1.2395  -11.7040  -49.0410   23.1091  -31.5135   34.7678
   26.5590   42.1517  -43.4976  -11.4038    0.0813  -22.6720  -27.3243   -2.0378

  Columns 9 through 10

  -16.9797   26.5590
  -48.7212   42.1517
    1.2395  -43.4976
```

```
  -11.7040  -11.4038
  -49.0410    0.0813
   23.1091  -22.6720
  -31.5135  -27.3243
   34.7678   -2.0378
  -83.5549   16.3236
   16.3236  -84.1716
```

```
c = [-2.1727  2.6994  2.2758  4.1461  -3.9388  3.8894  0.9504
-4.8792  -2.6169  -3.7638]'
```

Οι επιπλέον περιορισμοί του προβλήματος περιγράφονται από τις μήτρες A και b:

```
A =
  Columns 1 through 8

   10.8000    0.5000    9.2000    2.3000    1.5000    8.0000    0.2000    3.5000
    0.9000   -0.8000    0.6500    0.1300    0.7600    0.4500    0.2800   -0.6000

  Columns 9 through 10

   -5.0000   -1.0000
    0.1000   -0.5000


b =
   -0.7000
    2.5000
```

Το τυχαίο αρχικό σημείο y επιλέχτηκε να είναι αυστηρά επιτρεπτό:

```
y = [-0.6131  0.3644  -0.3945  0.0833  -0.6983  0.3958  -0.2433
0.7200 0.7073  0.1871]'
```

Το πρόγραμμα *generaltuy* βρήκε το ολικό ελάχιστο $x0$ του γενικού προβλήματος 4.1 σε μόλις 5 επαναλήψεις:

