



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΡΙΘΜΗΤΙΚΑ ΠΕΙΡΑΜΑΤΑ ΜΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ SEQUENTIAL QUADRATIC PROGRAMMING

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Λ. Νίκου

Επιβλέπων : Νικόλαος Γ. Μαράτος
Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2012



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

ΑΡΙΘΜΗΤΙΚΑ ΠΕΙΡΑΜΑΤΑ ΜΕ ΤΟΝ ΑΛΓΟΡΙΘΜΟ SEQUENTIAL QUADRATIC PROGRAMMING

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέξανδρος Λ. Νίκου

Επιβλέπων : Νικόλαος Γ. Μαράτος
Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 17^η Οκτωβρίου 2012.

.....
Ν. Μαράτος
Καθηγητής ΕΜΠ

.....
Τ. Κουσιουρής
Καθηγητής ΕΜΠ

.....
Κ. Τζαφέστας
Αν. Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2012

.....

Αλέξανδρος Λ. Νίκου

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αλέξανδρος Νίκου 2012.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η επίλυση προβλημάτων βελτιστοποίησης με ισοτικούς περιορισμούς. Τα προβλήματα με ανισοτικούς περιορισμούς μπορούν εύκολα να μετασχηματιστούν σε ισοδύναμα με ισοτικούς. Η αναλυτική λύση του προβλήματος προϋποθέτει την επίλυση συστήματος μη γραμμικών εξισώσεων κάτι που είναι πολύπλοκο και μπορεί να επιτευχθεί για προβλήματα χαμηλής τάξης και πολυπλοκότητας. Χρησιμοποιούνται επαναληπτικές αλγοριθμικές διαδικασίες που έχουν εφαρμογή σε οποιοδήποτε πρόβλημα. Στα πλαίσια της εργασίας υλοποιήθηκε ο αλγόριθμος Επαναληπτικού Τετραγωνικού Προγραμματισμού (SQP) στο περιβάλλον του MATLAB. Ο αλγόριθμος εφαρμόστηκε σε 24 προβλήματα τα οποία αντλήθηκαν από τη βιβλιογραφία και προέρχονται από επιστημονικούς τομείς όπως οικονομία, χημεία, βιολογία κτλ. Καταγράφηκαν τα αποτελέσματα που παρατηρήθηκαν σε πίνακες και σχήματα και έγινε σύγκριση με τα αποτελέσματα άλλων αλγορίθμων που υπάρχουν στο πακέτο λογισμικού. Από τα αριθμητικά πειράματα που έγιναν ο αλγόριθμος αποδείχθηκε αρκετά σταθερός και αξιόπιστος και κατάλληλος για την επίλυση οποιασδήποτε μορφής την οποία μπορεί να πάρει το πρόβλημα του βελτιστοποίησης, ανάλογα με το είδος και το πλήθος των περιορισμών του.

Λέξεις Κλειδιά

τεχνικές βελτιστοποίησης, ισοτικοί περιορισμοί, μη γραμμικός προγραμματισμός, αλγόριθμος Επαναληπτικού Τετραγωνικού Προγραμματισμού, αριθμητικά πειράματα

Abstract

The scope of this thesis was the solution of optimization problems with equality constraints. The problems with inequality constraints can easily be transformed in equivalent ones with equality constraints. The analytical solution of the problem requires the solution of a system of non-linear equations, which is very complex and can be achieved for problems of low class and complexity. Repetitive algorithmic procedures are used, which can be applied to any problem. For the purpose of this thesis, the Sequential Quadratic Problem (SQP) algorithm was used in MATLAB environment. The algorithm was applied in 24 problems from various fields such as economy, chemistry, biology etc, which were derived from bibliography. The results were depicted in tables and schemes and compared with results from other algorithms of the software package. From the numerical experiments that were performed, the algorithm was shown to be quite stable and reliable and appropriate for the solution of any admissible form which the optimization problem can take, in proportion with the type and number of its constraints.

Key Words

optimization techniques, equality constraints, nonlinear programming, Sequential Quadratic Programming algorithm, SQP, numerical experiments

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε την περίοδο Φεβρουάριος 2012 – Σεπτέμβριος 2012 στη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών ΕΜΠ με αρκετό κόπο και προσωπική εργασία.

Θα ήθελα να ευχαριστήσω, πάνω από όλους, τον καθηγητή μου κύριο Νικόλαο Μαράτο, που με εμπιστεύτηκε για την εργασία και μου παρείχε τις πολύτιμες γνώσεις του και την βοήθειά του σε οποιοδήποτε πρόβλημα αντιμετώπισα. Η καθοδήγησή του υπήρξε καταπληκτική και η εν γένει συνεργασία μας καθ' όλη τη διάρκεια εκπόνησης της εργασίας ήταν άψογη.

Επίσης θέλω να ευχαριστήσω τον πατέρα μου Λευτέρη, τη μητέρα μου Χρυσάνθη και την αδερφή μου Βιργινία καθώς στέκονται πάντα δίπλα μου. Τους φίλους μου Αρετή, Γιώργο, Κώστα και Μαργαρίτη για όλα αυτά που έζησα μαζί τους και κατάφεραν να χρωματίσουν τα φοιτητικά μου χρόνια με έντονες πινελιές. Τέλος, θέλω να ευχαριστήσω τη γενιά μου γιατί καταφέραμε – μέσα από τα μαθήματα της αμφισβήτησης, της αξιοπρέπειας και της αλληλεγγύης που δώσαμε – να ψηλαφίσουμε την ουτοπία μας.

Περιεχόμενα

Περιεχόμενα.....σελ.	11
Κεφάλαιο 1 : Εισαγωγή.....σελ.	13
Κεφάλαιο 2 : Βασικές Μαθηματικές Έννοιες.....σελ.	17
Κεφάλαιο 3 : Το Πρόβλημα Βελτιστοποίησης.....σελ.	23
Κεφάλαιο 4 : Ο Αλγόριθμος Sequential Quadratic Programming.....σελ.	33
Κεφάλαιο 5 : Αριθμητικά Πειράματα με τον Αλγόριθμο SQP.....σελ.	45
Συμπεράσματα.....σελ.	95
Παράρτημα : Κώδικας MATLAB και Οδηγίες.....σελ.	97
Βιβλιογραφία – Αναφορές.....σελ.	109

Κεφάλαιο 1

Εισαγωγή

Πολλά πρακτικά προβλήματα σχεδίασης ή λήψης απόφασης μπορούν να εκφραστούν σαν πρόβλημα βελτιστοποίησης. Προβλήματα βελτιστοποίησης απαντώνται σε πολλά επιστημονικά πεδία όπως στη φυσική, στη χημεία, στη βιολογία, στην οικονομία, αλλά και σε πρακτικούς τομείς όπως στην παραγωγή και τον σχεδιασμό διαδικασιών, στο σχεδιασμό αεροσκαφών, στις πετρελαϊκές βιομηχανίες κ.τ.λ. Ο χώρος της βελτιστοποίησης στα εφαρμοσμένα μαθηματικά αναφέρεται στην αναζήτηση βέλτιστων παραμέτρων ενός – συνήθως περίπλοκου – συστήματος. Η βελτιστοποίηση είναι πολλές φορές γνωστή ως Μαθηματικός Προγραμματισμός.

Ιστορικά πρώτοι ο Fermat και ο Lagrange βρήκαν φόρμουλες βασισμένες σε αναλυτικές μεθόδους για να λύνουν προβλήματα βελτιστοποίησης. Έπειτα ο Gauss και ο Newton πρότειναν επαναληπτικές διαδικασίες για κίνηση προς την κατεύθυνση της βέλτιστης λύσης. Το πρώτο πεδίο βελτιστοποίησης που βρέθηκε ήταν ο γραμμικός προγραμματισμός από τον G.B. Dantzig, αν και μεγάλο κομμάτι της θεωρίας διατυπώθηκε από τον L. Kantorovich το 1939. Ο Dantzig δημοσίευσε τον αλγόριθμο Simplex το 1947 και ο J. Neumann, την ίδια χρονιά, ανέπτυξε τη θεωρία duality. Μεγάλη ήταν η συμβολή των H. Kuhn και A. Tucker, οι οποίοι το 1951 διατύπωσαν τις συνθήκες βελτίστου για προβλήματα μη γραμμικού προγραμματισμού.

Ας υποθέσουμε ότι οι μεταβλητές x_1, x_2, \dots, x_n και οι μεταξύ τους σχέσεις περιγράφουν ένα φαινόμενο ή διαδικασία ή σύστημα. Το πρόβλημα σχεδίασης ή απόφασης συνίσταται στο να διαλέξουμε τιμές για τις μεταβλητές αυτές. Αν μπορούσαμε να εκφράσουμε ποσοτικά (μέσω μια βαθμωτής συνάρτησης f) την

σχετική αξία (ποιότητα) της κάθε απόφασης, τότε η ελαχιστοποίηση (ή μεγιστοποίηση) του κριτηρίου ποιότητας f θα δώσει την “καλύτερη” λύση στο πρόβλημα σχεδίασης ή απόφασης. Οι παράγοντες που ενδεχομένως περιορίζουν τις τιμές που μπορούν να πάρουν οι μεταβλητές x_1, x_2, \dots, x_n λαμβάνονται υπόψη σαν περιορισμοί κατά την βελτιστοποίηση του κριτηρίου ποιότητας f .

Τα προβλήματα βελτιστοποίησης επομένως, είναι μαθηματικά μοντέλα τα οποία περιγράφουν με κάποιον τρόπο την πραγματικότητα. Έτσι το φυσικό πρόβλημα μετατοπίζεται σε μαθηματικές εξισώσεις οι οποίες και λύνονται με χρήση μαθηματικών μεθόδων που είναι γνωστές ως τεχνικές βελτιστοποίησης. Βεβαίως υπάρχουν πρακτικά προβλήματα όπου δεν είναι δυνατό να βρεθεί κριτήριο ποιότητας f που να εκφράζει ικανοποιητικά τη σχετική αξία των αποφάσεων ή προβλήματα όπου μπορούν να προταθούν περισσότερα από ένα κριτήρια ποιότητας. Η παρούσα εργασία αφορά προβλήματα στα οποία ένα μοναδικό κριτήριο ποιότητας f (ή αντικειμενική συνάρτηση) προκύπτει αβίαστα από την διατύπωση του προβλήματος.

Η δυσκολία επίλυσης του προβλήματος βελτιστοποίησης εξαρτάται από τον αριθμό των αγνώστων μεταβλητών και τον αριθμό των περιορισμών. Όσο αυξάνουν οι αριθμοί αυτοί, τόσο πιο δύσκολο γίνεται το πρόβλημα και η χρήση ηλεκτρονικού υπολογιστή γίνεται ολοένα και πιο απαραίτητη. Προφανώς όσο αυξάνουν οι δυνατότητες των υπολογιστών τόσο περισσότερα και πιο σύνθετα προβλήματα μπορούν να επιλυθούν. Τα προβλήματα βελτιστοποίησης χωρίζονται σε 3 κατηγορίες ανάλογα με τον συνολικό αριθμό μεταβλητών και περιορισμών.

- Προβλήματα μικρής κλίμακας: έχουν λιγότερες από 5 μεταβλητές και περιορισμούς
- Προβλήματα μέσης κλίμακας: ο συνολικός αριθμός μεταβλητών και περιορισμών είναι από 5 έως 100
- Προβλήματα μεγάλης κλίμακας: έχουν εκατοντάδες ή ακόμα και χιλιάδες μεταβλητές και περιορισμούς

Ένας από τους τρόπους που μπορούν να κατηγοριοποιηθούν τα προβλήματα βελτιστοποίησης, είναι ανάλογα με την μορφή που έχουν σε αυτά το κριτήριο

ποιότητας f και οι περιορισμοί. Μία από τις κατηγορίες προβλημάτων, αρκετά μεγάλη και με πολλές πρακτικές εφαρμογές, είναι το πρόβλημα βελτιστοποίησης του κριτηρίου ποιότητας f με ισοτικούς περιορισμούς, το οποίο και πραγματεύεται η παρούσα εργασία.

Στο δεύτερο κεφάλαιο παρουσιάζονται οι μαθηματικές έννοιες και οι ορισμοί που είναι απαραίτητοι για την θεωρία των τεχνικών βελτιστοποίησης. Επίσης ορίζονται οι περισσότεροι από τους μαθηματικούς συμβολισμούς που θα χρησιμοποιηθούν στα επόμενα κεφάλαια.

Στο τρίτο κεφάλαιο παρουσιάζεται αναλυτικά η γενική μορφή του προβλήματος βελτιστοποίησης με ισοτικούς περιορισμούς που είναι το βασικό πρόβλημα της εργασίας. Καταγράφονται τα θεωρήματα ικανών και αναγκαίων συνθηκών που είναι απαραίτητα για να την αναλυτική λύση του προβλήματος.

Στο τέταρτο κεφάλαιο παρουσιάζεται αναλυτικά ο αλγόριθμος SQP που χρησιμοποιήθηκε για τις ανάγκες της παρούσας εργασίας καθώς και όλα τα στάδια και τα βήματα που χρησιμοποιεί για να λύνει τα προβλήματα.

Στο πέμπτο κεφάλαιο παρουσιάζεται η λύση πολλών προβλημάτων βελτιστοποίησης με τον αλγόριθμο SQP σε πίνακες και γραφικές παραστάσεις καθώς και σύγκριση του με άλλους αλγορίθμους.

Στο Παράρτημα παρουσιάζεται ο κώδικας των προγραμμάτων που κατασκευάστηκαν και υλοποιούν του αλγορίθμους που αναλύθηκαν στα προηγούμενα κεφάλαια, για την επίλυση του προβλήματος βελτιστοποίησης με ισοτικούς περιορισμούς. Τα προγράμματα είναι γραμμένα σε κώδικα του MATLAB και μπορεί να τρέξουν μόνο στο περιβάλλον του. Επίσης παρατίθεται ένας σύντομος οδηγός για το περιβάλλον του MATLAB για να το πώς μπορεί κανείς να γράψει και να τρέξει τα παραπάνω προγράμματα με ακρίβεια.

Κεφάλαιο 2

Βασικές Μαθηματικές Έννοιες

Πριν προχωρήσουμε στη διατύπωση του προβλήματος βελτιστοποίησης θα ήταν χρήσιμο και βολικό να αναφερθούν κάποιες βασικές μαθηματικές έννοιες και ορισμοί καθώς και τα σύμβολα και οι συμβάσεις που θα χρησιμοποιηθούν στην παρούσα διπλωματική εργασία.

Στη συνέχεια θα συμβολίζεται με x ένα διάνυσμα διαστάσεων $n \times 1$

$$x = [x_1, x_2, \dots, x_n]^T$$

όπου

$$x_i \in \mathbb{R} \text{ με } i = 1, 2, \dots, n$$

Ο Ευκλείδειος χώρος όλων των n -διανυσμάτων είναι ο \mathbb{R}^n . Δηλαδή με βάση το παραπάνω ορισμό του διανύσματος

$$x \in \mathbb{R}^n$$

Ορισμός 2.1 Τα διανύσματα a_1, a_2, \dots, a_k θα λέγονται γραμμικώς ανεξάρτητα αν ισχύει η συνεπαγωγή

$$\sum_{i=1}^k \lambda_i a_i = 0 \Rightarrow \lambda_i = 0 \text{ με } \lambda_i \in \mathbb{R} \text{ και } i = 1, 2, \dots, k$$

Εναλλακτικά, τα διανύσματα a_1, a_2, \dots, a_k θα λέγονται γραμμικώς ανεξάρτητα αν ισχύει

$$\text{rank} [a_1 \ : \ a_2 \ : \ \dots \ : \ a_k] = k$$

Στη συνέχεια της εργασίας θα χρησιμοποιηθούν 2 πολύ βασικές νόρμες διανυσμάτων οι οποίες ορίζονται εδώ. Ευκλείδεια Νόρμα (ή Νόρμα-2) ενός διανύσματος ορίζεται ως

$$\| x \|_2 = \left[\sum_{i=1}^n x_i^2 \right]^{\frac{1}{2}}$$

Νόρμα-1 ενός διανύσματος ορίζεται ως

$$\| x \|_1 = \sum_{i=1}^n | x_i |$$

Για τις ανάγκες της διπλωματικής εργασίας θα χρειαστούν οι παρακάτω βασικοί ορισμοί που αφορούν τους πίνακες, γνωστοί από το πεδίο της Γραμμικής Άλγεβρας.

Ορισμός 2.2 Μια τετραγωνική μήτρα A , διαστάσεων $n \times n$, λέγεται συμμετρική αν ισχύει η ισότητα

$$A = A^T$$

Μια συμμετρική μήτρα έχει όλες τις ιδιοτιμές τις πραγματικές.

Ορισμός 2.3 Μια συνάρτηση

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j = x^T A x$$

λέγεται τετραγωνική μορφή.

Ορισμός 2.4 Αν $x^T A x > 0$ για κάθε $x \in \mathbb{R}^n$ με $x \neq 0$ τότε η τετραγωνική μορφή λέγεται θετικά ορισμένη.

Αν $x^T A x \geq 0$ για κάθε $x \in \mathbb{R}^n$ και υπάρχει $x \neq 0$ ώστε να είναι $x^T A x = 0$ τότε η τετραγωνική μορφή λέγεται θετικά ημιορισμένη.

Θεώρημα 2.5 Έστω μια συμμετρική μήτρα A , διαστάσεων $n \times n$.

Η A είναι θετικά ορισμένη αν και μόνο αν όλες οι ιδιοτιμές της είναι αυστηρά μεγαλύτερες του μηδενός.

Η A είναι θετικά ημιορισμένη αν και μόνο αν όλες οι ιδιοτιμές της είναι μεγαλύτερες του μηδενός και υπάρχει τουλάχιστον μία ιδιοτιμή ίση με το μηδέν.

Αν η A δεν είναι συμμετρική αλλά είναι τετραγωνική, διαστάσεων $n \times n$, τα παραπάνω ισχύουν για τη μήτρα

$$\bar{A} = \frac{A + A^T}{2}$$

Ορισμός 2.6 Μια τετραγωνική μήτρα Q , διαστάσεων $n \times n$, λέγεται ορθογώνια αν έχει τις γραμμές και τις στήλες της ορθογώνια και μοναδιαία διανύσματα. Εναλλακτικά αν ισχύει

$$Q^T = Q^{-1}$$

το οποίο συνεπάγεται

$$Q^T Q = Q Q^T = I$$

Στη συνέχεια παρουσιάζονται κάποιες βασικές έννοιες και ορισμοί που είναι γνωστοί από την ανάλυση.

Ορισμός 2.7 Θεωρούμε μια πραγματική βαθμωτή συνάρτηση f με n μεταβλητές ως εξής

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$$

Θα λέμε ότι η f είναι συνεχής στο σημείο \mathbf{x} αν δεδομένου ενός $\varepsilon > 0$

$$\exists \delta > 0 : \| f(\mathbf{y}) - f(\mathbf{x}) \|_2 < \varepsilon$$

$$\forall \mathbf{y} : \| \mathbf{y} - \mathbf{x} \|_2 < \delta$$

Στη συνέχεια θεωρούμε μια πραγματική διανυσματική συνάρτηση h n μεταβλητών, η οποία ουσιαστικά είναι ένας πίνακας διαστάσεων $m \times 1$, ως εξής

$$h : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad h(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_m(\mathbf{x})]^T$$

θα λέμε ότι η h είναι συνεχής στο σημείο \mathbf{x} , αν κάθε μια από τις συναρτήσεις

$$h_i : \mathbb{R}^n \rightarrow \mathbb{R} \text{ με } i = 1, 2, \dots, m$$

είναι συνεχής στο σημείο \mathbf{x} .

Στο εξής, όταν αναφέρεται μια συνάρτηση f θα εννοείται πραγματική βαθμωτή συνάρτηση n μεταβλητών και όταν αναφέρεται μια συνάρτηση h θα εννοείται πραγματική διανυσματική συνάρτηση n μεταβλητών, όπως αυτές ορίστηκαν παραπάνω.

Κάθε συνεχής συνάρτηση f θα συμβολίζεται με

$$f \in C$$

Έστω μια συνεχής συνάρτηση f για την οποία υπάρχουν οι μερικές παράγωγοι τάξης k και είναι συνεχείς συναρτήσεις. Αυτό θα συμβολίζεται με

$$f \in C^k$$

Τώρα, είναι πολύ σημαντικό να οριστεί η έννοια της ισοϋψής καμπύλης μια συνάρτησης f , καθώς θα παρουσιαστούν σε επόμενο κεφάλαιο πολλά σχήματα που θα αναπαρίστανται γραφικά ισοϋψείς καμπύλες.

Ορισμός 2.8 Έστω μια συνάρτηση δύο μεταβλητών $f(x) = f(x_1, x_2)$ και μια σταθερά $c \in \mathbb{R}$ που ανήκει στο πεδίο τιμών της f . Το σύνολο

$$I_c = \{ (x_1, x_2) \in \mathbb{R}^2 : f(x_1, x_2) = c \}$$

λέγεται ισοϋψής καμπύλη της συνάρτησης f με τιμή c .

Ορισμός 2.9 Αν $f \in C^1$ τότε η κλίση της συνάρτησης f στο x ορίζεται ως εξής

$$\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right]^T$$

Η κλίση είναι ένας πίνακας διαστάσεων $n \times 1$, κάθετο στην ισοϋψή της συνάρτησης f με φορά προς τα εκεί που αυξάνεται η συνάρτηση.

Ορισμός 2.10 Αν $f \in C^2$ τότε η μήτρα Hessian της f στο x ορίζεται ως εξής

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}$$

Η μήτρα δευτέρων παραγώγων έχει διαστάσεις $n \times n$ και είναι συμμετρική διότι

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j} = \frac{\partial^2 f(x)}{\partial x_j \partial x_i} \quad \forall i, j = 1, \dots, n \text{ με } i \neq j$$

Ορισμός 2.11 Αν $h \in C^1$ τότε η Ιακωβιανή (Jacobian) μήτρα της συνάρτησης h στο x ορίζεται ως εξής

$$\nabla h(x) = [\nabla h_1(x) : \nabla h_2(x) : \dots : \nabla h_m(x)]$$

όπου

$$\nabla h_i(x) = [\frac{\partial h_i(x)}{\partial x_1}, \frac{\partial h_i(x)}{\partial x_2}, \dots, \frac{\partial h_i(x)}{\partial x_n}]^T \text{ με } i = 1, 2, \dots, m$$

Η Ιακωβιανή μήτρα έχει διαστάσεις $n \times m$.

Τέλος παρατίθεται ο ορισμός του ελαχίστου μιας συνάρτησης f πάνω σε ένα σύνολο F , που είναι από τις βασικότερες έννοιες στην θεωρία βελτιστοποίησης.

Ορισμός 2.12 Ένα σημείο $\hat{x} \in F$ λέγεται τοπικό ελάχιστο της f πάνω στο σύνολο F , αν υπάρχει $\varepsilon > 0$ τέτοιο ώστε

$$f(x) \geq f(\hat{x})$$

$$\forall x : \|x - \hat{x}\|_2 \leq \varepsilon$$

Αν η ανισότητα

$$f(x) > f(\hat{x})$$

ικανοποιείται για κάθε

$$x \in F, x \neq \hat{x} \text{ και } \|x - \hat{x}\|_2 \leq \varepsilon$$

τότε το \hat{x} λέγεται αυστηρό τοπικό ελάχιστο της f .

Ορισμός 2.13 Ένα σημείο $\hat{x} \in F$ λέγεται γενικό ελάχιστο της f πάνω στο σύνολο F αν

$$f(x) \geq f(\hat{x}), \forall x \in F$$

Κεφάλαιο 3

Το Πρόβλημα Βελτιστοποίησης

Μετά την αναφορά σε βασικές μαθηματικές έννοιες, μπορεί να οριστεί το γενικό πρόβλημα βελτιστοποίησης, το οποίο στη γενική του μορφή εκφράζεται ως εξής

$$\min_x \{ f(x), x \in F \}$$

Το διάνυσμα x αποτελεί τον άγνωστο του προβλήματος, η τιμή του οποίου αναζητείται ώστε η συνάρτηση f να παρουσιάζει ελάχιστη τιμή πάνω στο σύνολο F .

Η συνάρτηση f είναι μια πραγματική βαθμωτή συνάρτηση n μεταβλητών. Ονομάζεται κριτήριο ποιότητας ή αντικειμενική συνάρτηση κόστους ή ενεργειακή συνάρτηση σε μερικά επιστημονικά πεδία. Συνήθως υποθέτεται ότι είναι συνεχώς διαφορίσιμη ενώ είναι πάντοτε συνεχής.

Το σύνολο F είναι ένα υποσύνολο του χώρου \mathbb{R}^n , ονομάζεται επιτρεπτό σύνολο τιμών (feasible points set) του διανύσματος x και ορίζεται από ένα πλήθος ισοτικών και ανισοτικών περιορισμών. Είναι αυτό που καθορίζει το είδος του ως προς επίλυση προβλήματος βελτιστοποίησης. Στη γενική του μορφή το σύνολο F γράφεται ως εξής

$$F = \{ x \in \mathbb{R}^n : h_i(x) = 0, g_j(x) \leq 0 \text{ με } i = 1, 2, \dots, m \text{ και } j = 1, 2, \dots, l \}$$

Οι συναρτήσεις h_i, g_j αναπαριστούν τους ισοτικούς και ανισοτικούς περιορισμούς αντίστοιχα.

Ανάλογα με το είδος των συναρτήσεων f , g_j , h_i και του επιτρεπτού συνόλου F διακρίνονται οι εξής κατηγορίες προβλημάτων βελτιστοποίησης

- Γραμμικός προγραμματισμός: οι f , g_j , h_i είναι όλες γραμμικές συναρτήσεις
- Τετραγωνικός προγραμματισμός: η f είναι τετραγωνική συνάρτηση και οι g_j , h_i είναι γραμμικές
- Πρόβλημα χωρίς περιορισμούς: η f είναι μη γραμμική συνάρτηση και $F = \mathbb{R}^n$
- Πρόβλημα με περιορισμούς: η f είναι μη γραμμική συνάρτηση και $F \subset \mathbb{R}^n$

Η παρούσα διπλωματική εργασία ασχολείται με την τελευταία κατηγορία προβλημάτων, τα προβλήματα βελτιστοποίησης με ισοτικούς περιορισμούς. Έτσι ορίζεται το πρόβλημα αυτό ως

$$\min_{x \in \mathbb{R}^n} \{ f(x) \} \quad (P1)$$

$$\text{subject to } h(x) = 0$$

Η f είναι μια μη γραμμική πραγματική βαθμωτή συνάρτηση n μεταβλητών

$$f: \mathbb{R}^n \rightarrow \mathbb{R}, f \in C^1, f(x) = f(x_1, x_2, \dots, x_n)$$

Η h είναι μια μη γραμμική πραγματική διανυσματική συνάρτηση n μεταβλητών η οποία αναπαριστά τους ισοτικούς περιορισμούς

$$h: \mathbb{R}^n \rightarrow \mathbb{R}^m, h \in C^1, h(x) = [h_1(x), h_2(x), \dots, h_m(x)]^T$$

$$\text{με } h_i: \mathbb{R}^n \rightarrow \mathbb{R}$$

Ο ακέραιος αριθμός m δηλώνει το πλήθος των ισοτικών περιορισμών που έχει το πρόβλημα. Προκειμένου να εξασφαλιστεί η γραμμική ανεξαρτησία μεταξύ των ισοτικών περιορισμών, απαιτείται το πλήθος τους να είναι μικρότερο από τον αριθμό των μεταβλητών του προβλήματος. Δηλαδή απαιτείται

$$m < n$$

Το πρόβλημα (P1) μπορεί να γραφτεί με την βοήθεια της παρακάτω ισοδυναμίας

$$\begin{aligned}h(x) &= 0 \\ \Leftrightarrow [h_1(x), h_2(x), \dots, h_m(x)]^T &= 0 \\ \Leftrightarrow h_i(x) &= 0 \text{ με } i = 1, 2, \dots, m\end{aligned}$$

ως εξής

$$\begin{aligned}\min_{x \in \mathbb{R}^n} \{ f(x) \} & \qquad \qquad \qquad (P1') \\ \text{subject to } h_i(x) &= 0 \text{ με } i = 1, 2, \dots, m\end{aligned}$$

Στη βιβλιογραφία συναντάτε η διατύπωση του προβλήματος άλλες φορές όπως ορίστηκε το (P1) άλλες φορές όπως το (P1').

Το επιτρεπτό σύνολο του προβλήματος (P1) είναι το

$$F = \{ x \in \mathbb{R}^n : h_i(x) = 0 \text{ με } i = 1, 2, \dots, m \}$$

Δηλαδή κάθε σημείο x που ικανοποιεί τους ισοτικούς περιορισμούς ανήκει στο σύνολο F .

Όταν θέλουμε να επιλύσουμε ένα πρόβλημα της μορφής (P1) με επιτρεπτό σύνολο F , αναζητούμε ένα διάνυσμα \hat{x} το οποίο θα ανήκει στο επιτρεπτό σύνολο του προβλήματος, ενώ ταυτόχρονα θα ελαχιστοποιεί την τιμή της αντικειμενικής συνάρτησης κόστους πάνω σε αυτό. Θεωρητικά βέλτιστη λύση του προβλήματος θεωρείται το γενικό ελάχιστο της αντικειμενικής συνάρτησης κόστους πάνω στο επιτρεπτό σύνολο. Η ύπαρξη όμως γενικού ελαχίστου δεν είναι πάντοτε εξασφαλισμένη αλλά και στις περιπτώσεις όπου είναι, η εύρεσή του είναι μια αρκετά δύσκολη διαδικασία, εξαρτάται από πολλούς παράγοντες και μπορεί να επιτευχθεί σε ορισμένες μόνο περιπτώσεις και κυρίως σε περιπτώσεις όπου το πρόβλημα έχει κάποια ειδική μορφή. Το πλήθος όμως των περιπτώσεων όπου, χάρη της ειδικής μορφής του προβλήματος, είναι εξασφαλισμένη η ύπαρξη γενικού ελαχίστου και εφικτή η εύρεση του, είναι μικρό συγκριτικά με το σύνολο των προβλημάτων που

αντιμετωπίζονται στην πράξη, με αποτέλεσμα να δεχόμαστε σαν ικανοποιητική λύση του προβλήματος βελτιστοποίησης (P1) με επιτρεπτό σύνολο F κάθε (αυστηρό ή όχι) τοπικό ελάχιστο της αντικειμενικής συνάρτησης f, που μπορούμε να βρεθεί, πάνω στο επιτρεπτό σύνολο F.

Σε πολλά πρακτικά προβλήματα συναντώνται προβλήματα στα οποία αναζητείται ένα διάνυσμα x το οποίο μεγιστοποιεί την τιμή της αντικειμενικής συνάρτησης κόστους. Αυτό μπορεί πολύ εύκολα να το αναχθεί σε πρόβλημα της μορφής (P1) με την βοήθεια της ισότητας

$$\max_x \{ f(x), x \in F \} = -\min_x \{ -f(x), x \in F \}$$

Όπως αναφέρθηκε και προηγουμένως η εργασία ασχολείται με προβλήματα που έχουν μόνο ισοτικούς περιορισμούς όπως το πρόβλημα (P1). Υπάρχει ωστόσο ένας πολύ εύκολος τρόπος να αναχθούν τα προβλήματα με ισοτικούς και ανισοτικούς περιορισμούς στην μορφή του προβλήματος (P1) (Μαράτος, αναφορά [3]).

Έστω ένα πρόβλημα βελτιστοποίησης με ισοτικούς και ανισοτικούς περιορισμούς της μορφής

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \{ f(x) \} & \quad (P2) \\ \text{subject to} \quad h_i(x) = 0 & \quad \mu\epsilon \quad i = 1, \dots, m \\ g_j(x) \leq 0 & \quad \mu\epsilon \quad j = 1, 2, \dots, l \end{aligned}$$

Οι συναρτήσεις f και h_i κατά τα γνωστά ορίστηκαν προηγουμένως. Η g είναι μια μη γραμμική πραγματική διανυσματική συνάρτηση n μεταβλητών

$$\begin{aligned} g : \mathbb{R}^n \rightarrow \mathbb{R}^l, g \in C^1, g(x) = [g_1(x), g_2(x), \dots, g_m(x)]^T \\ \mu\epsilon \quad g_i : \mathbb{R}^n \rightarrow \mathbb{R} \end{aligned}$$

Το πρόβλημα (P2) ανάγεται σε ισοδύναμο πρόβλημα με μόνο ισοτικούς περιορισμούς και παίρνει την παρακάτω μορφή

$$\begin{aligned} \min_{x \in \mathbb{R}^{n+1}} \{ f(x) \} & \quad (P2') \\ \text{subject to} \quad h_i(x) = 0 & \text{ με } i = 1, \dots, m \\ g_j(x) + x_{n+j}^2 = 0 & \text{ με } j = 1, \dots, l \end{aligned}$$

Το πρόβλημα (P2') είναι ουσιαστικά ένα πρόβλημα της μορφής (P1) αλλά έχει πλέον $n+1$ μεταβλητές και $m+1$ ισοτικούς περιορισμούς. Δηλαδή να μεν απαλείφθηκαν οι ανισοτικοί περιορισμοί, αλλά ταυτόχρονα αυξήθηκαν οι άγνωστες μεταβλητές του προβλήματος και οι ισοτικοί περιορισμοί πράγμα που συνεπάγεται μεγαλύτερη δυσκολία στην επίλυση του προβλήματος αφού αυξάνεται η πολυπλοκότητα των πράξεων.

Αφού καταγράφηκε το πρόβλημα βελτιστοποίησης (P1) τώρα μπορεί να δειχθεί πως μπορεί κανείς να επιλύσει αναλυτικά το πρόβλημα και να βρει τη βέλτιστη λύση \hat{x} . Θα χρειαστούν οι ακόλουθοι ορισμοί.

Ορισμός 3.1 Συνάρτηση Lagrange για το πρόβλημα (P1) ορίζεται η μη γραμμική πραγματική βαθμωτή συνάρτηση $n+m$ μεταβλητών

$$L : \mathbb{R}^{n+m} \rightarrow \mathbb{R}, \quad L(x, \lambda) = f(x) + \sum_{i=1}^m \lambda_i h_i(x) = f(x) + \lambda^T h(x)$$

$$\text{με } \lambda \in \mathbb{R}^m, \quad \lambda = [\lambda_1, \lambda_2, \dots, \lambda_m]^T$$

$$\text{όπου } \lambda_i \in \mathbb{R}$$

Οι πραγματικοί αριθμοί λ_i ονομάζονται πολλαπλασιαστές Lagrange του προβλήματος.

Η κλίση της συνάρτησης Lagrange δίνεται από τον τύπο

$$\nabla L(x, \lambda) = \begin{bmatrix} \nabla_x L(x, \lambda) \\ \nabla_\lambda L(x, \lambda) \end{bmatrix}$$

Όπου

$$\nabla_x L(x, \lambda) = \nabla f(x) + \sum_{i=1}^m \lambda_i \nabla h_i(x) = \nabla f(x) + \nabla h(x)\lambda$$

και

$$\nabla_\lambda L(x, \lambda) = h(x)$$

Η μήτρα δευτέρων παραγώγων της συνάρτησης Lagrange ορίζεται ως

$$\nabla_{xx}^2 L(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x)$$

Ορισμός 3.2 Ένα σημείο $\hat{x} \in \mathbb{R}^n$ λέγεται κανονικό σημείο των περιορισμών αν οι κλίσεις των ισοτικών περιορισμών είναι γραμμικά ανεξάρτητες στο σημείο αυτό. Δηλαδή αν τα διανύσματα

$$\nabla h_1(\hat{x}), \nabla h_2(\hat{x}), \dots, \nabla h_m(\hat{x})$$

είναι γραμμικά ανεξάρτητα ή εναλλακτικά αν

$$\text{rank} [\nabla h_1(\hat{x}) : \nabla h_2(\hat{x}) : \dots : \nabla h_m(\hat{x})] = m$$

$$\Leftrightarrow \text{rank} [\nabla h(\hat{x})] = m$$

Στο χώρο των n διαστάσεων κάθε ένας από τους περιορισμούς $h_i(x)$ ορίζει μια υπερεπιφάνεια διάστασης $n - 1$. Η υπερεπιφάνεια που ορίζουν όλοι αυτοί οι περιορισμοί γράφεται ως

$$H = \{ x \in \mathbb{R}^n : h(x) = 0 \}$$

Πρόταση 3.3 Αν το σημείο $\hat{x} \in \mathbb{R}^n$ είναι κανονικό σημείο της υπερεπιφάνειας H , τότε το εφαπτόμενο επίπεδο της H στο \hat{x} είναι

$$T = \{ y \in \mathbb{R}^n : \nabla h(\hat{x})^T y = 0 \}$$

Με βάση τον ορισμό 2.11 της Ιακωβιανής ισχύει

$$\nabla h(x)^T y = 0 \Leftrightarrow \nabla h_i(x)^T y = 0 \text{ με } i = 1, 2, \dots, m$$

Τώρα μπορούν να παρουσιαστούν οι συνθήκες βελτίστου που πρέπει να ικανοποιεί το διάνυσμα x για να αποτελεί λύση του προβλήματος (P1).