```
x0 = [1  1  -1  -1  1  -1  1  -1  1  -1]'
```

Κεφάλαιο 6

Παρατηρήσεις - Συμπεράσματα

1. Ο αλγόριθμος των *Horst* και *Tuy* περιλαμβάνει μια σειρά από βήματα. Κάθε φορά που χρησιμοποιούμε το πρόγραμμα *localmin* για να βρούμε ένα τοπικό ελάχιστο σημαδεύει και μια καινούργια επανάληψη. Κατά τη διάρκεια της επανάληψης τόσο το σημείο x^o όσο και το κομμάτι του συνόλου D που δεν έχει ακόμα εξερευνηθεί δεν αλλάζουν, αλλά από τη μια επανάληψη στην άλλη το σύνολο D μειώνεται από μια τομή $Tuy < e^T(Z^o)^{-1}, x - x^o > \;\; \geq 1$, ενώ το x^o αντικαθίσταται από μια καλύτερη κορυφή του συνόλου D. Συνεπώς κάτω από αυτές τις συνθήκες μπορεί να δει κανείς ότι κάθε στιγμή το τρέχων x^o ικανοποιεί αυστηρά όλες τις προηγούμενες τομές. Επίσης αν το x^o είναι κορυφή του τρέχοντος συνόλου στην i επανάληψη, τότε θα είναι και κορυφή του αρχικού συνόλου D.

2. Η τιμή της σταθεράς ε$\geq$ 0 μπορεί και να επιλεχθεί από τον χρήστη. Ωστόσο υπενθυμίζουμε ότι ο αλγόριθμος βρίσκει ένα ολικό ε - ελάχιστο. Πρακτικά αυτό σημαίνει ότι εάν το ε είναι μεγάλο τότε θα χρειάζονται λίγες επαναλήψεις, ωστόσο θα έχουμε μικρή ακρίβεια κατά την εύρεση της λύσης. Από την άλλη μεριά, εάν το ε είναι μικρό, τότε θα συμβαίνει ακριβώς το αντίθετο: θα χρειάζονται πολλές επαναλήψεις μεν αλλά θα έχουμε επιτύχει μεγάλη ακρίβεια. Επιπλέον υπενθυμίζουμε ότι το ε παίζει πολύ μεγάλο ρόλο στον αλγόριθμο αφού ορίζει την αληθοτιμή της ποσότητας $f(x^o) < \alpha$. Όταν το ε είναι μεγάλο και η διαφορά ανάμεσα στην κορυφή που δίνει τη βέλτιστη λύση και τη δεύτερη καλύτερη κορυφή είναι μικρή, υπάρχει περίπτωση ο αλγόριθμος να μας οδηγήσει σε εσφαλμένο

αποτελέσματα. Αυτό όμως μπορεί να αποφευχθεί με κατάλληλη επιλογή του ε. Ωστόσο δεν είναι δυνατόν να γνωρίζουμε εκ των προτέρων ποια είναι η διαφορά στις τιμές που παίρνει η συνάρτηση f στις δυο καλύτερες κορυφές, κάτι το οποίο συνιστά ένα πρόβλημα κατά την επιλογή του ε.

3. Προκειμένου να ελεγχθεί κατά πόσο το πρόγραμμα *generaltuy* τερμάτιζε σε σωστό σημείο δοκιμάστηκαν μήτρες με μεγάλες διαστάσεις. Παρατηρήθηκε ότι όσο μεγάλωνε η διάσταση του προβλήματος αυξάνονταν κατακόρυφα οι επαναλήψεις που χρειαζόταν να εκτελέσει το πρόγραμμα πριν τερματίσει. Επίσης λόγω του μεγάλου αριθμού δεδομένων που έπρεπε να κρατήσει ο υπολογιστής στη μνήμη αυξάνονταν επίσης και οι υπολογιστικές απαιτήσεις. Ωστόσο υπήρξαν περιπτώσεις που αν και η διάσταση ήταν μεγάλη (π.χ. στο παράδειγμα 2, παράγραφος 5.3) το πρόγραμμα τελείωνε σε ελάχιστο χρόνο. Σε γενικές γραμμές όμως, και ειδικά σε μικρές διαστάσεις, μπορεί να παρατηρήσει κανείς ότι το πρόγραμμα της επαλήθευσης βρίσκει το ολικό ελάχιστο γρηγορότερα από το πρόγραμμα *generaltuy*. Παρόλα αυτά δεν μειώνεται η χρησιμότητα του αλγορίθμου των *Horst* και *Tuy*, ο οποίος έχει χρησιμοποιηθεί επανειλημμένα σε διάφορες εφαρμογές, κάτι το οποίο οφείλεται στη γενική διατύπωση του.

4. Κατά τη μελέτη της συμπεριφοράς του προγράμματος *generaltuy* όταν οι μήτρες A και b είναι κενές και όταν εισάγουν κάποιους επιπλέον γραμμικούς περιορισμούς διαπιστώθηκε ότι στη δεύτερη περίπτωση, λόγω της αυξημένης πολυπλοκότητας, το πρόγραμμα χρειαζόταν μεγαλύτερο χρόνο και έκανε περισσότερες επαναλήψεις προτού καταλήξει στο ολικό ε - ελάχιστο, ακόμα και αν ξεκινούσε από το ίδιο αρχικό σημείο. Η σύγκριση έγινε χρησιμοποιώντας κάθε φορά τις ίδιες μήτρες Q και c. Η διαφορά στον αριθμό των επαναλήψεων είναι αμελητέα για πολύ μικρές διαστάσεις, γίνεται όμως αισθητή όσο η διάσταση του προβλήματος μεγαλώνει.

Κεφάλαιο 7

Παράρτημα

7.1 Γενικά για το $Matlab$

Όλα τα προγράμματα που υλοποιήθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας γράφτηκαν με τη γλώσσα προγραμματισμού του $Matlab$. Το $Matlab$ ($Matrix$ $laboratory$) είναι ένα σύγχρονο εργαλείο της πληροφορικής και σήμερα αποτελεί ένα ολοκληρωμένο λογισμικό πακέτο για επιστημονικό και αριθμητικό υπολογισμό με πολύ καλή υποστήριξη γραφικών [5]. Αν και ,εν γένει, κάποιες γλώσσες προγραμματισμού (όπως η C ή η $C++$) είναι πολύ ταχύτερες από τη γλώσσα προγραμματισμού του $Matlab$, αυτό ωστόσο παρέχει σημαντικές διευκολύνσεις όσον αφορά τη λύση μαθηματικών προβλημάτων. Συγκεκριμένα είναι ειδικά σχεδιασμένο για υπολογισμούς επί πινάκων, επίλυση γραμμικών συστημάτων, εύρεση ιδιοτιμών και ιδιοδιανυσμάτων κοκ. Επίσης χρησιμοποιεί αλγόριθμους υψηλής αξιοπιστίας και ακρίβειας και παρέχει έτοιμα προγράμματα και συναρτήσεις προς διευκόλυνση του χρήστη. Οι ιδιότητες αυτές καθιστούν το $Matlab$ το ιδανικό εργαλείο για την επίλυση του προβλήματος 4.1 και γενικότερα προβλημάτων βελτιστοποίησης. Τέλος αξίζει να σημειωθεί ότι επειδή το $Matlab$ είναι αριθμητικό πρόγραμμα, πράγμα που σημαίνει ότι χρησιμοποιεί αριθμητική πεπερασμένης ακρίβειας και παράγει προσεγγιστικές λύσεις, είμαστε λίγο πιο ελαστικοί όσον αφορά στην ικανοποίηση κάποιον ισοτικών περιορισμών (π.χ. στο πρόγραμμα $generaltuy$ η συνθήκη $if\,|x_i^o| = 1$ έχει γραφτεί ως: $if\,|x^o(i)| - 1 \leq 1e - 06$).

7.2 Το πρόγραμμα *generaltuy*

Το πρόγραμμα *generaltuy* αποτελεί την προγραμματιστική υλοποίηση του αλγορίθμου των *Horst* και *Tuy* με σκοπό την επίλυση του προβλήματος 4.1. Το πρόγραμμα παίρνει ως εισόδους την αρνητικά ορισμένη μήτρα Q, το c, τις μήτρες A και b και τη μεταβλητή *repeats*. Όπως αναφέρθηκε στην παράγραφο 3.3 υπάρχει περίπτωση ο αλγόριθμος των *Horst* και *Tuy* να μην τερματίσει. Για το σκοπό αυτό ο χρήστης έχει τη δυνατότητα να εισάγει το μέγιστο αριθμό επαναλήψεων που θέλει να κάνει το πρόγραμμα πριν τερματίσει βίαια (δηλαδή χωρίς να εκπληρωθεί η συνθήκη τερματισμού). Αξίζει να σημειωθεί ότι για το συμβολισμό της σταθεράς $\varepsilon \geq 0$ χρησιμοποιήθηκε το *ee*. Οι μεταβλητές w και h καθορίζουν τη μετάβαση από το ένα βήμα στο άλλο ανάλογα με την τιμή που παίρνουν (0 ή 1). Το $n3$ είναι η μεταβλητή που ελέγχει το τερματισμό του προγράμματος. Έχει καθ' όλη τη διάρκεια του προγράμματος την τιμή 0, όταν όμως η συνθήκη τερματισμού ικανοποιηθεί τότε παίρνει την τιμή 1 και το πρόγραμμα τερματίζει.

Στη συνέχεια ακολουθεί ο κώδικας του προγράμματος *generaltuy*:

```
function[x0,y]=generaltuy(Q,c,A,b,repeats)

w=0; h=0; n3=0; CR=1; k=0;
X1=[ ];
II=[ ];
[n n2]=size(Q);
[la la2]=size(A);
[wb1 wb2]=size(b);
S=zeros(n,1);

% Bhma 0
if (la==0 & wb1==0)
    y=-ones(n,1)+2*rand(n,1);
else
    while (sum(S)~=wb1)
        y=-ones(n,1)+2*rand(n,1);
```

```
        S=A*y<b;
    end
end
x0=localmin(A,b,Q,c,y);
ee=1.9;

%Bhma 1
while(CR~=0)
    while(w==0)
        if (h==0)
            X1=[ ];
            U1=[ ];
            Z0=[ ];
            II=[ ];
            Z1=diag(sign(x0));
            f=0.5*x0'*Q*x0+c'*x0;
            a=f-ee;
            CR=1;
            for i=1:n
                if (abs(x0(i))-1<=1e-06)
                    Z0=[Z0;Z1(i,:)];
                end
            end
            for i=1:la
                if ((A(i,:)*x0-b(i,:))==0)
                    Z0=[Z0;A(i,:)];
                end
            end
            e=ones(n,1);
            Z0=-inv(Z0);
            for i=1:n
                y2=Z0(:,i);
```