Θεώρημα 3.4 (Αναγκαίες συνθήκες 1^{ης} τάξης)

Θεωρούμε ένα πρόβλημα βελτιστοποίησης της μορφής (P1). Αν ένα σημείο $\hat{x} \in \mathbb{R}^n$ είναι τοπικό ακρότατο του προβλήματος και κανονικό σημείο των περιορισμών, τότε υπάρχει $\hat{\lambda} \in \mathbb{R}^m$ τέτοιο ώστε

$$\nabla L(\hat{x}, \hat{\lambda}) = 0 \Leftrightarrow \begin{bmatrix} \nabla_x L(\hat{x}, \hat{\lambda}) \\ \nabla_\lambda L(\hat{x}, \hat{\lambda}) \end{bmatrix} = 0 \Leftrightarrow \begin{bmatrix} \nabla_x L(\hat{x}, \hat{\lambda}) = 0 \\ \nabla_\lambda L(\hat{x}, \hat{\lambda}) = 0 \end{bmatrix}$$

$$\Leftrightarrow \boxed{\begin{array}{l} \nabla f(\hat{x}) + \nabla h(\hat{x}) \hat{\lambda} = 0 \\ h(\hat{x}) = 0 \end{array}}$$

Θεώρημα 3.5 (Αναγκαίες συνθήκες 2^{ης} τάξης)

Θεωρούμε ένα πρόβλημα βελτιστοποίησης της μορφής (P1). Αν ένα σημείο $\hat{x} \in \mathbb{R}^n$ είναι τοπικό ελάχιστο του προβλήματος και κανονικό σημείο των περιορισμών, τότε υπάρχει $\hat{\lambda} \in \mathbb{R}^m$ τέτοιο ώστε

- $\nabla f(\hat{x}) + \nabla h(\hat{x}) \hat{\lambda} = 0$
- $h(\hat{x}) = 0$
- $y^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) y \geq 0$ για κάθε $y \in T$

Θεώρημα 3.6 (Ικανές συνθήκες)

Αν τα σημεία $\hat{x} \in \mathbb{R}^n$ και $\hat{\lambda} \in \mathbb{R}^m$ ικανοποιούν τις σχέσεις

- $\nabla f(\hat{x}) + \nabla h(\hat{x}) \hat{\lambda} = 0$
- $h(\hat{x}) = 0$
- $y^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) y > 0$ για κάθε $y \in T$ με $y \neq 0$, δηλαδή η μήτρα $\nabla_{xx}^2 L(\hat{x}, \hat{\lambda})$ είναι θετικά ορισμένη πάνω στο υπερεπίπεδο T

τότε το \hat{x} αποτελεί αυστηρό τοπικό ελάχιστο της f πάνω στην υπερεπιφάνεια H .

Η τρίτη σχέση των ικανών συνθηκών μπορεί να γραφεί με έναν απλούστερο τρόπο, ο οποίος είναι ευκολότερα υλοποιήσιμος με το λογισμικό του MATLAB.

Η μήτρα $\nabla h(x)$ μπορεί να γραφτεί με βάση την παραγοντοποίηση QR ως

$$\nabla h(x) = Q R = [Q_1 : Q_2] \begin{bmatrix} \hat{R} \\ \vdots \\ 0 \end{bmatrix} = Q_1 \hat{R}$$

με τις μήτρες να έχουν διαστάσεις

$$Q : n \times n, R : n \times m, Q_1 : n \times m, Q_2 : n \times (n - m), \hat{R} : m \times m$$

Η μήτρα Q είναι ορθογώνια ($Q^T Q = Q Q^T = I$), η μήτρα \hat{R} είναι άνω τριγωνική και η μήτρα Q_2 είναι πλήρους βαθμού και εκτείνει τον χώρο των ορθογωνίων προς την $\nabla h(x)^T$ διανυσμάτων. Πράγματι αν

$$y = Q_2 w, \text{ με } y \in \mathbb{R}^n \text{ και } w \in \mathbb{R}^{n-m}$$

τότε

$$\nabla h(x)^T y = \nabla h(x)^T Q_2 w = \hat{R}^T Q_1^T Q_2 w = 0$$

διότι

$$Q^T Q = I \Leftrightarrow \begin{bmatrix} Q_1^T \\ \vdots \\ Q_2^T \end{bmatrix} [Q_1 \ : \ Q_2] = I \Leftrightarrow \begin{bmatrix} Q_1^T Q_1 & Q_1^T Q_2 \\ Q_2^T Q_1 & Q_2^T Q_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

επομένως

$$Q_1^T Q_2 = 0$$

Άρα λοιπόν με τη σχέση $y = Q_2 w$, η τρίτη σχέση των αναγκαίων συνθηκών γράφεται

$$y^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) y = w^T \{ Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 \} w$$

Για να αποτελεί το \hat{x} αυστηρό τοπικό ελάχιστο αρκεί να ισχύουν οι δυο πρώτες σχέσεις των ικανών συνθηκών και

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 > 0 \quad (3.7)$$

Όπως αναφέρθηκε και προηγουμένως δεχόμαστε ως ικανοποιητική λύση του προβλήματος (P1) κάθε τοπικό ελάχιστο που μπορεί να βρεθεί. Στην πράξη ορίζονται σαν επιθυμητά σημεία του προβλήματος (P1) τα σημεία που ικανοποιούν τις αναγκαίες συνθήκες 1^{ης} τάξης. Το σύνολο των επιθυμητών σημείων (desirable points set) είναι

$$D = F \cap D' \cap D''$$

με

$$D' = \{ x \in \mathbb{R}^n : \nabla f(x) + \nabla h(x) \lambda = 0, \text{ για κάποια } \lambda \in \mathbb{R}^m \}$$

και

$$D'' = \{ x \in \mathbb{R}^n : \text{rank}[\nabla h(x)] = m \}$$

άρα

$$D = \{ x \in \mathbb{R}^n : h(x) = 0, \text{rank}[\nabla h(x)] = m, \\ \nabla f(x) + \nabla h(x) \lambda = 0, \text{ για κάποια } \lambda \in \mathbb{R}^m \}$$

Το σύνολο D περιέχει, εκτός από τοπικά ελάχιστα, και τοπικά μέγιστα και σαγματικά σημεία του προβλήματος (P1). Με τη χρήση όμως της σχέσης (3.7) μπορούν εύκολα να υπολογιστούν τα αυστηρά τοπικά ελάχιστα του προβλήματος.

Για να λύσει δηλαδή κανείς αναλυτικά το πρόβλημα (P1) πρέπει να κάνει χρήση των θεωρημάτων για τις συνθήκες βελτίστου που χαρακτηρίζουν την λύση του προβλήματος. Γενικά οι εξισώσεις των θεωρημάτων αποτελούν ένα σύστημα $n+m$ μη γραμμικών εξισώσεων με $n+m$ αγνώστους τα διανύσματα \hat{x} και $\hat{\lambda}$ διαστάσεων $n \times 1$ και $m \times 1$ αντίστοιχα. Όμως, ένας τέτοιος τρόπος δεν είναι αποδοτικός και μπορεί να επιτευχθεί για μικρά συνήθως προβλήματα (προβλήματα με μικρή διάσταση του διανύσματος x) τα οποία επιπλέον έχουν και μικρό συνολικό πλήθος περιορισμών. Για την πλειοψηφία όμως των προβλημάτων βελτιστοποίησης, ο κατευθείαν υπολογισμός της λύσης είναι πρακτικά αδύνατος, για αυτό χρησιμοποιούνται επαναληπτικές διαδικασίες που θα αναφερθούν στο επόμενο κεφάλαιο.

Κεφάλαιο 4

Ο Αλγόριθμος Sequential Quadratic Programming

Ο μόνος πρακτικός τρόπος επίλυσης προβλημάτων βελτιστοποίησης της μορφής (P1), που είναι δύσκολο έως αδύνατο να λυθούν αναλυτικά, στηρίζεται στη χρήση αλγορίθμων. Η επίλυση του προβλήματος (P1) με επαναληπτική διαδικασία υλοποιείται με αλγορίθμους βελτιστοποίησης. Ένας αλγόριθμος επίλυσης προβλημάτων βελτιστοποίησης, είναι μια αριθμητική μέθοδος επαναληπτικού χαρακτήρα, η οποία από ένα δοσμένο σημείο x^0 κατασκευάζει μια ακολουθία σημείων

$$x^0, x^1, \dots, x^{k-1}, x^k, x^{k+1}, \dots$$

Ουσιαστικά ο αλγόριθμος αποτελεί μία απεικόνιση που δίνει στο επόμενο σημείο x^{k+1} της ακολουθίας, από το προηγούμενο x^k . Οι βασικές ιδιότητες ενός αλγορίθμου επίλυσης προβλημάτων βελτιστοποίησης είναι

- σύγκλιση: με ορισμένες προϋποθέσεις ισχύει

$$\lim_{k \rightarrow \infty} x^k = \hat{x}$$

Αυτή η ιδιότητα εξασφαλίζει ότι το σημείο που προσεγγίζεται από την ακολουθία $x^0, x^1, \dots, x^{k-1}, x^k$ που κατασκευάζει ο αλγόριθμος, είναι η λύση \hat{x} του προβλήματος.

Αν για κάθε αρχικό σημείο η αντίστοιχη ακολουθία που κατασκευάζεται από τον αλγόριθμο συγκλίνει σε κάποιο επιθυμητό σημείο τότε ο αλγόριθμος έχει την ιδιότητα της γενικής σύγκλισης.

- Ταχύτητα σύγκλισης: εκφράζει το πόσο γρήγορα η ακολουθία $x^0, x^1, \dots, x^{k-1}, x^k$ που δημιουργείται από τον αλγόριθμο, προσεγγίζει την επιθυμητή τιμή.

Κάθε αλγόριθμος βελτιστοποίησης απαρτίζεται από 2 βασικά στοιχεία. Την κατεύθυνση έρευνας $d(x^k)$, η οποία αποτελεί την κατεύθυνση πάνω στην οποία κινούμενοι από το αρχικό σημείο x^k αναζητούμε το σημείο x^{k+1} και το μήκος βήματος σ^k , το οποίο είναι η απόσταση του σημείου x^{k+1} από το σημείο x^k κατά την κατεύθυνση $d(x^k)$. Είναι δηλαδή

$$x^{k+1} = x^k + \sigma^k d(x^k)$$

Οι αλγόριθμοι βελτιστοποίησης είναι επαναληπτικές μέθοδοι, με την έννοια ότι ξεκινούν από ένα αρχικό σημείο x^0 και κατασκευάζουν μια ακολουθία x^0, \dots, x^k και είναι επίσης μέθοδοι καθόδου. Αυτό σημαίνει ότι κάθε νέο σημείο x^{k+1} που κατασκευάζει ο αλγόριθμος μειώνει την τιμή $z(x^k)$ μια συνεχούς συνάρτησης z που χαρακτηρίζει το προς επίλυση πρόβλημα, σχετικά με το προηγούμενο σημείο. Δηλαδή

$$z(x^{k+1}) < z(x^k)$$

Η ιδιότητα αυτή της καθόδου παίζει σημαντικό ρόλο για τη σύγκλιση του αλγορίθμου. Είναι όμως αναγκαία συνθήκη και όχι ικανή.

Όπως αναφέρθηκε στο προηγούμενο κεφάλαιο, επιθυμητά σημεία για την λύση του προβλήματος (P1) ορίζονται τα σημεία που ικανοποιούν τις αναγκαίες συνθήκες 1^{ης} τάξης. Δηλαδή κάθε σημείο x που ανήκει στο σύνολο επιθυμητών σημείων D . Θεωρείται ότι ένας αλγόριθμος έλυσε το πρόβλημα (P1) αν κατασκεύασε μια ακολουθία x^0, \dots, x^k που συγκλίνει σε κάποιο επιθυμητό σημείο $x \in D$.

Παρόλο που το σύνολο D περιλαμβάνει και άλλα σημεία εκτός από τοπικά ελάχιστα, λόγω της σημαντικής ιδιότητας της καθόδου σπάνια ένας αλγόριθμος συγκλίνει σε σημείο του D που δεν είναι τοπικό ελάχιστο.

Οι διάφοροι αλγόριθμοι διαφέρουν μεταξύ τους κυρίως ως προς τον τρόπο που ορίζουν την κατεύθυνση έρευνας, το μήκος βήματος αλλά και την συνάρτηση καθόδου z . Επίσης διαφέρουν και ανάλογα με την πληροφορία που είναι διαθέσιμη δηλαδή αν είναι διαθέσιμες μόνο οι τιμές των συναρτήσεων ή μαζί και οι πρώτες παράγωγοι των συναρτήσεων αυτών ή μαζί και οι δεύτερες παράγωγοι. Για την επίλυση του προβλήματος (P1), στα πλαίσια της εργασίας, θα χρησιμοποιηθεί ο αλγόριθμος επαναληπτικού τετραγωνικού προγραμματισμού (Sequential Quadratic Programming) ο οποίος για συντομία θα γράφεται SQP.

Ο αλγόριθμος SQP είναι μια επαναληπτική διαδικασία και σε κάθε επανάληψη εκτελείται μια αλληλουχία βημάτων που θα παρουσιαστούν στη συνέχεια. Ο αριθμός της επανάληψης θα συμβολίζεται με τον ακέραιο αριθμό k . Κάθε μεταβλητή με άνω δείκτη k θα δηλώνει την τιμή της μεταβλητής στην k επανάληψη του αλγορίθμου.

Χρησιμοποιεί την τεχνική της έρευνας γραμμής καθώς κάθε επόμενο σημείο υπολογίζεται με βάση το προηγούμενο ως εξής

$$x^{k+1} = x^k + \sigma^k d^k$$

όπου x^k είναι το τρέχον σημείο, σ^k είναι το μήκος βήματος και d^k η κατεύθυνση έρευνας όλα υπολογισμένα στην επανάληψη k . Επειδή πολλοί αλγόριθμοι δεν συγκλίνουν για αρχικά σημεία μακριά από την λύση \hat{x} χρησιμοποιείται η μέθοδος της έρευνας γραμμής για να επιτευχθεί σύγκλιση στην επιθυμητή λύση από κάθε αρχικό σημείο x^0 .

Ο αλγόριθμος ορίζει την κατεύθυνση έρευνας d^k ως τη λύση του βοηθητικού υποπροβλήματος τετραγωνικού προγραμματισμού

$$\min_{d^k \in \mathbb{R}^n} \left\{ \nabla f(x^k)^T d^k + \frac{1}{2} (d^k)^T Q^k d^k \right\} \quad (\text{QP})$$

$$\text{subject to } h_i(x^k) + \nabla h_i(x^k)^T d^k = 0, i = 1, \dots, m$$

το οποίο αποτελεί μια τοπική προσέγγιση Taylor του προβλήματος (P1) γύρω από το τρέχον σημείο x^k .

Η πρώτη μορφή της μεθόδου SQP προτάθηκε από τον R.Wilson η οποία χρησιμοποιούσε την Q^k ως τη μήτρα δευτέρων παραγώγων (Hessian) της συνάρτησης Lagrange. Ο R.Fletcher είχε την ιδέα για επίλυση βοηθητικών προγραμμάτων τετραγωνικού προγραμματισμού. Οι S.Han και M.Powell αντικατέστησαν την Hessian της συνάρτησης Lagrange με μια μήτρα που αποτελεί Quasi – Newton προσέγγιση της. Ο αλγόριθμος που θα παρουσιαστεί βασίζεται σε αυτή τη μέθοδο και ανήκει στην κατηγορία των Quasi – Newton μεθόδων αφού δεν απαιτεί τον υπολογισμό της Hessian της συνάρτησης Lagrange. Η μήτρα Q^k έχει διαστάσεις $n \times n$ και πρέπει να είναι συμμετρική και θετικά ορισμένη. Η μήτρα αυτή ενημερώνεται (ή ανανεώνεται) σε κάθε επανάληψη με τη γνωστή μέθοδο των Broyden – Fletcher – Goldfarb – Shanno (γνωστή στη βιβλιογραφία ως BFGS formula). Η μέθοδος αυτή χρησιμοποιείται για να παραμένει η Q^k συμμετρική και θετικά ορισμένη μετά από κάθε ανανέωσή της, σε κάθε μια από τις επαναλήψεις του αλγορίθμου. Στο Βήμα 0 στο ξεκίνημα του αλγορίθμου ορίζεται η Q^0 ως η μοναδιαία μήτρα διαστάσεων $n \times n$ δηλαδή $Q^0 = I_n$. Η ανανέωση της Q^k γίνεται στο Βήμα 5.

Το βοηθητικό πρόβλημα τετραγωνικού προγραμματισμού (QP) μπορεί να λυθεί κάνοντας χρήση του θεωρήματος 3.4 για τις αναγκαίες συνθήκες πρώτης τάξης. Έτσι προκύπτει

$$Q^k d^k + \nabla f(x^k) + \nabla h(x^k) \lambda^k = 0 \quad (4.1)$$

$$\nabla h(x^k)^T d^k + h(x^k) = 0 \quad (4.2)$$

από τις οποίες προκύπτει το σύστημα

$$\begin{bmatrix} Q^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla f(x^k) \\ -h(x^k) \end{bmatrix} \quad (S)$$

Το σύστημα (S) είναι γνωστό από τη βιβλιογραφία ως σύστημα Karush – Kuhn – Tucker (KKT System). Είναι ένα γραμμικό σύστημα $n+m$ εξισώσεων με $n+m$ αγνώστους. Το σύστημα αυτό μπορεί να λυθεί τη βοήθεια του MATLAB αν απλά δοθούν οι πίνακες ως είσοδο στη συνάρτηση `mldivide`. Το MATLAB θα ψάξει την αποδοτικότερη αριθμητική μέθοδο και θα το λύσει. Λύση του συστήματος αυτού είναι η κατεύθυνση έρευνας d^k (διάνυσμα διαστάσεων $n \times 1$) και οι πολλαπλασιαστές Lagrange του προβλήματος λ^k (διάνυσμα διαστάσεων $m \times 1$). Στην επανάληψη k οι πολλαπλασιαστές Lagrange λ^k είναι οι τρέχουσες εκτιμήσεις των πολλαπλασιαστών $\hat{\lambda}$.

Θεώρημα 4.1 Αν η μήτρα Q^k είναι συμμετρική και θετικά ορισμένη και ισχύει

$$\text{rank}[\nabla h(x^k)] = m$$

τότε η μήτρα

$$T = \begin{bmatrix} Q^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix}$$

αντιστρέφεται.

Με την απαίτηση που έχει γίνει προηγουμένως για την Q^k (να είναι συμμετρική θετικά ορισμένη) αλλά και από τις αναγκαίες συνθήκες 1^{ης} τάξης που μιλάνε για κανονικά σημεία περιορισμών (ορισμός 3.2) οι απαιτήσεις του θεωρήματος ικανοποιούνται άρα η μήτρα T αντιστρέφεται πάντα. Το βοηθητικό πρόγραμμα τετραγωνικού προγραμματισμού επιλύεται αριθμητικά σε κάθε επανάληψη στο Βήμα 1 του αλγορίθμου.

Για να οριστεί πως ο αλγόριθμος SQP υπολογίζει το μήκος βήματος σ^k είναι απαραίτητο να αναφερθεί η μέθοδος των συναρτήσεων ποινής.

Η βασική ιδέα των συναρτήσεων ποινής είναι να αντικατασταθεί το πρόβλημα (P1) με ένα ισοδύναμο πρόβλημα χωρίς περιορισμούς της μορφής

$$\min_{x \in \mathbb{R}^n} \{ \gamma(x, c) \} \quad (P3)$$

με

$$\gamma(x, c) = f(x) + c P(x)$$

όπου γ είναι η συνάρτηση ποινής, c είναι η παράμετρος ποινής και $P(x)$ ο όρος ποινής.

Στον αλγόριθμο αυτό χρησιμοποιείται μια ακριβή συνάρτηση ποινής της μορφής

$$\gamma(x, c) = f(x) + c \|h(x)\|_1$$

Η ακριβής συνάρτηση ποινής ορίζει ότι υπάρχει πεπερασμένο $c > 0$ τέτοιο ώστε κάθε τοπικό ελάχιστο του προβλήματος (P3) να είναι τοπικό ελάχιστο του προβλήματος (P1). Όμως η γ είναι μη – διαφορίσιμη συνάρτηση και έτσι δεν είναι εφικτό να λυθεί απευθείας το ισοδύναμο πρόβλημα με χρήση των θεωρημάτων αναγκαίων και ικανών συνθηκών. Ορίζονται όμως για την γ κατευθυνόμενες παράγωγοι.