```
        a1=y2'*Q*y2;
        a2=(Q*x0+c)'*y2;
        thita(i)=-(1/a1)*(a2+sqrt(a2^2-2*a1*ee));
        Z0(:,i)=thita(i)*y2;
    end
    II(:,:,CR)=Z0;
    u00=Z0'\e;
end

%Bhma 2
k=k+1
for i=1:CR
    Z=II(:,:,i);
    ZZ=inv(Z);
    x00=-ZZ*x0;
    u1=ZZ'*e;
    U1=[U1,u1];
    AA=[A;-ZZ];
    bb=[b;x00];
    x1 = linprog(-u1,AA,bb,[ ],[ ],-ones(n,1),ones(n,1));
    X1=[X1,x1];
    if ((0.5*x1'*Q*x1+c'*x1)<a)
        la=la+1;
        A=[A;-u00'];
        b=[b;-u00'*x0-1];
        x0=localmin(A,b,Q,c,x1);
        w=0;
        h=0;
        break
    else
        w=1;
        h=1;
```

```
        end
    end
end

%Bhma 3
w=0;
aa=0;
for i=1:CR
    if ((U1(:,i)'*(X1(:,i)-x0))<=1)
       aa=aa+1;
    end
end
if (aa==CR)
    n3=1
    break
end
if (n3~=1)
    CRold=CR;
    ii=1;
    while (ii<=CRold)
        if ((U1(:,ii)'*(X1(:,ii)-x0))>1)
            CR=CR-1;
            Z=II(:,:,ii);
            for j=ii:CR
                II(:,:,j)=II(:,:,j+1);
            end
            l1=Z\(X1(:,ii)-x0);
            L=sum(l1);
            l=l1./L;
            [l11 l22]=size(l);
            v=Z*l;
            a1=v'*Q*v;
```

```
            a2=(Q*x0+c)'*v;
            thita1=-(1/a1)*(a2+sqrt(a2^2-2*a1*ee));
            v=thita1*v;
            for j=1:l11
                if (l(j)>1e-04)
                    CR=CR+1;
                    Z2=Z;
                    Z2(:,j)=v;
                    II(:,:,CR)=Z2;
                end
            end
        end
        ii=ii+1;
    end
    X1=[ ];
    U1=[ ];
    if k==repeats
        CR=0;
        break
    end
end
end
```

7.3 Το πρόγραμμα *localmin*

Το πρόγραμμα *localmin* είναι η προγραμματιστική υλοποίηση ενός αλγορίθμου βελτιστοποίησης, ο οποίος ανήκει στις μεθόδους καθόδου. Αυτό σημαίνει ότι κάθε νέο σημείο x_{k+1} της ακολουθίας που κατασκευάζει ο αλγόριθμος μειώνει την τιμή της συνεχούς συνάρτησης f, δηλαδή θα ισχύει $f(x_{k+1}) < f(x_k)$. Ο αλγόριθμος εύρεσης τοπικού ελαχίστου απαρτίζεται από δυο βασικά στοιχεία: την κατεύθυνση έρευνας $h(x_k)$ και το μήκος βήματος τ_k, με τρόπο ώστε

$$x_{k+1} = x_k + \tau_k h(x_k), \text{ όπου } \tau_k \in \mathbb{R}_+^*$$

Το μήκος βήματος είναι ,εν γένει, το ελάχιστο της συνάρτησης $f(\cdot)$ κατά μήκος της κατεύθυνσης h. Ο αλγόριθμος εύρεσης τοπικού ελαχίστου τερματίζει όταν βρεθεί σημείο που να ικανοποιεί τις αναγκαίες συνθήκες βελτίστου 1ης τάξης για το πρόβλημα:

$$min\{f(x) = \frac{1}{2}x^TQx + c^Tx : \ Ax \leq b, -1 \leq x_j \leq 1, \ j = 1...n\} \tag{7.1}$$

Το πρόβλημα 7.1 είναι ένα πρόβλημα με ανισοτικούς περιορισμούς και για τη λύση του θα χρησιμοποιηθεί η μέθοδος προβολής της κλίσεως [1]. Κατά τον υπολογισμό της κατεύθυνσης έρευνας θα πρέπει να εξασφαλίζεται ότι αυτή είναι επιτρεπτή. Υπενθυμίζουμε τον εξής ορίσμό:

Ορισμός 1

Ένα διάνυσμα $h_k \in \mathbb{R}^n$ λέγεται επιτρεπτή κατεύθυνση για ένα σύνολο L στο σημείο $x_k \in L$ αν και μόνο αν $\bar{\tau}_i > 0$ τέτοιο ώστε $x_k + \tau h_k \in L$ για κάθε $\tau \in [0, \bar{\tau}]$. Προφανώς πρέπει $intL \neq \emptyset$.

Για να εξασφαλίσουμε ότι $x_k \in L$, $k = 1, 2, ...$ και $f(x_{k+1}) \leq f(x_k)$, $k = 1, 2, ...$ απαιτούμε η κατεύθυνση έρευνας h_k να είναι επιτρεπτή κατεύθυνση για το L στο x_k και ταυτόχρονα να είναι κατεύθυνση καθόδου για την f, δηλαδή πρέπει $\nabla f(x_k)^T h_k \leq 0$. Ωστόσο μπορεί να παρατηρήσει κανείς ότι μόνο οι ενεργοί περιορισμοί στο σημείο x_k αρκούν για τον προσδιορισμό μιας επιτρεπτής κατεύθυνσης. Την παρατήρηση αυτή, μαζί με την παρακάτω πρόταση εκμεταλλεύεται η μέθοδος προβολής της κλίσεως, πάνω στην οποία στηρίζεται το πρόγραμμα *localmin*:

Πρόταση 1

Θεωρούμε το γενικό πρόβλημα βελτιστοποίησης

$$\min_x\{f(x) : \ g_j(x) \leq 0, \ j = 1, 2, ..., l\}$$

. Αν $x^* \in \mathbb{R}^n$ είναι τοπικό ελάχιστο του προβλήματος αυτού, τότε το x^* είναι επίσης τοπικό ελάχιστο του προβλήματος

$$\min_x\{f(x) : \ g_j(x) = 0, \ j \in I(x^*)\}$$

Εκμεταλλευόμενοι αυτή την πρόταση θα μπορούσαμε να λύσουμε αντί του προβλήματος 7.1 το αντίστοιχο πρόβλημα που θα περιέχει μόνο ισοτικούς περιορισμούς.

Ωστόσο το σύνολο $I(x^*)$ δεν είναι γνωστό εκ των προτέρων. Για αυτό το λόγο η μέθοδος προβολής της κλίσεως εντοπίζει συστηματικά, με διαδοχικές δοκιμές, το σύνολο $I(x^*)$.

Το πρόβλημα 7.1 γράφεται και ως:

$$min\{f(x) = \frac{1}{2}x^TQx + c^Tx : A0x \leq b0\} \tag{7.2}$$

όπου αν I είναι η μοναδιαία μήτρα και $e^T = [1\,1\ldots1]$ τότε $A0^T = [I \quad -I \quad A]$ και $b0^T = [e \quad e \quad b]$.

Έστω $A0 : m \times n$ και $b0 \in \mathbb{R}^m$. Ορίζουμε τότε το ενεργό σύνολο

$$I0 = \{i \in \{1, 2, ..., m\} : a0_i^Tx = b0_i\}$$

και τη μήτρα $A1$ που έχει ως γραμμές τις γραμμές της $A0$ για τις οποίες ισχύει $a0_i^Tx = b0_i$. Για να υπολογίσουμε την κατεύθυνση έρευνας λύνουμε το πρόβλημα ελαχιστοποίησης:

$$\min_h\{\nabla f(x_k)^Th : a0_i^Th = 0,\, i \in I0,\, h^Th = 1\} \tag{7.3}$$

Ο περιορισμός $h^Th = 1$ προστίθεται για να έχει το πρόβλημα 7.3 πεπερασμένη λύση ενώ ο περιορισμός $a0_i^Th = 0$, $i \in I0$ είναι ένας ισοδύναμος τρόπος γραφής του περιορισμού $a0_i^Tx = b0_i$, $\forall\, i \in I0$. Οι συνθήκες βελτίστου στη λύση h του 7.3 είναι:

$$\nabla f(x_k) + \sum_{i \in I0} \lambda_i \cdot a0_i + 2\rho \cdot h = 0 \tag{7.4}$$

$$a0_i^T \cdot h = 0,\, i \in I0 \tag{7.5}$$

$$h^T \cdot h = 1 \tag{7.6}$$

Από τη σχέση (7.4) παίρνουμε ότι:

$$h = -\frac{1}{2\rho}(\nabla f(x_k) + \sum_{i \in I0} \lambda_i \cdot a0_i)$$

και από τη σχέση (7.5) ισχύει ότι:

$$A1 \cdot h = 0 \Leftrightarrow -\frac{1}{2\rho}A1(\nabla f(x_k) + A1^T\lambda) = 0 \Leftrightarrow A1A1^T\lambda = -A1\nabla f(x_k) \Leftrightarrow$$

$$\Leftrightarrow \lambda = -(A1 \cdot A1^T)^{-1} A1 \cdot \nabla f(x_k)$$

Οπότε με αντικατάσταση στην (7.4) και λαμβάνοντας υπ' όψη και την (7.6) προκύπτει τελικά ότι:

$$h = -[I - A1^T \cdot (A1 \cdot A1^T)^{-1} \cdot A1] \nabla f(x_k)$$

Μπορεί να αποδειχθεί ([1]) ότι η κατεύθυνση h είναι επιτρεπτή κατεύθυνση για το σύνολο L και ταυτόχρονα είναι κατεύθυνση καθόδου για τη συνάρτηση f.

Στη συνέχεια πρέπει να υπολογιστεί το μήκος βήματος. Κάθε νέο σημείο $x_{k+1} = x_k + \tau \cdot h_k$ θα πρέπει να ικανοποιεί τους αρχικούς περιορισμούς, δηλαδή θα πρέπει:

$$A0(x_k + \tau_k \cdot h_k) \leq b0 \Leftrightarrow a0_i^T (x_k + \tau_k \cdot h_k) \leq b0_i,\ i \notin Io$$

$$\Leftrightarrow \tau_k \cdot a0_i^T \cdot h_k = b0_i - a0_i^T \cdot x_k \geq 0$$

Και επειδή $\tau_i \in \mathbb{R}_+^*$ προκύπτει τελικά ότι:

$$\tau_k = \min_i \{ \frac{b0_i - a0_i^T \cdot x_k}{a0_i^T \cdot h_k} : a0_i^T \cdot h_k > 0,\ i \notin I0 \}$$

Το πρόγραμμα *localmin* τερματίζει όταν βρεθεί σημείο που να ικανοποιεί τις αναγκαίες συνθήκες βελτίστου 1ης τάξης για το αρχικό πρόβλημα. Πρακτικά αυτό σημαίνει ότι η σταθερά *Lagrangre* $\mu = -(A1^T)^{-1} \nabla f(x_k)$ πρέπει να είναι μεγαλύτερη ή ίση με το μηδέν. Αν υπάρχει $i \in I0$ τέτοιο ώστε $\mu_i < 0$ τότε στην επόμενη επανάληψη $(k+1)$ λαμβάνεται σα δοκιμαστικό σύνολο ενεργών ανισοτικών περιορισμών το $A1' = A1 - \{i\}$.

Στη συνέχεια ακολουθεί ο **ψευδοκώδικας** του προγράμματος *localmin*:

Βήμα 0:

Έστω αρχικό σημείο x τέτοιο ώστε $A0 \cdot y < bo$ και $I0 = \emptyset$, $A1 = [a0_i^T]_{i \in I0}$, $k = 0$.

Βήμα 1:

Υπολογισμός κατεύθυνσης έρευνας:

- Αν η μήτρα $A1$ είναι κενή τότε $h_k = -\nabla f(x_k)$
- Αν η μήτρα $A1$ δεν είναι κενή τότε

$$h_k = -[I - A1^T \cdot (A1 \cdot A1^T)^{-1} \cdot A1] \nabla f(x_k).$$

Αν $||h_k|| = 0$ τότε πήγαινε στο βήμα 4.

Βήμα 2:

Υπολογισμός μήκους βήματος:

$$\tau_k = \min_i \{ \frac{b0_i - a0_i^T \cdot x_k}{a0_i^T \cdot h_k} : a0_i^T \cdot h_k > 0,\, i \notin I0 \}$$

Ας είναι ρ το $i \notin I0$ για το οποίο συμβαίνει το παραπάνω ελάχιστο.

Βήμα 3:

Ενημέρωση:

$x_{k+1} = x_k + \tau_k \cdot h_k, \quad I0 = I0 \cup \{\rho\}$. Επικολλάμε τη γραμμή $a0_\rho^T$ στη μήτρα $A1$ και θέτουμε $k = k+1$. Επιστροφή στο βήμα 1.

Βήμα 4:

Υπολογισμός $\mu = -(A1^T)^{-1} \nabla f(x_k)$.

- Εάν $\mu_i \geq 0,\ i = 1, 2, ..., n$ τότε τερματισμός.
- Εάν υπάρχουν i τέτοια ώστε $\mu_i < 0$ τότε θέστε

$$q = arg \min_i \{\mu_i\}$$

  και αφαιρέστε τη γραμμή q από τη μήτρα $A1$. Επιστροφή στο βήμα 1.

Στη συνέχεια ακολουθεί ο κώδικας του προγράμματος *localmin*. Αξίζει να σημειωθεί ότι οι μεταβλητές w και q καθορίζουν τη μετάβαση από το ένα βήμα στο άλλο ανάλογα με την τιμή που παίρνουν (0 ή 1) ενώ το l δείχνει τον αριθμό των επαναλήψεων.