Η παράγωγος της ακριβούς συνάρτησης ποινής γ στην κατεύθυνση έρευνας d^k ορίζεται ως

$$\Theta(x^k, d^k, c) = \nabla f(x^k)^T d^k - c \|h(x^k)\|_1$$

Ο υπολογισμός αυτός γίνεται στο Βήμα 2 του αλγορίθμου.

Η σταθερά ποινής c αποτελεί πολύ σημαντικό στοιχείο για τον αλγόριθμο. Αποδεικνύεται ότι (αναφορά [4], αναφορά [10]) αν η σταθερά c είναι αρκετά μεγάλη αλλά πεπερασμένη τότε ο αλγόριθμος συγκλίνει για οποιοδήποτε $x^0 \in \mathbb{R}^n$. Η ταχύτητα σύγκλισης (τάξη σύγκλισης) του αλγορίθμου είναι υπεργραμμική (superlinear convergence).

Η $\gamma(x, c)$ αποτελεί συνάρτηση καθόδου για το πρόβλημα(P1). Κάθε νέο σημείο που κατασκευάζει ο αλγόριθμος μειώνει την τιμή της γ σε σχέση με το προηγούμενο σημείο δηλαδή

$$\gamma(x^{k+1}) < \gamma(x^k)$$

Η συνθήκη καθόδου απαιτείται να ισχύει σε κάθε επανάληψη του αλγορίθμου για να έχουμε σύγκλιση σε επιθυμητό σημείο \hat{x} (με $\hat{x} \in D$) το οποίο είναι τοπικό ελάχιστο.

Η συνάρτηση γ μειώνεται από το x^k κατά την κατεύθυνση της d^k αν ισχύει

$$\theta(x^k, d^k, c) < 0$$

Για να εξασφαλίζεται η σημαντική αυτή ιδιότητα της καθόδου, που είναι αναγκαία συνθήκη για τη σύγκλιση του αλγορίθμου, πρέπει η παράμετρος ποινής να είναι θετική και να έχει αρκετά μεγάλη τιμή (πεπερασμένη). Αν η αρχική τιμή που δίνεται στο πρόγραμμα για την παράμετρο c δεν είναι αρκετή για να ισχύει η συνθήκη καθόδου, σε κάποια επανάληψη του αλγορίθμου, τότε η παράμετρος c αυξάνεται κατά μια τιμή μέσα, στην τρέχουσα επανάληψη, ώστε να ισχύει η συνθήκη καθόδου.

Ξεκινώντας από την σχέση (4.1) των αναγκαίων συνθηκών προκύπτει

$$(4.1) \Leftrightarrow \nabla f(x^k) = -Q^k d^k - \nabla h(x^k)\lambda^k$$

$$\Leftrightarrow \nabla f(x^k)^T d^k = - (d^k)^T Q^k d^k - (\lambda^k)^T \nabla h(x^k)^T d^k \quad (4.3)$$

$$(4.2) \Leftrightarrow \nabla h(x^k)^T d^k = -h(x^k) \quad (4.4)$$

$$(4.3), (4.4) \Leftrightarrow \nabla f(x^k)^T d^k = -(d^k)^T Q^k d^k + (\lambda^k)^T h(x^k) \quad (4.5)$$

Με βάση την σχέση (4.5) η παράγωγος της ακριβούς συνάρτησης ποινής γράφεται

$$\Theta(x^k, d^k, c) = \nabla f(x^k)^T d^k - c \|h(x^k)\|_1$$

$$\stackrel{(4.5)}{\Leftrightarrow} \Theta(x^k, d^k, c) = -(d^k)^T Q^k d^k + (\lambda^k)^T h(x^k) - c \|h(x^k)\|_1$$

$$\Leftrightarrow \Theta(x^k, d^k, c) = -(d^k)^T Q^k d^k - \left\{ c \|h(x^k)\|_1 - (\lambda^k)^T h(x^k) \right\} \quad (4.6)$$

Για να εξασφαλίζεται η συνθήκη καθόδου πρέπει η Θ να είναι αρνητική σε κάθε επανάληψη k . Για τον πρώτο όρο της σχέσης (4.6) ισχύει

$$(d^k)^T Q^k d^k > 0$$

αφού η Q^k είναι συμμετρική και θετικά ορισμένη. Για τον όρο που είναι μέσα στην αγκύλη χρησιμοποιούμε έναν έλεγχο (test) στο Βήμα 1' του αλγορίθμου. Αν ο όρος αυτός είναι αρνητικός τότε υπάρχει τροποποίηση της παραμέτρου c , αλλιώς παραμένει ως έχει.

Για το μήκος βήματος πρέπει να ισχύει πάντα $\sigma^k > 0$ και ο στόχος είναι να υπολογίζεται έτσι ώστε η συνάρτηση γ να μειώνεται. Ο υπολογισμός του γίνεται με τον γνωστό κανόνα του Armijo (αναφορά [1]). Ο κανόνας Armijo απαιτεί η συνάρτηση γ από το σημείο x^k μέχρι το επόμενο σημείο

$$x^{k+1} = x^k + \sigma^k d^k$$

να μειώνεται αρκετά. Ο κανόνας του Armijo υλοποιείται στο Βήμα 4 του αλγορίθμου.

Το κριτήριο τερματισμού του αλγορίθμου είναι

$$\|d^k\|_2^2 \leq \varepsilon$$

όπου ε ένας πολύ μικρός θετικός αριθμός. Αν η κατεύθυνση έρευνας d^k (λύση του βοηθητικού προβλήματος τετραγωνικού προγραμματισμού) μηδενιστεί τότε από το (S) προκύπτει ότι το σημείο x^k ικανοποιεί τις αναγκαίες συνθήκες πρώτης τάξης άρα είναι η επιθυμητή λύση του προβλήματος. Για να τερματιστεί ο αλγόριθμος απαιτείται το μέτρο της κατεύθυνσης έρευνας να είναι μικρότερο από τον αριθμό ε δηλαδή να πλησιάζει κοντά στο 0 και άρα το x^k να αποτελεί επιθυμητή λύση του προβλήματος. Ανάλογα με την ακρίβεια του αποτελέσματος που θέλει κανείς μπορεί να βάλει για παράδειγμα $\varepsilon = 10^{-16}$ ή ακόμα μικρότερο.

Αφού επεξηγήθηκαν αναλυτικά τα επί μέρους βήματα, οι συναρτήσεις και οι παράμετροι που χρησιμοποιεί ο αλγόριθμος στη συνέχεια παρουσιάζεται η τελική του προγραμματιστική μορφή.

Αλγόριθμος SQP

Βήμα 0 $k = 0, x^0 \in \mathbb{R}^n, Q^0 = I_n, \beta \in (0,1), \alpha \in (0,0.5)$

$$0 < \varepsilon \ll 1, c^{-1} > 0, \delta > 0, \xi > 0$$

Βήμα 1 Επίλυση του γραμμικού συστήματος (S)

$$\begin{bmatrix} Q^k & \nabla h(x^k) \\ \nabla h(x^k)^T & 0 \end{bmatrix} \begin{bmatrix} d^k \\ \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla f(x^k) \\ -h(x^k) \end{bmatrix}$$

Βήμα 1 Έλεγχος παραμέτρου c^k

$$t(x^k, c^{k-1}) = (c^{k-1} - \xi) \|h(x^k)\|_1 - (\lambda^k)^T h(x^k)$$

Αν $t(x^k, c^{k-1}) \geq 0$ τότε

$$c^k = c^{k-1}$$

αλλιώς

$$c^k = \max\left\{c^{k-1} + \delta, \frac{(\lambda^k)^T h(x^k)}{\|h(x^k)\|_1} + \xi\right\}$$

Βήμα 2 Υπολογισμός της παραγώγου της γ στην κατεύθυνση d^k

$$\theta(x^k, d^k, c^k) = \nabla f(x)^T d^k - c^k \|h(x^k)\|_1$$

Βήμα 3 Αν $\|d^k\|_2^2 \leq \varepsilon$ τότε τερματισμός

Βήμα 4 Υπολογισμός μήκους βήματος $\sigma^k = \beta^{\eta_k}$. Εύρεση ελάχιστου μη αρνητικού ακεραίου η_k τέτοιο ώστε

$$\gamma(x^k + \beta^{\eta_k} d^k, c^k) - \gamma(x^k, c^k) \leq \alpha \beta^{\eta_k} \theta(x^k, d^k, c^k)$$

Βήμα 5 Ενημέρωση

$$p^k = \sigma^k d^k$$

$$x^{k+1} = x^k + p^k$$

$$q^k = \nabla f(x^{k+1}) + \nabla h(x^{k+1}) \lambda^k - \nabla f(x^k) - \nabla h(x^k) \lambda^k$$

$$r^k = \psi^k q^k + (1 - \psi^k) Q^k p^k$$

$$\psi^k = \begin{cases} 1, & \text{αν } v^k \geq 0.2 w^k \\ \frac{0.8 w^k}{w^k - v^k}, & \text{αν } v^k < 0.2 w^k \end{cases}$$

$$v^k = (p^k)^T q^k$$

$$w^k = (p^k)^T Q^k p^k$$

$$Q^{k+1} = Q^k - \frac{Q^k p^k (p^k)^T Q^k}{w^k} + \frac{r^k (r^k)^T}{(p^k)^T r^k}$$

$k = k + 1$ και επιστροφή στο **Βήμα 1**

Η προγραμματιστική υλοποίηση του αλγορίθμου έγινε στο περιβάλλον του MATLAB. Ο κώδικας και οι οδηγίες για να τρέξει κανείς το πρόγραμμα αναφέρονται στο Παράρτημα της εργασίας.

Η μόνη απαίτηση του αλγορίθμου SQP είναι να υπάρχουν διαθέσιμα σε αρχεία συναρτήσεων η αντικειμενική συνάρτηση κόστους, οι περιορισμοί και οι κλίσεις τους. Τα αρχεία αυτά πρέπει να δέχονται ένα σημείο x και να επιστρέφουν τις τιμές της εκάστοτε συνάρτησης στο σημείο αυτό. Οι συναρτήσεις αυτές καλούνται σε κάθε επανάληψη οπότε το να υπάρχουν σε ξεχωριστά αρχεία συναρτήσεων δίνουν μεγάλη ταχύτητα στις αριθμητικές πράξεις του αλγορίθμου. Η συνάρτηση SQP που έχει υλοποιηθεί παίρνει ως είσοδο την αντικειμενική συνάρτηση κόστους, την κλίση της, τους ισοτικούς περιορισμούς, την κλίση τους καθώς και το αρχικό σημείο x^0 . Επιστρέφει στο Command Window την βέλτιστη λύση \hat{x} , τους πολλαπλασιαστές Lagrange $\hat{\lambda}$, την ελάχιστη τιμή της αντικειμενικής συνάρτησης κόστους, και τον αριθμό των επαναλήψεων που χρειάστηκε ο αλγόριθμος για να φτάσει τη βέλτιστη λύση.

Οι αρχικές τιμές των σταθερών που χρησιμοποιήθηκαν για τον προγραμματισμό του SQP είναι

$$\beta = 0.7, \alpha = 0.1, c^{-1} = 10, \varepsilon = 10^{-16}, \delta = 2, \xi = 2$$

Κεφάλαιο 5

Αριθμητικά Πειράματα με τον Αλγόριθμο SQP

Στη συνέχεια θα χρησιμοποιηθεί ο αλγόριθμος SQP για να λυθούν πολλά προβλήματα της μορφής (P1). Παρουσιάζονται αρκετά παραδείγματα με διάφορων ειδών προβλήματα, με ποικίλες μορφές αντικειμενικών συναρτήσεων κόστους και ισοτικών περιορισμών. Τα παραδείγματα έχουν αντληθεί από τη βιβλιογραφία και πολλά από αυτά εμφανίζονται σε επιστημονικούς κλάδους όπως η οικονομία, η χημεία, η βιολογία κτλ. Για κάθε παράδειγμα αναφέρεται σε παρένθεση η βιβλιογραφική πηγή από όπου αντλήθηκε. Τα παραδείγματα που δεν φέρουν κάποια αναφορά είναι της έμπνευσης του επιβλέποντα της παρούσας διπλωματικής εργασίας, καθηγητή Ν.Μαράτου. Τα παραδείγματα και τα αποτελέσματα τους έχουν υλοποιηθεί με το λογισμικό MATLAB.

Αρχικά παρατίθεται το πρόβλημα κάθε παραδείγματος στη μορφή του βασικού προβλήματος (P1) για να φαίνονται καθαρά η αντικειμενική συνάρτηση κόστους και οι περιορισμοί. Για κάθε πρόβλημα καταγράφονται ο αριθμός των μεταβλητών και ο αριθμός των περιορισμών. Στη συνέχεια παρουσιάζεται η αναλυτική λύση \hat{x} για να γίνει η σύγκριση με τα αποτελέσματα του αλγορίθμου. Σκοπός της εργασίας είναι να μελετηθούν τα αποτελέσματα του αλγορίθμου SQP και για αυτό θα καταγραφούν ενδεικτικά τα αποτελέσματα που επιστρέφει η συνάρτηση `fmincon` χωρίς να δοθεί ιδιαίτερη έμφαση στην ανάλυση της καθώς δεν είναι στα πλαίσια της παρούσας εργασίας. Η συνάρτηση `fmincon` είναι συνάρτηση από τις βιβλιοθήκες του MATLAB και λύνει προβλήματα της μορφής (P1). Τα αποτελέσματα έχουν παρουσιαστεί σε πίνακες για να φαίνεται καθαρά το αποτέλεσμα που εμφανίζει η συνάρτηση SQP

συγκριτικά με την αναλυτική λύση αλλά και με το αποτέλεσμα της `fmincon`. Κάθε αριθμητικό παράδειγμα έχει τρέξει για διάφορα αρχικά σημεία x^0 τα οποία αναφέρονται στην πρώτη στήλη των πινάκων. Στη δεύτερη στήλη καταγράφεται ο αριθμός των επαναλήψεων που χρειάστηκε η κάθε συνάρτηση για να βρει την τελική λύση \hat{x} . Στην τρίτη στήλη των πινάκων παρουσιάζεται το αποτέλεσμα που επιστρέφει κάθε συνάρτηση. Για προβλήματα με 2 μεταβλητές ($n = 2$) έχουν γίνει και γραφικές παραστάσεις για να φανεί η συμπεριφορά του SQP για διάφορα αρχικά σημεία και τι ενδιάμεσα σημεία βρίσκει ο αλγόριθμος για κάθε ένα από αυτά.

Για τα αποτελέσματα των προγραμμάτων έχει επιλεγεί η λύση να εμφανίζεται με ακρίβεια 4 δεκαδικών ψηφίων για να μπορούν οι πίνακες να είναι ευανάγνωστοι. Αυτό γίνεται με την εντολή `format short` του MATLAB. Για μεγαλύτερη ακρίβεια μπορεί να χρησιμοποιηθεί η εντολή `format long` για ακρίβεια 9 δεκαδικών ψηφίων ή η εντολή `format shortEng` για ακρίβεια 4 δεκαδικών ψηφίων με μια δεκαδική δύναμη.

Για τη συνέχεια ορίζεται ως επιθυμητή λύση ή επιθυμητό αποτέλεσμα των αλγορίθμων κάθε σημείο \hat{x} το οποίο ανήκει στο σύνολο επιθυμητών σημείων D . Δηλαδή, κάθε σημείο \hat{x} που ικανοποιεί τις αναγκαίες συνθήκες $1^{ηs}$ τάξης. Το σύνολο D όμως περιέχει εκτός από τοπικά ελάχιστα, τοπικά μέγιστα και σαγματικά σημεία. Η αναλυτική λύση που αναφέρεται σε κάθε παράδειγμα είναι τοπικό ελάχιστο του προβλήματος. Σε κάποια προβλήματα όπου οι αλγόριθμοι επιστρέφουν επιθυμητή λύση \hat{x} , διαφορετική όμως από την αναλυτική λύση, υπολογίζεται η μήτρα

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2$$

με τη μέθοδο που αναφέρθηκε στο κεφάλαιο 3. Αν η μήτρα είναι θετικά ορισμένη, τότε το \hat{x} είναι ένα άλλο αυστηρό τοπικό ελάχιστο του προβλήματος.

Η μεγαλύτερη αριθμητική δεκαδική τιμή που μπορεί να επεξεργαστεί το MATLAB είναι

$$\text{realmax} = 1.7977 \times 10^{308}$$

και η αντίστοιχη μικρότερη είναι

$$\text{realmin} = 2.2251 \times 10^{-308}$$

Κάθε αριθμός με μεγαλύτερη τιμή από την `realmax` και μικρότερη από τη `realmin` εμφανίζεται στο MATLAB ως `Inf` (άπειρο). Συνεπώς όταν στους αλγόριθμους που τρέχει το MATLAB εμφανίζονται αριθμητικές πράξεις με τιμές εκτός των παραπάνω ορίων, οι μεταβλητές παίρνουν την τιμή `Inf`. Όταν γίνονται πράξεις με μεταβλητές που έχουν τιμή `Inf` το πρόγραμμα δίνει στις τελικές μεταβλητές την τιμή `NaN` (Not-a-Number) καθώς δεν γνωρίζει την τιμή τους. Έτσι το πρόγραμμα «κολλάει» και δεν εμφανίζει επιθυμητά αποτελέσματα στην έξοδο. Πατώντας `Ctrl+C` εξαναγκάζεται το πρόγραμμα να σταματήσει. Έπειτα, με τη βοήθεια των μεταβλητών `global` και τις εξωτερικής εμβέλειας που έχουν, μπορεί να κρατηθεί η τιμή των μεταβλητών αυτών για μερικές επαναλήψεις, όσο δηλαδή η τιμή της είναι στα παραπάνω όρια. Σε τέτοια παραδείγματα που οι τιμές των μεταβλητών γίνονται σε κάθε επανάληψη ολοένα και μεγαλύτερες ο αλγόριθμος SQP αποκλίνει, οι τιμές των μεταβλητών περνάνε τα παραπάνω όρια και το πρόγραμμα «κολλάει». Σε τέτοια παραδείγματα αναγράφεται στην τρίτη στήλη των πινάκων (που καταγράφονται τα αποτελέσματα του SQP) το χαρακτηριστικό «Not Converge».

Σε μερικά παραδείγματα όπου ο αλγόριθμος SQP αποκλίνει παρατηρήθηκε ότι αυξάνοντας την τιμή της παραμέτρου ποινής `c` κατά μια πολύ μεγάλη τιμή ο αλγόριθμος επιστρέφει επιθυμητό αποτέλεσμα. Αυτό είναι λογικό καθώς ο αλγόριθμος συγκλίνει για μεγάλες τιμές της παραμέτρου `c`. Η αρχική τιμή που έχει δοθεί στο πρόγραμμα είναι `c = 10`. Θα αναφέρεται ρητά στα παραδείγματα που έχει γίνει τροποποίηση της παραμέτρου `c`.