```
function[y]=localmin(A,b,Q,c,y)
A1=[ ];
l=1;
w=0;
q=0;
e=1e-012;
[n n2]=size(Q);
A0=[eye(n);-eye(n);A];
b0=[ones(2*n,1);b];
```

```
[a01 ao2]=size(A0);

%Bhma 0
for i=1:n
    if (y(i)==1)|(y(i)==-1)
        y=0.99*y;
    end
end
f=0.5*y'*Q*y+c'*y;
I0=[ ];
for i=1:a01
    I1(i)=i;
end
k=0;
A1=A0(I0,:);
%Bhma 1
while (l~=(a01+1))
    while (w==0)
        [a11,a12]=size(A1);
        gradfy=Q*y+c;
        if (a11~=0)
            h=-gradfy + A1'*((A1*A1')\(A1*gradfy));
        else
            h=-gradfy;
        end
        g=norm(h);
        if (norm(h)<e)     %Bhma 4
            mu=-A1'\gradfy;
            i3=mu>=0;
            [i31 i32]=size(i3);
            if (sum(i3)==i31)
                q=1;
```

```
            break
        else
            [mumin,q]=min(mu);
            A1(q,:)=[];
        end
    else
        w=1;
    end
end
if (q==1)
    break
end
w=0;
%Bhma 2
AUX1=A0(I1,:)*h;
I2=AUX1>0;
AUX2=(b0(I1(I2))-A0(I1(I2),:)*y)./(AUX1(I2));
[tau in]=min(AUX2);
I12=I1(I2);
p=I12(in);

%Bhma 3
y=y+tau*h;
I0(l)=p;
l=l+1;
I1;
[i11 i12]=size(I1);
for i=1:i12
    if I1(1,i)==p
        I1(:,i)=[];
        break
    end
```