Το κριτήριο τερματισμού του αλγορίθμου SQP είναι

$$\|d^k\|_2^2 \leq \varepsilon \Leftrightarrow \|d^k\|_2 \leq \sqrt{\varepsilon}$$

Ανάλογα με την ακρίβεια που θέλει κανείς στα αποτελέσματα του, βάζει και την αντίστοιχη τιμή στο ε . Στον αλγόριθμο SQP η τιμή που έχει επιλεγεί είναι $\varepsilon = 10^{-16}$ για να επιτευχθεί αποτελέσματα με ακρίβεια 8 δεκαδικών ψηφίων.

Σε κάποια παραδείγματα ο αλγόριθμος SQP «κολλάει» και δεν εμφανίζει κάποιο αποτέλεσμα για $\varepsilon = 10^{-16}$, χωρίς ωστόσο να αποκλίνει. Σε τέτοια περίπτωση αν αλλάξει η τιμή του ε , το πρόγραμμα τρέχει και εμφανίζει επιθυμητό αποτέλεσμα. Στην συνέχεια θα αναφέρεται ρητά στα παραδείγματα στα οποία έχει γίνει τροποποίηση της σταθεράς $\varepsilon = 10^{-16}$, δηλαδή του κριτηρίου τερματισμού.

Στην συνέχεια παρουσιάζονται τα 24 παραδείγματα που επιλύθηκαν με τον αλγόριθμο SQP.

Παράδειγμα 1

(Αναφορά [1])

$$\min_{x \in \mathbb{R}^2} \{ x_1^2 + x_2^2 \}$$

$$\text{subject to } x_1 + 2x_2 - 6 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (1.2, 2.4), \hat{\lambda} = -2.4, f(\hat{x}) = 7.2$$

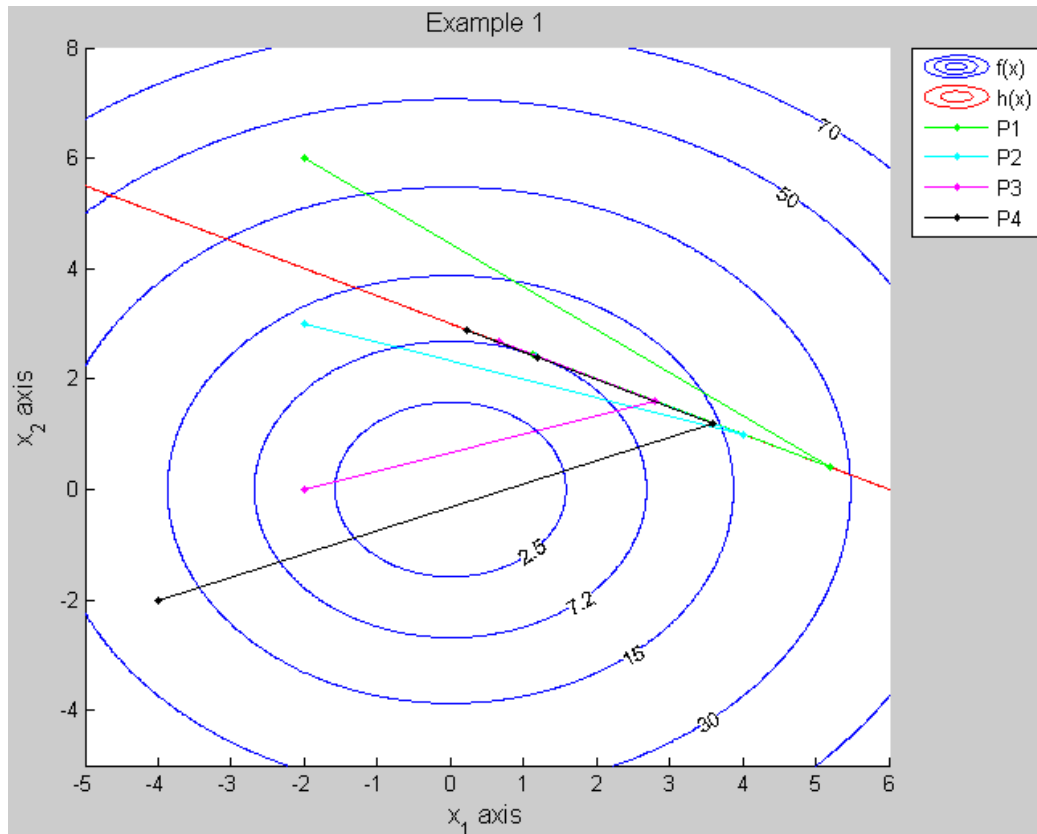
- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (-2,6)	3	(1.2000, 2.4000)
P2: (-2,3)	3	(1.2000, 2.4000)
P3: (-2,0)	3	(1.2000, 2.4000)
P4: (-4, -2)	3	(1.2000, 2.4000)
(7, 7)	3	(1.2000, 2.4000)
(1, -9)	3	(1.2000, 2.4000)
(150, -100)	4	(1.2000, 2.4000)

fmincon		
x^0	iterations	\hat{x}
P1: (-2,6)	3	(1.2000, 2.4000)
P2: (-2,3)	4	(1.2000, 2.4000)
P3: (-2,0)	3	(1.2000, 2.4000)
P4: (-4, -2)	3	(1.2000, 2.4000)
(7, 7)	3	(1.2000, 2.4000)
(1, -9)	3	(1.2000, 2.4000)
(150, -100)	3	(1.2000, 2.4000)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Με μπλε χρώμα φαίνονται οι ισοϋψείς της αντικειμενικής συνάρτησης και η τιμή της σε κάθε καμπύλη, με κόκκινο ο περιορισμός και τα 4 αρχικά σημεία P1, P2, P3, P4 με διαφορετικά χρώματα το καθένα, τα ενδιάμεσα σημεία που βρίσκει ο αλγόριθμος για κάθε αρχικό σημείο μέχρι να φτάσει στην επιθυμητή λύση.



Παράδειγμα 2

(Αναφορά [8])

$$\min_{x \in \mathbb{R}^2} \{ 2(x_1^2 + x_2^2 - 1) - x_1 \}$$

$$\text{subject to } x_1^2 + x_2^2 - 1 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (1, 0), \hat{\lambda} = -1.5, f(\hat{x}) = -1$$

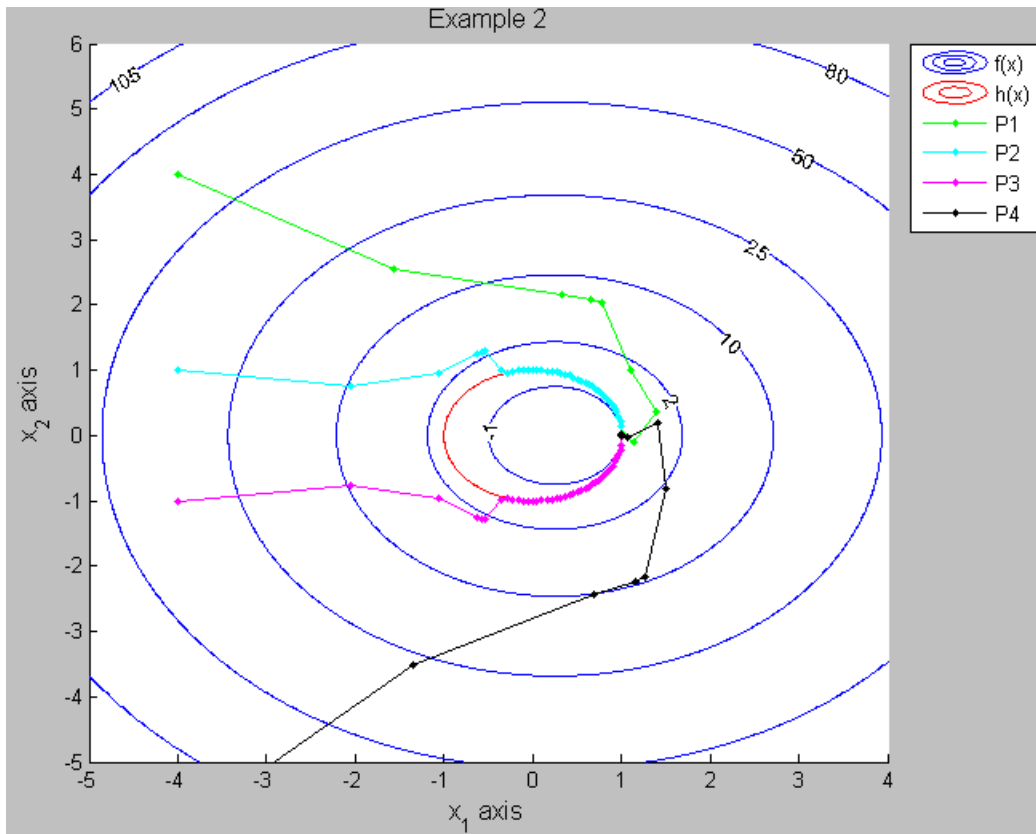
- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (-4,4)	12	(1.0000, 0.0000)
P2: (-4,1)	43	(1.0000, 0.0000)
P3: (-4, -1)	43	(1.0000, 0.0000)
P4: (-4, -6)	11	(1.0000, 0.0000)
(1, -5)	9	(1.0000, 0.0000)
(4, 8)	9	(1.0000, 0.0000)
(-2, -9)	11	(1.0000, 0.0000)
(-100, 100)	26	(1.0000, 0.0000)

fmincon		
x^0	iterations	\hat{x}
P1: (-4,4)	18	(1.0000, 0.0000)
P2: (-4,1)	15	(1.0000, 0.0000)
P3: (-4, -1)	15	(1.0000, 0.0000)
P4: (-4, -6)	13	(1.0000, 0.0000)
(1, -5)	8	(1.0000, 0.0000)
(4, 8)	9	(1.0000, 0.0000)
(-2, -9)	11	(1.0000, 0.0000)
(-100, 100)	21	(1.0000, 0.0000)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Παρατηρήθηκε ότι για τα αρχικά σημεία P2 και P3 ο αλγόριθμος εκτελεί μεγάλο αριθμό επαναλήψεων και τα ενδιάμεσα σημεία βρίσκονται πάνω στον περιορισμό. Γενικά για να έχουμε υπεργραμμική ταχύτητα σύγκλισης θέλουμε το μήκος βήματος να έχει τιμή $\sigma_k = 1$. Στα σημεία αυτά το μήκος βήματος μειώνεται κάτω από την τιμή 1 σε κάθε επανάληψη, η ταχύτητα σύγκλισης παύει να είναι υπεργραμμική (γίνεται γραμμική) και συνεπώς ο αλγόριθμος εκτελεί πολλές επαναλήψεις μέχρι να φτάσει στην επιθυμητή λύση. Το φαινόμενο αυτό είναι γνωστό ως Maratos Effect (αναφορά [8]) καθώς πρώτος το παρατήρησε ο επιβλέπων της παρούσας εργασίας Ν. Μαράτος.



Παράδειγμα 3

(Αναφορά [8])

$$\min_{x \in \mathbb{R}^2} \{ x_1^2 + x_2^2 \}$$

$$\text{subject to } (x_1 + 1)^2 + x_2^2 - 4 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

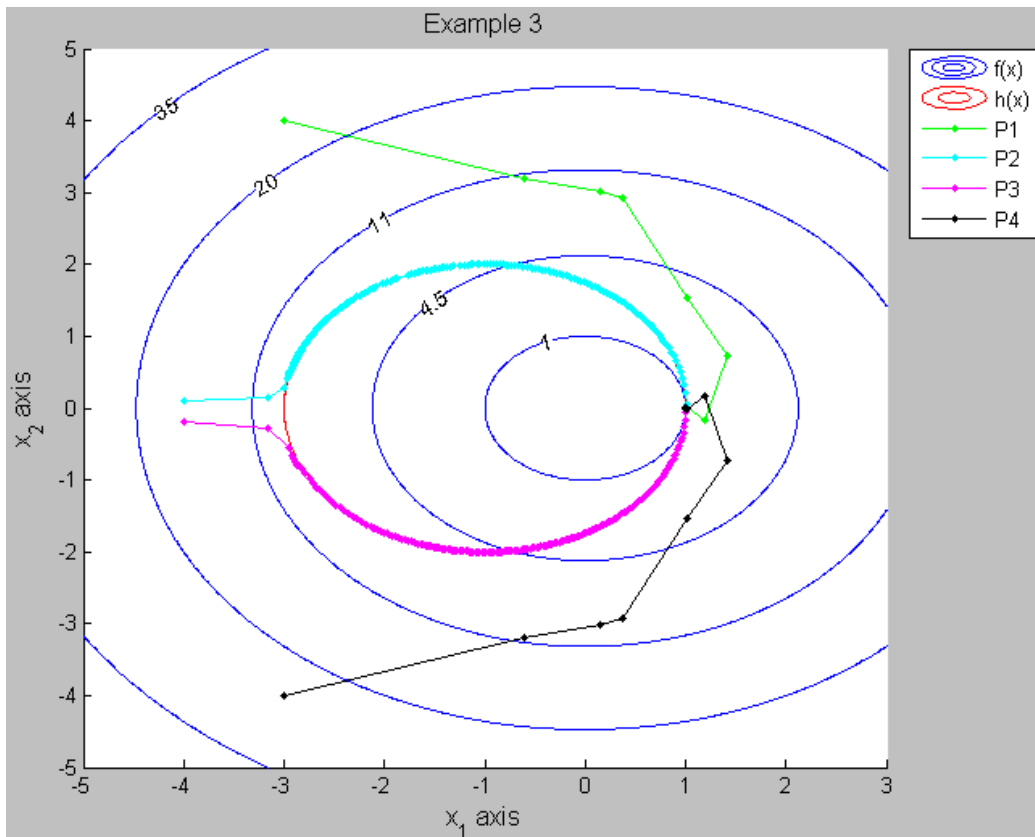
$$\hat{x} = (1, 0), \hat{\lambda} = -0.5, f(\hat{x}) = 1$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (-3,4)	10	(1.0000, 0.0000)
P2: (-4,0.1)	206	(1.0000, 0.0000)
P3: (-4, -0.2)	184	(1.0000, 0.0000)
P4: (-3, -4)	10	(1.0000, 0.0000)
(4, 7)	8	(1.0000, 0.0000)
(-6, 9)	10	(1.0000, 0.0000)
(2, -10)	9	(1.0000, 0.0000)
(-90, -200)	17	(1.0000, 0.0000)

fmincon		
x^0	iterations	\hat{x}
P1: (-3,4)	10	(1.0000, 0.0000)
P2: (-4,0.1)	21	(1.0000, 0.0000)
P3: (-4, -0.2)	19	(1.0000, 0.0000)
P4: (-3, -4)	10	(1.0000, 0.0000)
(4, 7)	8	(1.0000, 0.0000)
(-6, 9)	11	(1.0000, 0.0000)
(2, -10)	8	(1.0000, 0.0000)
(-90, -200)	17	(1.0000, 0.0000)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις. Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Το Maratos Effect που σχολιάστηκε προηγουμένως εμφανίζεται και σε αυτό το πρόβλημα.



Παράδειγμα 4

(Αναφορά [21])

$$\min_{x \in \mathbb{R}^2} \{ x_1^2 + 2x_2^2 \}$$

$$\text{subject to } (x_1 - 2)^2 + 2(x_2 - 1)^2 - 1 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

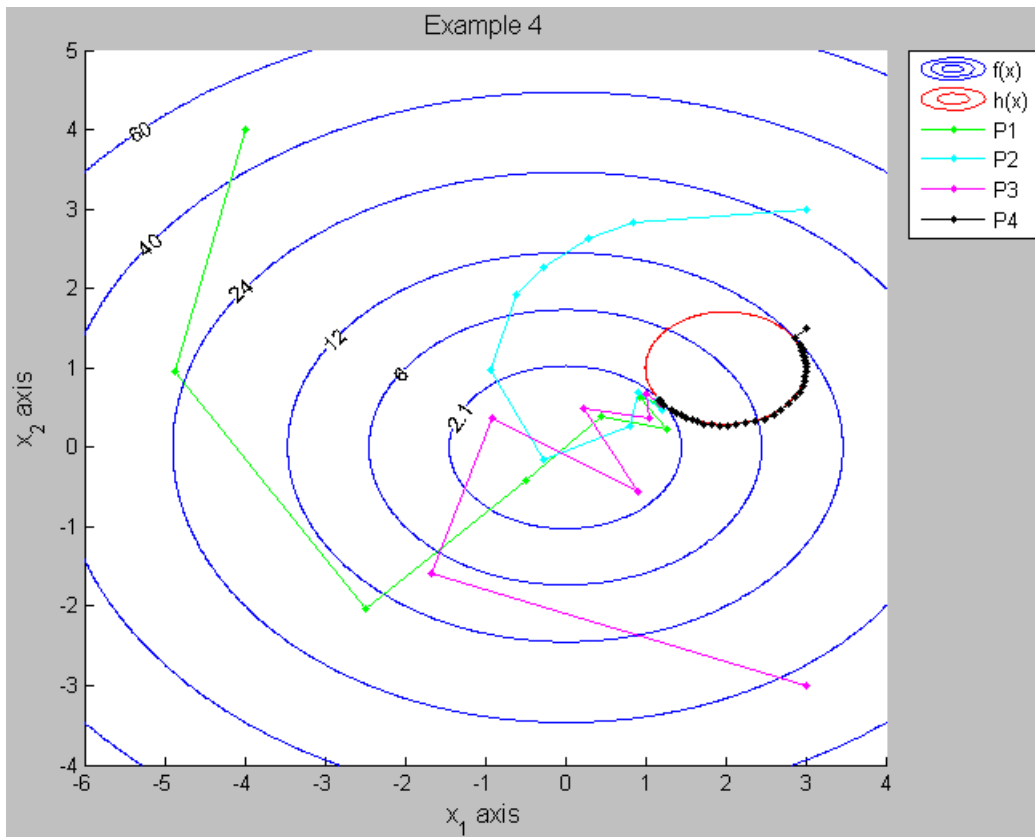
$$\hat{x} = (1.1835, 0.5918), \hat{\lambda} = 1.4495, f(\hat{x}) = 2.1010$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (-4,4)	12	(1.1835, 0.5918)
P2: (3,3)	14	(1.1835, 0.5918)
P3: (3, -3)	11	(1.1835, 0.5918)
P4: (3,1.5)	40	(1.1835, 0.5918)
(0, 2)	11	(1.1835, 0.5918)
(-5, 8)	12	(1.1835, 0.5918)
(1, 30)	15	(1.1835, 0.5918)
(40, -500)	20	(1.1835, 0.5918)

fmincon		
x^0	iterations	\hat{x}
P1: (-4,4)	11	(1.1835, 0.5918)
P2: (3,3)	15	(1.1835, 0.5918)
P3: (3, -3)	11	(1.1835, 0.5918)
P4: (3,1.5)	15	(1.1835, 0.5918)
(0, 2)	10	(1.1835, 0.5918)
(-5, 8)	13	(1.1835, 0.5918)
(1, 30)	16	(1.1835, 0.5918)
(40, -500)	19	(1.1835, 0.5918)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις. Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Το Maratos Effect που σχολιάστηκε προηγουμένως εμφανίζεται και σε αυτό το πρόβλημα.