```
    end
    I1;
    A1=[A1;A0(p,:)];
    f=0.5*y'*Q*y+c'*y;
    k=k+1;
end
```

7.4 Το πρόγραμμα επαλήθευσης *globalmin*

Το πρόγραμμα επαλήθευσης είναι ένα μικρό πρόγραμμα που δημιουργήθηκε για την επαλήθευση της εξόδου του προγράμματος *generaltuy*, όταν οι μήτρες A και b είναι κενές. Η λογική του είναι απλή. Όπως ήδη αναφέρθηκε, το πρόβλημα 4.1 ικανοποιεί τις ιδιότητες 3 και 4 του κεφαλαίου 2 και επομένως θα έχει ένα ολικό ελάχιστο που να είναι ακραίο σημείο. Το πρόγραμμα *globalmin* αρχικά δημιουργεί μια μήτρα X που έχει ως στήλες της όλα τα ακραία σημεία του συνόλου $D = \{x : -1 \leq x_j \leq 1,\ j = 1...n\}$. Αν n είναι ο αριθμός των γραμμών της Q τότε το X είναι ένας πίνακας με διάσταση $(n \times 2^n)$. Στη συνέχεια γίνεται ο έλεγχος για να βρεθεί ποιο είναι το ολικό ελάχιστο. Για κάθε ακραίο σημείο του D, δηλαδή για κάθε στήλη του πίνακα X, υπολογίζεται η τιμή της συνάρτησης και αποθηκεύεται στον πίνακα f. Αφού έχουν εξεταστεί όλα τα ακραία σημεία βρίσκεται ποιο από αυτά είναι το ολικό ελάχιστο.

Στη συνέχεια ακολουθεί ο κώδικας του προγράμματος *globalmin*:

```
function[x,fx]=globalmin(Q,c)

[n n2]=size(Q);
f=zeros(1,2^n);
x=zeros(n,1);
X=zeros(n,2^n);
l=2;

% Sxhmatismos pinaka X
for i=1:n
    for m=1:1:2^(i-1)
```

```
        for j=m:2^i:2^n
            X(i,j)=1;
        end
        for j=(l-1+m):2^i:2^n
            X(i,j)=-1;
        end
    end
        l=l+2^(i-1);
end

%Elegxos elaxistou
for i=1:2^n
    x=X(:,i);
    f(i)=0.5*x'*Q*x + c'*x;
end [p,t]=min(f);
x=X(:,t); %dianusma pou dinei to elaxisto
fx=0.5*x'*Q*x + c'*x;
```

Βιβλιογραφία

[1] *Ν.Γ. Μαράτος*, Τεχνικές Βελτιστοποίησης (Σημειώσεις), (1990).

[2] *Reiner Horst and Panos M. Pardalos*, Non Convex Optimization and its Applications - Handbook of Global Optimization.

[3] *Reiner Horst and Hoang Tuy*, Global Optimization - Deterministic Approaches, (1995).

[4] *Ν. Παπαγεωργίου*, Κυρτή Ανάλυση (Σημειώσεις), (2001).

[5] *Γ. Παπαγεωργίου, Χαράλαμπος Τσίτουρας, Ιωάννης Φάμελης*, Εισαγωγή και Εφαρμογές στο Mathematica & Matlab, (2003).

[6] *Mangasarian*, (1969).

[7] *Rockafellar*, (1970).

[8] *Luenberger*, (1984).

[9] *Bazaraa, Sherali, Shetty*, (1993).

[10] *Philips, Rosen*, (1993).

[11] *Danninger, Bornze*, (1993).

[12] *Neumaier*, (1992).

[13] *Efroymson, Ray*, (1966).

[14] *Florian, Robillard, Achim*, (1972).

[15] *Veinott*, (1969).

[16] *Zangwill*, (1966).

[17] *Bracken, McCormick*, (1968).

[18] *Hirsch, Dantzig*, (1954).

[19] *Balinski*, (1961).

[20] *Cabot*, (1974).

[21] *Tuy*, (1964).

[22] *Murty*, (1968).

[23] *Falk, Soland*, (1969).

[24] *Byrd, Dert, Rinooy, Kan et al.*, (1990).

[25] *Philips, Rosen, Van Vliet*, (1992).

[26] *Rosen, Van Vliet*, (1987).