Παράδειγμα 5

(Αναφορά [20])

$$\min_{x \in \mathbb{R}^2} \{ -x_1 + 10(x_1^2 + x_2^2 - 1) \}$$

$$\text{subject to } x_1^2 + x_2^2 - 1 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (1, 0), \hat{\lambda} = -9.5, f(\hat{x}) = -1$$

- Αλγοριθμική λύση:

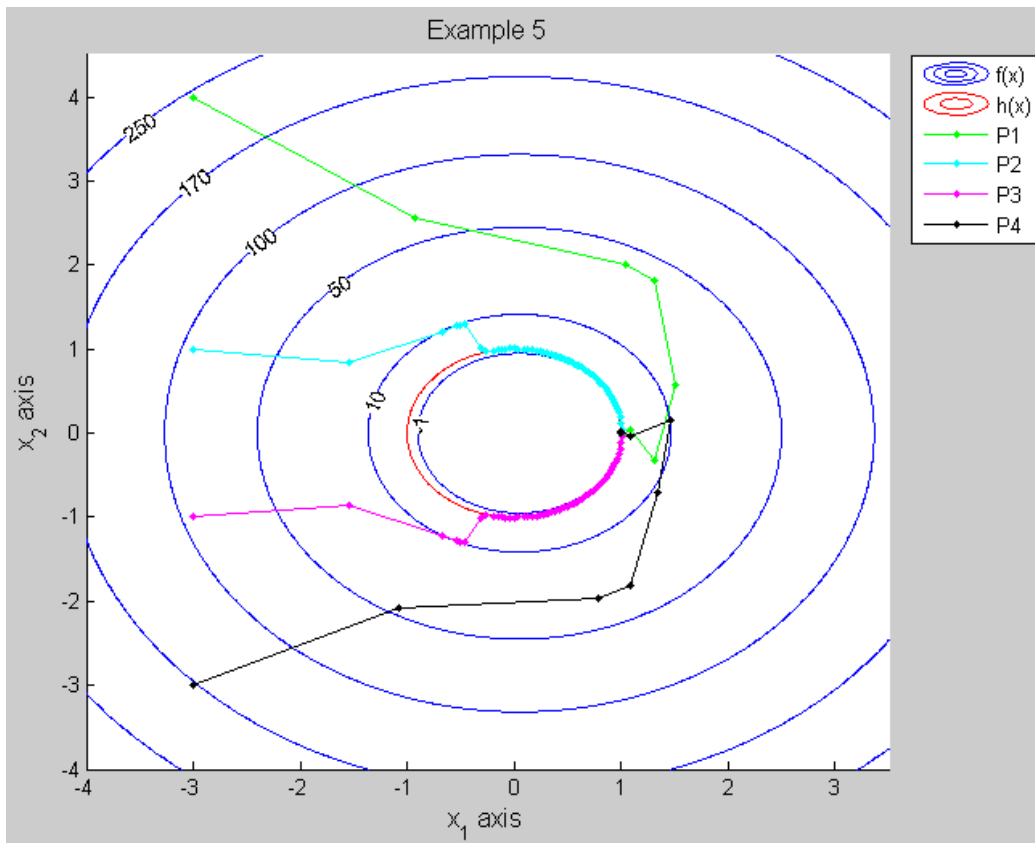
SQP		
x^0	iterations	\hat{x}
P1: (-3,4)	10	(1.0000, 0.0000)
P2: (-3,1)	62	(1.0000, 0.0000)
P3: (-3, -1)	62	(1.0000, 0.0000)
P4: (-3, -3)	11	(1.0000, 0.0000)
(-6, 8)	12	(1.0000, 0.0000)
(-2, 2)	13	(1.0000, 0.0000)
(0, 1)	35	(1.0000, 0.0000)
(0, -1)	35	(1.0000, 0.0000)
(-5, 5)	12	(1.0000, 0.0000)

fmincon		
x^0	iterations	\hat{x}
P1: (-3,4)	19	(1.0000, 0.0000)
P2: (-3,1)	20	(1.0000, 0.0000)
P3: (-3, -1)	20	(1.0000, 0.0000)
P4: (-3, -3)	17	(1.0000, 0.0000)
(-6, 8)	-	-
(-2, 2)	-	-
(0, 1)	-	-
(0, -1)	-	-
(-5, 5)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση από την SQP.

Για τα αρχικά σημεία $x^0 = (-6, 8)$, $x^0 = (-2, 2)$, $x^0 = (0, 1)$, $x^0 = (0, -1)$ και $x^0 = (-5, 5)$ η συνάρτηση fmincon σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 12,13,35,35 και 12 επαναλήψεις αντίστοιχα.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Το Maratos Effect που σχολιάσαμε προηγουμένως εμφανίζεται και σε αυτό το πρόβλημα.



Παράδειγμα 6

(Αναφορά [14])

$$\min_{x \in \mathbb{R}^2} \{ (x_1 - 2)^2 + (x_2 - 1)^2 \}$$

$$\text{subject to } 0.25x_1^2 + x_2^2 - 1 \leq 0$$

$$x_1 - 2x_2 + 1 = 0$$

Με βάση τη μορφή (P2') που αναφέρθηκε στο δεύτερο κεφάλαιο το πρόβλημα αυτό είναι ισοδύναμο με το

$$\min_{x \in \mathbb{R}^3} \{ (x_1 - 2)^2 + (x_2 - 1)^2 \}$$

$$\text{subject to } 0.25x_1^2 + x_2^2 - 1 + x_3^2 = 0$$

$$x_1 - 2x_2 + 1 = 0$$

- $n = 3, m = 2$
- Αναλυτική λύση για το αρχικό πρόβλημα:

$$\hat{x} = (0.8229, 0.9114), \hat{\lambda} = (1.8466, 1.5945), f(\hat{x}) = 1.3935$$

- Αλγοριθμική λύση για το ισοδύναμο πρόβλημα:

SQP		
x^0	iterations	\hat{x}
(1, 1, 1)	9	(0.8229, 0.9114, 0.0000)
(-2, 3, -5)	12	(0.8229, 0.9114, 0.0000)
(2, 0, -2)	10	(0.8229, 0.9114, 0.0000)
(21, 11, 17)	15	(0.8229, 0.9114, 0.0000)
(7, -9, 11)	14	(0.8229, 0.9114, 0.0000)
(-18, -3, -13)	15	(0.8229, 0.9114, 0.0000)
(25, -30, 100)	15	(0.8229, 0.9114, 0.0000)

fmincon		
x^0	iterations	\hat{x}
(1, 1, 1)	8	(0.8229, 0.9114, 0.0000)
(-2, 3, -5)	12	(0.8229, 0.9114, 0.0000)
(2, 0, -2)	10	(0.8229, 0.9114, 0.0000)
(21, 11, 17)	16	(0.8229, 0.9114, 0.0000)
(7, -9, 11)	14	(0.8229, 0.9114, 0.0000)
(-18, -3, -13)	14	(0.8229, 0.9114, 0.0000)
(25, -30, 100)	18	(0.8229, 0.9114, 0.0000)

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα με ανισοτικούς περιορισμούς που ανάγεται σε πρόβλημα με ισοτικούς με βάση τη μορφή (P2') που αναφέρθηκε στο κεφάλαιο 2. Το ισοδύναμο πρόβλημα είναι 3 μεταβλητών με 2 περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις.

Παράδειγμα 7

(Αναφορά [22])

$$\min_{x \in \mathbb{R}^3} \{ -x_1 x_2 x_3 \}$$

$$\text{subject to } x_1 + 2x_2 + 2x_3 - 72 = 0$$

- $n = 3, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (24, 12, 12), \hat{\lambda} = 144, f(\hat{x}) = -3456$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
(10,10,10)	11	(24.0000, 12.0000, 12.0000)
(0,0,0)	10	(24.0000, 12.0000, 12.0000)
(-1,2,-7)	12	(24.0000, 12.0000, 12.0000)
(-5,0,-4)	18	(24.0000, 12.0000, 12.0000)
(1,2,2)	10	(24.0000, 12.0000, 12.0000)
(-3,-4,-2)	-	Not Converge

fmincon		
x^0	iterations	\hat{x}
(10,10,10)	8	(24.0000, 12.0000, 12.0000)
(0,0,0)	9	(24.0000, 12.0000, 12.0000)
(-1,2,-7)	-	-
(-5,0,-4)	15	(24.0000, 12.0000, 12.0000)
(1,2,2)	-	-
(-3,-4,-2)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 3 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

Για τα αρχικά σημεία $x^0 = (-1, 2, -7)$ και $x_0 = (1, 2, 2)$ η συνάρτηση `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 12 και 10 επαναλήψεις αντίστοιχα.

Για το αρχικό σημείο $x^0 = (1, 2, 2)$ η SQP «κολλάει» και δεν επιστρέφει κανένα αποτέλεσμα. Αλλάζοντας την τιμή της παραμέτρου ποινής σε $c = 10^{20}$ έχουμε λύση σε 10 επαναλήψεις.

Για $x^0 = (-3, -4, -2)$ η SQP αποκλίνει καθώς όπως βλέπουμε στον παρακάτω πίνακα η τιμή του x^k ήδη στην 6^η επανάληψη γίνεται πολύ μεγάλη και γίνονται πράξεις με τιμές εκτός των ορίων του MATLAB. Προφανώς και η `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα για τα σημεία αυτά.

x^0	k	x^k	c^k
(-3, -4, -2)	6	$(3.2 \times 10^{45}, -0.9 \times 10^{45}, -0.6 \times 10^{45})$	5.6×10^{40}
	7	$(30.0 \times 10^{93}, -8.9 \times 10^{93}, -6.5 \times 10^{93})$	5.6×10^{40}
	7		2.4×10^{192}
	8	$(9.3 \times 10^{192}, -2.8 \times 10^{192}, -1.9 \times 10^{192})$	2.4×10^{192}

Παράδειγμα 8

(Αναφορά [18])

$$\begin{aligned} & \min_{x \in \mathbb{R}^2} \{ x_2 \} \\ \text{subject to} \quad & 2x_1^2 - x_1^3 - x_2 \leq 0 \\ & 2(1 - x_1)^2 - (1 - x_1)^3 - x_2 \leq 0 \end{aligned}$$

Με βάση τη μορφή (P2') που αναφέρθηκε στο δεύτερο κεφάλαιο το πρόβλημα αυτό είναι ισοδύναμο με το

$$\min_{x \in \mathbb{R}^4} \{ x_2 \}$$

$$\text{subject to } 2x_1^2 - x_1^3 - x_2 + x_3^2 = 0$$

$$2(1 - x_1)^2 - (1 - x_1)^3 - x_2 + x_4^2 = 0$$

- $n = 4, m = 2$
- Αναλυτική λύση για το αρχικό πρόβλημα:

$$\hat{x} = (0.5, 0.375), \hat{\lambda} = (0.5, 0.5), f(\hat{x}) = 0.375$$

- Αλγοριθμική λύση για το ισοδύναμο πρόβλημα:

SQP		
x^0	iterations	\hat{x}
(0,0,-6,1)	12	(0.5000, 0.3750, 0.0000, 0.0000)
(1,0,-1,-4)	14	(0.5000, 0.3750, 0.0000, 0.0000)
(1,-2,-1,-10)	15	(0.5000, 0.3750, 0.0000, 0.0000)
(3,-4,5,1)	15	(0.5000, 0.3750, 0.0000, 0.0000)
(12,45,-12,210)	56	(0.5000, 0.3750, 0.0000, 0.0000)
(4,-5,6,-1)	14	(0.5000, 0.3750, 0.0000, 0.0000)
(55,-60,50,65)	41	(0.5000, 0.3750, 0.0000, 0.0000)

fmincon		
x^0	iterations	\hat{x}
(0,0,-6,1)	11	(0.5000, 0.3750, 0.0000, 0.0000)
(1,0,-1,-4)	12	(0.5000, 0.3750, 0.0000, 0.0000)
(1,-2,-1,-10)	14	(0.5000, 0.3750, 0.0000, 0.0000)
(3,-4,5,1)	13	(0.5000, 0.3750, 0.0000, 0.0000)
(12,45,-12,210)	32	(0.5000, 0.3750, 0.0000, 0.0000)
(4,-5,6,-1)	10	(0.5000, 0.3750, 0.0000, 0.0000)
(55,-60,50,65)	21	(0.5000, 0.3750, 0.0000, 0.0000)

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα με ανισοτικούς περιορισμούς που ανάγεται σε πρόβλημα με ισοτικούς με βάση τη μορφή (P2') που αναφέρθηκε στο κεφάλαιο 2. Το ισοδύναμο πρόβλημα είναι 4 μεταβλητών με 2 περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

Παράδειγμα 9

(Αναφορά [13])

$$\begin{aligned} & \min_{x \in \mathbb{R}^2} \{ -1 \} \\ \text{subject to} \quad & x_1^2 + x_2^2 - 25 = 0 \\ & x_1 x_2 - 9 = 0 \end{aligned}$$

- $n = 2, m = 2$
- Αναλυτική λύση (4 λύσεις):

$$\hat{x} = \{ (4.6016, 1.9558), (-4.6016, -1.9558), (1.9558, 4.6016), (-1.9558, -4.6016) \}$$

$$\lambda = (0, 0), f(\hat{x}) = -1$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
(2, -1)	5	(4.6016, 1.9558)
(0, 1)	5	(1.9558, 4.6016)
(-3, 8)	6	(1.9558, 4.6016)
(-15, -7)	6	(-4.6016, -1.9558)
(30, 60)	8	(1.9558, 4.6016)
(3, 8)	5	(1.9558, 4.6016)
(-150, -120)	9	(-4.6016, -1.9558)
(1, -20)	7	(-1.9558, -4.6016)

fmincon		
x^0	iterations	\hat{x}
(2, -1)	5	(4.6016, 1.9558)
(0, 1)	5	(1.9558, 4.6016)
(-3, 8)	6	(1.9558, 4.6016)
(-15, -7)	6	(-4.6016, -1.9558)
(30, 60)	8	(1.9558, 4.6016)
(3, 8)	5	(1.9558, 4.6016)
(-150, -120)	9	(-4.6016, -1.9558)
(1, -20)	7	(-1.9558, -4.6016)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 2 μεταβλητών με 2 περιορισμούς με σταθερή αντικειμενική συνάρτηση κόστους. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις.

Παράδειγμα 10

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^2} \{ \ln(1 + x_1^2) - x_2 \}$$

$$\text{subject to } (1 + x_1^2)^2 + x_2^2 - 4 = 0$$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (0, 1.7321), \hat{\lambda} = 0.2887, f(\hat{x}) = -1.7321$$

- Αλγοριθμική λύση:

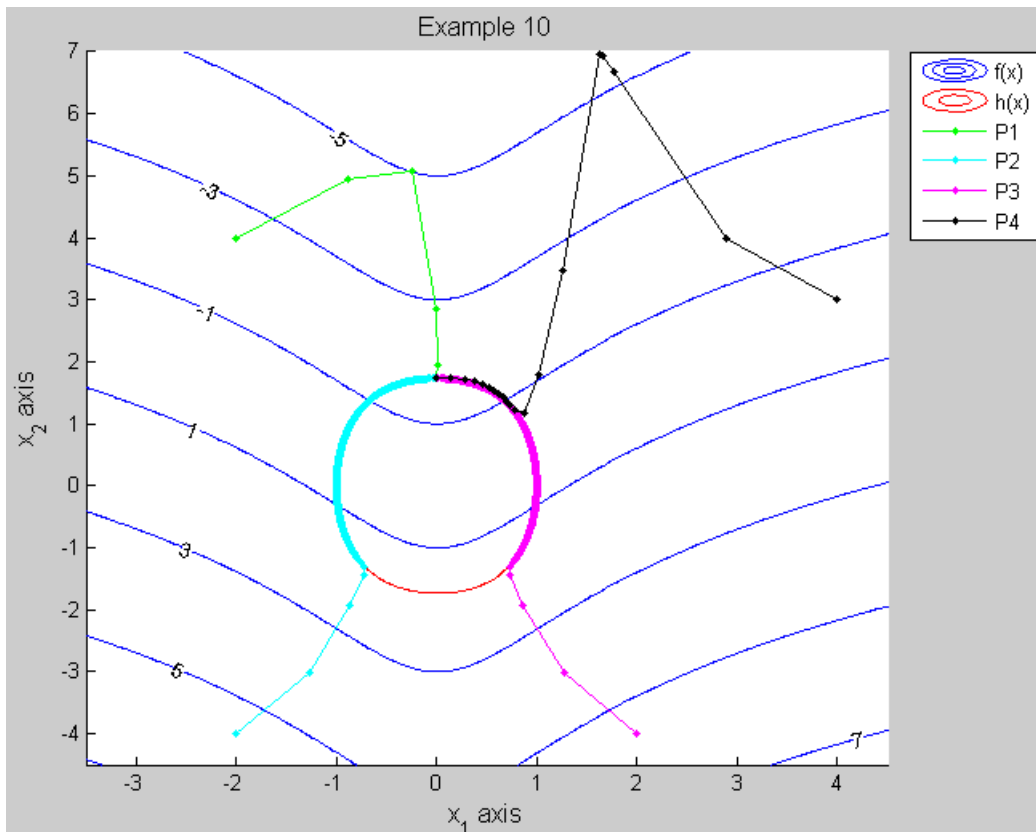
SQP		
x^0	iterations	\hat{x}
P1: (-2,4)	8	(0.0000, 1.7321)
P2: (-2, -4)	726	(0.0000, 1.7321)
P3: (2, -4)	726	(0.0000, 1.7321)
P4: (4,3)	22	(0.0000, 1.7321)
(-10, -10)	16	(0.0000, 1.7321)
(-5,3)	13	(0.0000, 1.7321)
(8, -13)	55	(0.0000, 1.7321)
(150, 100)	29	(0.0000, 1.7321)
(-30, -30)	31	(0.0000, 1.7321)

fmincon		
x^0	iterations	\hat{x}
P1: (-2,4)	9	(0.0000, 1.7321)
P2: (-2, -4)	17	(0.0000, 1.7321)
P3: (2, -4)	17	(0.0000, 1.7321)
P4: (4,3)	11	(0.0000, 1.7321)
(-10, -10)	-	-
(-5,3)	14	(0.0000, 1.7321)
(8, -13)	24	(0.0000, 1.7321)
(150, 100)	-	-
(-30, -30)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 2 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση από την SQP.

Για τα αρχικά σημεία $x^0 = (-10, -10)$, $x^0 = (150, 100)$ και $x^0 = (-30, -30)$ η συνάρτηση fmincon σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 16, 29 και 31 επαναλήψεις αντίστοιχα.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος. Το Maratos Effect που σχολιάστηκε προηγουμένως εμφανίζεται και σε αυτό το πρόβλημα.



Παράδειγμα 11

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^3} \{ (x_1 - x_2)^2 + (x_2 - x_3)^4 \}$$

$$\text{subject to } (1 + x_2^2) x_1 + x_3^4 - 3 = 0$$

- $n = 3, m = 1$
- Αναλυτική λύση (2 σημεία):

$$\hat{x} = \{ (1,1,1), (-1.8105, -1.8105, -1.8105) \}$$

$$\hat{\lambda} = 0, f(\hat{x}) = 0$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
(1, -5,3)	65	(1.0000, 1.0000,1.0000)
(-5,2, - 7)	226	(-1.8105, -1.8105, -1.8105)
(-100,100,100)	817	(1.0000, 1.0000,1.0000)
(0,0, 0)	286	(1.0000, 1.0000,1.0000)
(-2,1, 3)	1446	(1.0000, 1.0000,1.0000)
(-3, -4, -10)	598	(-1.8105, -1.8105, -1.8105)

fmincon		
x^0	iterations	\hat{x}
(1, -5,3)	26	(1.0022, 1.0022,0.9978)
(-5,2, - 7)	29	(-1.8122, -1.8122, -1.8113)
(-100,100,100)	40	(1.0028, 1.0028,0.9972)
(0,0, 0)	24	(0.9975, 0.9975,1.0025)
(-2,1, 3)	21	(1.0027, 1.0027,0.9973)
(-3, -4, -10)	27	(-1.8160, -1.8160, -1.8130)

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 3 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 . Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση από την SQP με κατάλληλη τροποποίηση της παραμέτρου τερματισμού ϵ .

Σε αυτό το πρόβλημα τα αποτελέσματα που επιστρέφει η fmincon δεν είναι καθόλου ακριβή αφού οι λύσεις παρουσιάζουν απόκλιση από την αναλυτική λύση, σε αντίθεση με την SQP που βρίσκει επιθυμητά αποτελέσματα με μεγάλη ακρίβεια.

Για το αρχικό σημείο $x^0 = (0,0,0)$ και $x^0 = (-3,-4,-10)$ η SQP δεν επιστρέφει επιθυμητό αποτέλεσμα για $\varepsilon = 10^{-16}$. Αλλάζοντας την τιμή αυτή σε $\varepsilon = 10^{-14}$ έχουμε επιθυμητό αποτέλεσμα σε 286 και 1446 επαναλήψεις αντίστοιχα.

Για το αρχικό σημείο $x^0 = (-3,-4,-10)$ η SQP δεν επιστρέφει επιθυμητό αποτέλεσμα για $\varepsilon = 10^{-16}$ ούτε για $\varepsilon = 10^{-14}$. Αλλάζοντας την τιμή σε $\varepsilon = 10^{-13}$ έχουμε την τελική λύση σε 598 επαναλήψεις.

Παράδειγμα 12

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^2} \{ 0.01 (x_1 - 1)^2 + (x_2 - x_1^2)^2 \}$$

$$\text{subject to } x_1 + 1 \leq 0$$

Με βάση τη μορφή (P2') που αναφέρθηκε στο δεύτερο κεφάλαιο το πρόβλημα αυτό είναι ισοδύναμο με το

$$\min_{x \in \mathbb{R}^3} \{ 0.01 (x_1 - 1)^2 + (x_2 - x_1^2)^2 \}$$

$$\text{subject to } x_1 + x_3^2 + 1 = 0$$

- $n = 3, m = 1$
- Αναλυτική λύση για το αρχικό πρόβλημα:

$$\hat{x} = (-1, 1), \hat{\lambda} = 0.04, f(\hat{x}) = 0.04$$

- Αλγοριθμική λύση για το ισοδύναμο πρόβλημα:

SQP		
x^0	iterations	\hat{x}
P1 (-3,6, -1)	108	(-1.0000, 1.0000,0.0000)
P2 (-3,0, -1)	568	(-1.0000, 1.0000,0.0000)
P3 (-3, -4, -1)	307	(-1.0000, 1.0000,0.0000)
(-5, -4, -10)	2513	(-1.0000, 1.0000,0.0000)
(7,12, 134)	12401	(-1.0000, 1.0000,0.0000)
(12, -9, 1)	4554	(-1.0000, 1.0000,0.0000)
(-10, -5, -12)	11858	(-1.0000, 1.0000,0.0000)
(-11, -5, 7)	39206	(-1.0000, 1.0000,0.0000)

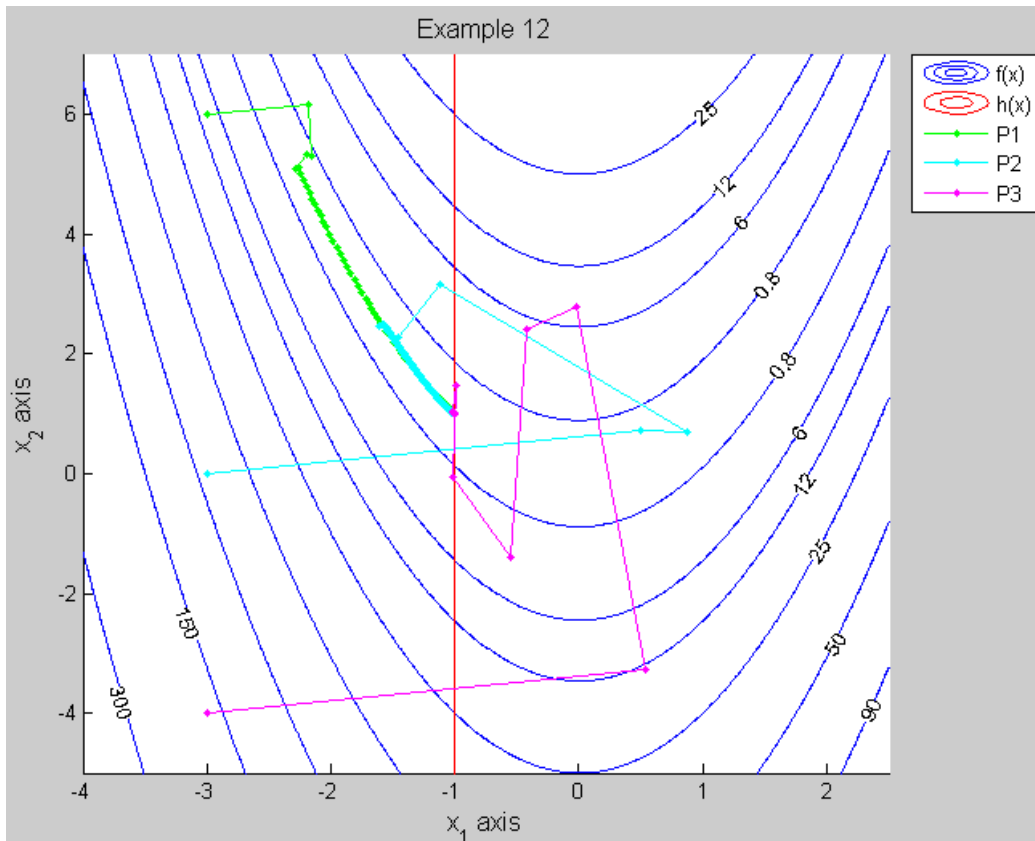
fmincon		
x^0	iterations	\hat{x}
P1 (-3,6, -1)	25	(-1.0000, 1.0000,0.0000)
P2 (-3,0, -1)	25	(-1.0000, 1.0000,0.0000)
P3 (-3, -4, -1)	33	(-1.0000, 1.0000,0.0000)
(-5, -4, -10)	26	(-1.0000, 1.0000,0.0000)
(7,12, 134)	34	(-1.0000, 1.0000,0.0000)
(12, -9, 1)	53	(-1.0000, 1.0000,0.0000)
(-10, -5, -12)	36	-
(-11, -5, 7)	35	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα με ανισοτικούς περιορισμούς που ανάγεται σε πρόβλημα με ισοτικούς με βάση τη μορφή (P2') που αναφέρθηκε στο κεφάλαιο 2. Το ισοδύναμο πρόβλημα είναι 3 μεταβλητών με 1 μη γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση από την SQP.

Για $x^0 = (12, -9, 1)$ η SQP δεν επιστρέφει επιθυμητό αποτέλεσμα για $\varepsilon = 10^{-16}$. Αλλάζοντας την τιμή αυτή σε $\varepsilon = 10^{-14}$ έχουμε επιθυμητό αποτέλεσμα σε 4554 επαναλήψεις.

Για τα αρχικά σημεία $x^0 = (-10, -5, -12)$ και $x^0 = (-11, -5, 7)$ η συνάρτηση `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 11858 και 39206 επαναλήψεις αντίστοιχα.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Παράδειγμα 13

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^3} \{ (x_1 + x_2)^2 + (x_2 + x_3)^2 \}$$

$$\text{subject to } x_1 + 2x_2 + 3x_3 - 1 = 0$$

- $n = 3, m = 1$
- Αναλυτική λύση :

$$\hat{x} = (0.5, -0.5, 0.5), \hat{\lambda} = 0, f(\hat{x}) = 0$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
(2,11,3)	9	(0.5000, -0.5000,0.5000)
(13, -1,9)	8	(0.5000, -0.5000,0.5000)
(-4,8,18)	9	(0.5000, -0.5000,0.5000)
(-2, -6, -4)	10	(0.5000, -0.5000,0.5000)
(-150,180,120)	9	(0.5000, -0.5000,0.5000)

fmincon		
x^0	iterations	\hat{x}
(2,11,3)	9	(0.5000, -0.5000,0.5000)
(13, -1,9)	9	(0.5000, -0.5000,0.5000)
(-4,8,18)	9	(0.5000, -0.5000,0.5000)
(-2, -6, -4)	9	(0.5000, -0.5000,0.5000)
(-150,180,120)	9	(0.5000, -0.5000,0.5000)

Στο παράδειγμα αυτό παρουσιάζεται ένα απλό πρόβλημα 3 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις. Έγιναν δοκιμές με πολλά αρχικά σημεία και για όλα υπήρχε επιθυμητή λύση με μεγάλη ακρίβεια και από τις δυο συναρτήσεις.

Παράδειγμα 14

(Αναφορά [13])

$$\begin{aligned} \min_{x \in \mathbb{R}^4} \{ & -x_1 x_2 x_3 x_4 \} \\ \text{subject to } & x_1^3 + x_2^2 - 1 = 0 \\ & x_1^2 x_4 - x_3 = 0 \\ & x_4^2 - x_2 = 0 \end{aligned}$$

- $n = 4, m = 3$

- Αναλυτική λύση (2 λύσεις):

$$\hat{x} = (0.7937, 0.7071, \pm 0.5297, \pm 0.8409)$$

$$\hat{\lambda} = (0.5, \mp 0.4719, 0.3536), f(\hat{x}) = -0.25$$

- Αλγοριθμική λύση:

SQP			
x_0	iterations	\hat{x}	$f(\hat{x})$
(0.8,0.8,0.8, 0.8)	7	(0.7937,0.7071, 0.5297,0.8409)	-0.25
(7, -3,1, 2)	20	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(-2,3, 4, -5)	24	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(4, -5, 6,8)	270	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(-5,2, -1,1)	19	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(7, -11,12,19)	1317	(0.7937,0.7071, 0.5297,0.8409)	-0.25
(-5,9, -2,0)	101	(0.0000,1.0000, 0.0000,1.0000)	0.00

fmincon			
x_0	iterations	\hat{x}	$f(\hat{x})$
(0.8,0.8,0.8, 0.8)	6	(0.7937,0.7071, 0.5297,0.8409)	-0.25
(7, -3,1, 2)	15	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(-2,3, 4, -5)	20	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(4, -5, 6,8)	27	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(-5,2, -1,1)	13	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(7, -11,12,19)	32	(0.7937,0.7071, -0.5297, -0.8409)	-0.25
(-5,9, -2,0)	31	(0.0000,1.0000, 0.0000,1.0000)	0.00

Στο παράδειγμα αυτό παρουσιάζεται ένα σύνθετο πρόβλημα 4 μεταβλητών με 3 δύσκολους μη γραμμικούς περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 .

Για το αρχικό σημείο $x^0 = (-5, 9, -2, 0)$ οι συναρτήσεις επιστρέφουν ένα σημείο \hat{x} το οποίο ικανοποιεί τις αναγκαίες συνθήκες 1^{ης} τάξης όμως ισχύει

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 = 2.1 \times 10^{-6} > 0$$

άρα το \hat{x} είναι ένα αυστηρό τοπικό ελάχιστο.

Παράδειγμα 15

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^5} \{ (x_1 - x_2)^2 + (x_2 - x_3)^3 + (x_3 - x_4)^4 + (x_4 - x_5)^4 \}$$

$$\text{subject to } x_1 + x_2^2 + x_3^3 - 3 = 0$$

$$x_2 - x_3^2 + x_4 - 1 = 0$$

$$x_1 x_5 - 1 = 0$$

- $n = 5, m = 3$
- Αναλυτική λύση:

$$\hat{x} = (1, 1, 1, 1, 1), \hat{\lambda} = 0, f(\hat{x}) = 0$$

- Αλγοριθμική λύση:

SQP			
x^0	iterations	\hat{x}	$f(\hat{x})$
(2, -1, 2, -1, 3)	92	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(-1, 2, 9, 3, -3)	5604	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(9, 2, -3, 1, -7)	678	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(-3, 4, 5, 1, 0)	81	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(11, -2, 3, -4, 1)	609	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(2, -3, 4, 5, -1)	223	(1.0000, 1.0000, 1.0000, 1.0000, 1.0000)	0.0
(5, -6, 7, 8, -1)	125	(0.6670, 0.7261, 1.2155, 1.7513, 1.4771)	-0.03

fmincon			
x^0	iterations	\hat{x}	$f(\hat{x})$
(2, -1, 2, -1, 3)	45	(1.0001, 1.0001, 0.9999, 0.9998, 0.9999)	0.0
(-1, 2, 9, 3, -3)	32	(0.6670, 0.7261, 1.2155, 1.7513, 1.4771)	-0.03
(9, 2, -3, 1, -7)	-	-	-
(-3, 4, 5, 1, 0)	30	(1.0001, 1.0001, 0.9999, 0.9998, 0.9999)	0.0
(11, -2, 3, -4, 1)	53	(1.0001, 1.0001, 0.9999, 0.9998, 0.9999)	0.0
(2, -3, 4, 5, -1)	40	(0.6670, 0.7261, 1.2155, 1.7513, 1.4771)	-0.03
(5, -6, 7, 8, -1)	59	(0.6670, 0.7261, 1.2155, 1.7513, 1.4771)	-0.03

Στο παράδειγμα αυτό παρουσιάζεται ένα σύνθετο πρόβλημα 5 μεταβλητών με 3 δύσκολους μη γραμμικούς περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 .

Σε αυτό το πρόβλημα τα αποτελέσματα που επιστρέφει η fmincon δεν είναι αρκετά ακριβή αφού οι λύσεις παρουσιάζουν μικρή απόκλιση από την αναλυτική λύση, σε αντίθεση με την SQP που βρίσκει επιθυμητά αποτελέσματα με μεγάλη ακρίβεια.

Για το αρχικό σημείο $x^0 = (9, 2, -3, 1, -7)$ η συνάρτηση fmincon σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 678 επαναλήψεις.

Για το αρχικό σημείο $x^0 = (5, -6, 7, 8, -1)$ οι συναρτήσεις επιστρέφουν ένα σημείο \hat{x} το οποίο ικανοποιεί τις αναγκαίες συνθήκες 1^{ης} τάξης όμως ισχύει

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 = \begin{bmatrix} 1.9396 & -1.1569 \\ -1.1569 & 1.7140 \end{bmatrix} > 0$$

άρα το \hat{x} είναι ένα αυστηρό τοπικό ελάχιστο.

Παράδειγμα 16

(Αναφορά [8])

$$\min_{x \in \mathbb{R}^5} \{ (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \}$$

$$\text{subject to } x_1^2 x_4 + \sin(x_4 - x_5) - 1 = 0$$

$$x_2 + x_3^4 x_4^2 - 2 = 0$$

- $n = 5, m = 2$
- Αναλυτική λύση:

$$\hat{x} = (1, 1, 1, 1, 1), \hat{\lambda} = 0, f(\hat{x}) = 0$$

- Αλγοριθμική λύση

SQP			
x^0	iterations	\hat{x}	$f(\hat{x})$
(3,3,3,3,3)	11	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(0.7,1.8,0.5,2,2)	10	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(0,1,0, -1, -2)	12	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(-1, -3, -9,4,1)	21	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(2,3,0,4, -1)	23	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(5, -6,7,8, -1)	21	(-0.8655, -0.8422, 1.2176,1.1371, 0.9883)	0.07

fmincon			
x^0	iterations	\hat{x}	$f(\hat{x})$
(3,3,3,3,3)	11	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(0.7,1.8,0.5,2,2)	8	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(0,1,0, -1, -2)	15	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(-1, -3, -9,4,1)	-	-	-
(2,3,0,4, -1)	15	(1.0000,1.0000, 1.0000,1.0000, 1.0000)	0.0
(5, -6,7,8, -1)	14	(-0.8655, -0.8422, 1.2176,1.1371, 0.9883)	0.07

Στο παράδειγμα αυτό παρουσιάζεται ένα σύνθετο πρόβλημα 5 μεταβλητών με 2 δύσκολους μη γραμμικούς περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 .

Για το αρχικό σημείο $x^0 = (-1, -3, -9, 4, 1)$ η συνάρτηση fmincon σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 21 επαναλήψεις.

Για το αρχικό σημείο $x^0 = (5, -6, 7, 8, -1)$ οι συναρτήσεις επιστρέφουν ένα σημείο \hat{x} το οποίο ικανοποιεί τις αναγκαίες συνθήκες 1^{ης} τάξης όμως ισχύει

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 = \begin{bmatrix} 0.7766 & 0.3388 & -0.7661 \\ 0.3388 & 2.4361 & -0.2535 \\ -0.7661 & -0.2535 & 1.8929 \end{bmatrix} > 0$$

άρα το \hat{x} είναι ένα αυστηρό τοπικό ελάχιστο.

Παράδειγμα 17

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^5} \{ e^{x_1 x_2 x_3 x_4 x_5} \}$$

$$\text{subject to } x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0$$

$$x_2 x_3 - 5 x_4 x_5 = 0$$

$$x_1^3 + x_2^3 + 1 = 0$$

- $n = 5, m = 3$
- Αναλυτική λύση:

$$\hat{x} = (-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)$$

$$\hat{\lambda} = (0.0402, -0.0380, 0.0052), f(\hat{x}) = 0.0539$$

- Αλγοριθμική λύση:

SQP			
x^0	iter	\hat{x}	$f(\hat{x})$
(-2,2,2, -1, -1)	17	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-2,1.5,2, -1, -1)	19	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-1,0,1, -2, -3)	22	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-5,1, -3,1,0)	95	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-2,0,1, -10, -5)	36	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(2,3, -1,9,2)	78	(-0.6991, -0.8700, -2.7899, 0.6967, 0.6967)	0.4389

fmincon			
x^0	iter	\hat{x}	$f(\hat{x})$
(-2,2,2, -1, -1)	7	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-2,1.5,2, -1, -1)	6	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-1,0,1, -2, -3)	17	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-5,1, -3,1,0)	14	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(-2,0,1, -10, -5)	19	(-1.7171, 1.5957, 1.8272, -0.7636, -0.7636)	0.0539
(2,3, -1,9,2)	6	(-0.6991, -0.8700, -2.7899, 0.6967, 0.6967)	0.4389

Στο παράδειγμα αυτό παρουσιάζεται ένα σύνθετο πρόβλημα 5 μεταβλητών με 3 δύσκολους μη γραμμικούς περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 .

Για το αρχικό σημείο $x^0 = (2,3, -1,9,2)$ οι συναρτήσεις επιστρέφουν ένα σημείο \hat{x} το οποίο ικανοποιεί τις αναγκαίες συνθήκες 1^{ης} τάξης όμως ισχύει

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 = \begin{bmatrix} 1.5245 & 0.6644 \\ 0.6644 & 1.4683 \end{bmatrix} > 0$$

άρα το \hat{x} είναι ένα αυστηρό τοπικό ελάχιστο.

Παράδειγμα 18

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^5} \{ (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4 \}$$

$$\text{subject to } x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0$$

$$x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0$$

$$x_1 x_5 - 2 = 0$$

- $n = 5, m = 3$
- Αναλυτική λύση:

$$\hat{x} = (1.1911, 1.3626, 1.4728, 1.6350, 1.6791)$$

$$\hat{\lambda} = (-0.0388, -0.0167, -0.0003), f(\hat{x}) = 0.0788$$

- Αλγοριθμική λύση

SQP			
x^0	iter	\hat{x}	$f(\hat{x})$
(-1,1, -1,1, -1)	19	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(1,1,0,2,1)	16	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(0,1,0,1,1)	15	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(2,2,2,2,2)	12	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(-3,4,5, -1,6)	29	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(-4,5, -1,8,9)	20	(-1.2468, 2.4222, 1.1750, -0.2132, -1.6041)	27.5

fmincon			
x^0	iter	\hat{x}	$f(\hat{x})$
(-1,1, -1,1, -1)	44	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(1,1,0,2,1)	12	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(0,1,0,1,1)	13	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(2,2,2,2,2)	12	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(-3,4,5, -1,6)	26	(1.1911, 1.3626, 1.4728, 1.6350, 1.6791)	0.0788
(-4,5, -1,8,9)	16	(-1.2468, 2.4222, 1.1750, -0.2132, -1.6041)	27.5

Στο παράδειγμα αυτό παρουσιάζεται ένα σύνθετο πρόβλημα 5 μεταβλητών με 3 δύσκολους μη γραμμικούς περιορισμούς. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 .

Για το αρχικό σημείο $x^0 = (-4,5, -1,8,9)$ οι συναρτήσεις επιστρέφουν ένα σημείο \hat{x} το οποίο ικανοποιεί τις αναγκαίες συνθήκες 1^{ης} τάξης όμως ισχύει

$$Q_2^T \nabla_{xx}^2 L(\hat{x}, \hat{\lambda}) Q_2 = \begin{bmatrix} 30.7451 & -18.0436 \\ -18.0436 & 14.3481 \end{bmatrix} > 0$$

άρα το \hat{x} είναι ένα αυστηρό τοπικό ελάχιστο.

Παράδειγμα 19

(Αναφορά [13])

$$\min_{x \in \mathbb{R}^{10}} \left\{ \sum_{j=1}^{10} e^{x_j} \left[c_j + x_j - \ln \left(\sum_{i=1}^{10} e^{x_i} \right) \right] \right\}$$

$$\text{subject to } e^{x_1} + 2e^{x_2} + 2e^{x_3} + e^{x_6} + e^{x_{10}} - 2 = 0$$

$$e^{x_4} + 2e^{x_5} + e^{x_6} + e^{x_7} - 1 = 0$$

$$e^{x_3} + e^{x_7} + e^{x_8} + 2e^{x_9} + e^{x_{10}} - 1 = 0$$

Οι σταθερές c_j έχουν τιμή με βάση τον πίνακα

j	c_j	j	c_j
1	-6.089	6	-14.986
2	-17.164	7	-24.100
3	-34.054	8	-10.708
4	-5.914	9	-26.662
5	-24.721	10	-22.179

- $n = 10, m = 3$
- Αναλυτική λύση:

$$\hat{x} = \begin{pmatrix} -3.20, -1.91, -0.24, -6.56, -0.72, \\ -7.3, -3.60, -4.02, -3.29, -2.33 \end{pmatrix}$$

$$\hat{\lambda} = (9.7851, 12.9689, 15.2221), f(\hat{x}) = -47.7611$$

Στο παράδειγμα αυτό παρουσιάζεται ένα εξαιρετικά σύνθετο πρόβλημα 10 μεταβλητών με 3 δύσκολους μη γραμμικούς περιορισμούς. Για τα πρόβλημα αυτό δεν έχουν χρησιμοποιηθεί πίνακες για την παρουσίαση των αποτελεσμάτων των συναρτήσεων επειδή η τάξη του συστήματος είναι μεγάλη και οι πίνακες δε θα είναι ευανάγνωστοι.

Για αρχικά σημεία

$$x^0 = \{ (-2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3), \\ (0, 0, 0, 0, 0, 0, 0, 0, 0, 0), \\ (0, 1, -2, -3, 1, 0, -1, -2, 2, 3), \\ (0, 7, -3, -5, 2, -4, 0, 1, -7, 2), \\ (3, -1, 1, -1, 2, -5, 0, 1, -4, 0) \}$$

η συνάρτηση SQP επιστρέφει με ακρίβεια την επιθυμητή λύση \hat{x} σε $\{ 68, 69, 2601, 198, 139 \}$ επαναλήψεις αντίστοιχα για κάθε αρχικό σημείο x^0 .

Για αρχικό σημείο

$$x^0 = \{ (1, 1, 1, 1, 1, 1, 1, 1, 1, 1) \}$$

η SQP «κολλάει» και δεν επιστρέφει κανένα αποτέλεσμα για $\varepsilon = 10^{-16}$. Αλλάζοντας την τιμή αυτή σε $\varepsilon = 10^{-15}$ έχουμε επιθυμητή λύση σε 98 επαναλήψεις.

Η συνάρτηση fmincon επιστρέφει την επιθυμητή λύση \hat{x} μόνο για

$$x^0 = \{ (-2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3, -2.3), \\ (0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \}$$

Για όλα τα υπόλοιπα αρχικά σημεία που αναφέρθηκαν παραπάνω σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα.

Παράδειγμα 20

$$\min_{x \in \mathbb{R}^2} \left\{ -\frac{1}{(x_1 - 1)^2 + 1} - x_1 x_2 \right\}$$

subject to $x_2 = 0$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (1, 0), \hat{\lambda} = 1, f(\hat{x}) = -1$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (2,8)	14	(1.0000, 0.0000)
P2: (2,5)	14	(1.0000, 0.0000)
P3: (2,2)	10	(1.0000, 0.0000)
P4: (2, -2)	9	(1.0000, 0.0000)
(3, 8)	15	(1.0000, 0.0000)
(-7, -15)	15	(1.0000, 0.0000)
(-100, -100)	27	(1.0000, 0.0000)
(500, -10)	31	(1.0000, 0.0000)
(50,50)	25	(1.0000, 0.0000)
(2, 600)	30	(1.0000, 0.0000)

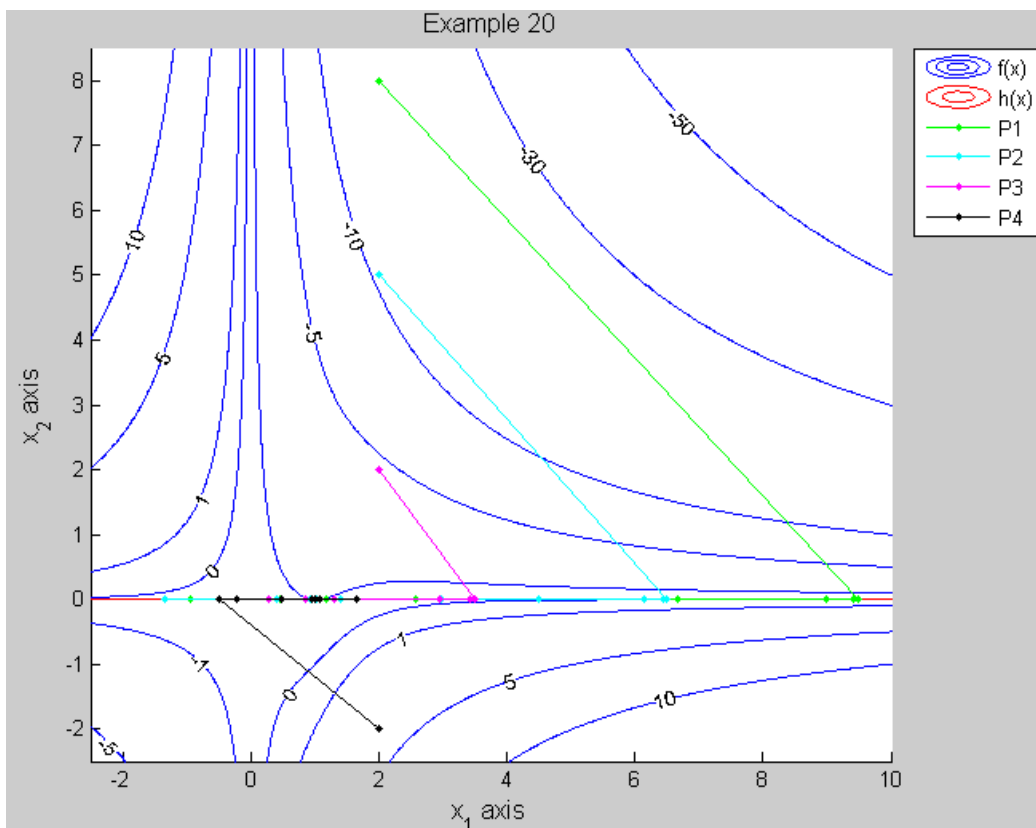
fmincon		
x^0	iterations	\hat{x}
P1: (2,8)	14	(1.0000, 0.0000)
P2: (2,5)	13	(1.0000, 0.0000)
P3: (2,2)	9	(1.0000, 0.0000)
P4: (2, -2)	8	(1.0000, 0.0000)
(3, 8)	15	(1.0000, 0.0000)
(-7, -15)	15	(1.0000, 0.0000)
(-100, -100)	1	(-200.0, 0.0000)
(500, -10)	1	(490.0, 0.0000)
(50,50)	1	(100.0, 0.0000)
(2, 600)	2	(601.5, 0.0000)

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 2 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

Για τα αρχικά σημεία $x_0 = (500, -10)$, $x_0 = (-100, 100)$ και $x_0 = (50, 50)$ η συνάρτηση `fmincon` δεν επιστρέφει επιθυμητή λύση σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 31,27 και 15 επαναλήψεις αντίστοιχα.

Για το αρχικό σημείο $x_0 = (2, 600)$ οι συναρτήσεις δεν επιστρέφουν επιθυμητό αποτέλεσμα. Αλλάζοντας την τιμή σε $\epsilon = 10^{-18}$ έχουμε επιθυμητό αποτέλεσμα από την SQP σε 30 επαναλήψεις.

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Παράδειγμα 21

$$\min_{x \in \mathbb{R}^2} \{ (x_1 - 1)^2 - x_1 x_2 e^{x_1} \}$$

subject to $x_2 = 0$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (1, 0), \hat{\lambda} = 2.7183, f(\hat{x}) = 0$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (2, 2)	-	Not Converge
P2: (2, 1)	-	Not Converge
P3: (2, 0.15)	5	(1.0000, 0.0000)
P4: (2, -1)	4	(1.0000, 0.0000)
P5: (2, -2)	4	(1.0000, 0.0000)
(-40, 130)	3	(1.0000, 0.0000)
(-50, 50)	3	(1.0000, 0.0000)
(4, 3)	-	Not Converge
(3, 1)	-	Not Converge
(1, 1)	-	Not Converge

fmincon		
x^0	iterations	\hat{x}
P1: (2, 2)	-	-
P2: (2, 1)	-	-
P3: (2, 0.15)	-	-
P4: (2, -1)	5	(1.0000, 0.0000)
P5: (2, -2)	5	(1.0000, 0.0000)
(-40, 130)	4	(1.0000, 0.0000)
(-50, 50)	4	(1.0000, 0.0000)
(4, 3)	-	-
(3, 1)	-	-
(1, 1)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 2 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

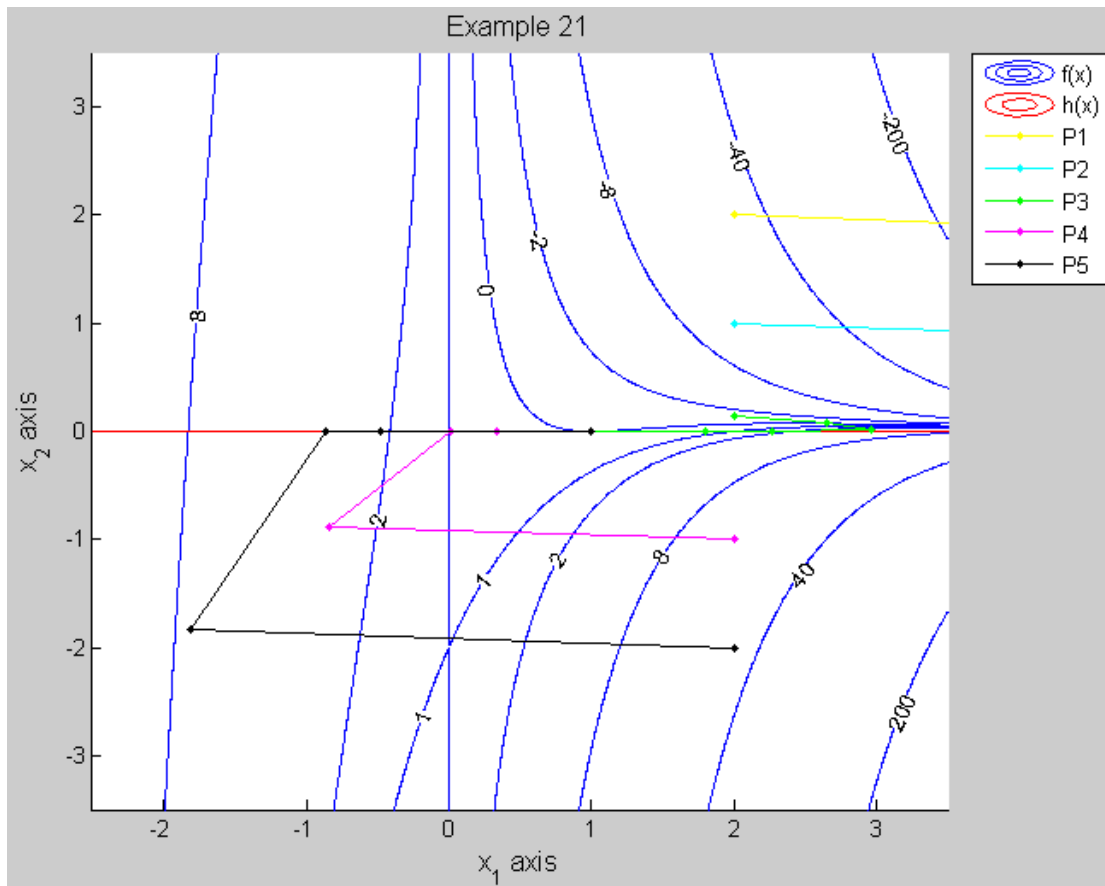
Για το αρχικό σημείο $x^0 = (2, t)$ με $t \geq 0.15$ η SQP αποκλίνει καθώς η τιμή του x^k γίνεται πολύ μεγάλη. Αλλάζοντας την τιμή της παραμέτρου ποινής σε $c = 10^4$ έχουμε λύση σε 3 επαναλήψεις για κάθε t . Η `fmincon` για το ίδιο σημεία σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα.

Για τα αρχικά σημεία $x^0 = (2, 0.15)$ η συνάρτηση `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 5 επαναλήψεις.

Για τα αρχικά σημεία $x^0 = (4, 3)$ και $x^0 = (3, 1)$ η SQP αποκλίνει καθώς όπως φαίνεται στον παρακάτω πίνακα η τιμή του x^k ήδη στην 2^η επανάληψη γίνεται πολύ μεγάλη και γίνονται πράξεις με τιμές εκτός των ορίων του MATLAB. Έγιναν αρκετές δοκιμές για τις σταθερές c και ϵ αλλά δεν υπήρχε επιθυμητή λύση. Προφανώς και η `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα για τα σημεία αυτά.

x^0	k	x^k	c^k
(4, 3)	0	(4.0000, 3.0000)	10
			223.4
	1	(573.1, 0.9000)	223.4
			2.2×10^{252}
(3, 1)	2	$(12.7 \times 10^{153}, 0.9000)$	2.2×10^{252}
	0	(3.0000, 1.0000)	10
			63.3
	1	(56.4, 0.3000)	63.3
		1.1×10^{27}	
	2	$(290.9 \times 10^{24}, 0.9000)$	1.1×10^{27}

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Παράδειγμα 22

$$\min_{x \in \mathbb{R}^2} \{ (x_1 - 1)^2 - x_1 x_2 e^{x_1} \}$$

subject to $x_2 = 0$

$$x_1^2 + x_2^2 - 16 \leq 0$$

Με βάση τη μορφή (P2') που αναφέρθηκε στο δεύτερο κεφάλαιο το πρόβλημα αυτό είναι ισοδύναμο με το

$$\min_{x \in \mathbb{R}^3} \{ (x_1 - 1)^2 - x_1 x_2 e^{x_1} \}$$

subject to $x_2 = 0$

$$x_1^2 + x_2^2 - 16 + x_3^2 = 0$$

- $n = 3, m = 2$
- Αναλυτική λύση για το αρχικό πρόβλημα:

$$\hat{x} = (1, 0), \hat{\lambda} = (2.7183, 0), f(\hat{x}) = 0$$

- Αλγοριθμική λύση για το ισοδύναμο πρόβλημα:

SQP		
x^0	iterations	\hat{x}
P1: (2, 2, 0.1)	444	(1.000, 0.0000, 3.8730)
P2: (2, 1, 0.2)	224	(1.000, 0.0000, 3.8730)
P3: (2, 0.15, 10)	7	(1.000, 0.0000, 3.8730)
P4: (2, -1, 5)	13	(1.000, 0.0000, 3.8730)
P5: (2, -2, 3)	14	(1.000, 0.0000, 3.8730)
(3, 1, 0.3)	162	(1.000, 0.0000, 3.8730)
(4, 3, 0.3)	16	(1.000, 0.0000, 3.8730)
(1, 1, 10)	10	(1.000, 0.0000, 3.8730)
(2, 3, 1)	-	Not Converge

fmincon		
x^0	iterations	\hat{x}
P1: (2, 2, 0.1)	-	-
P2: (2, 1, 0.2)	15	(1.000, 0.0000, 3.8730)
P3: (2, 0.15, 10)	7	(1.000, 0.0000, 3.8730)
P4: (2, -1, 5)	8	(1.000, 0.0000, 3.8730)
P5: (2, -2, 3)	9	(1.000, 0.0000, 3.8730)
(3, 1, 0.3)	-	-
(4, 3, 0.3)	-	-
(1, 1, 10)	9	(1.000, 0.0000, 3.8730)
(2, 3, 1)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα με ανισοτικούς περιορισμούς που το ανάγουμε σε πρόβλημα με ισοτικούς με βάση τη μορφή (P2') που είπαμε στο κεφάλαιο 2. Το ισοδύναμο πρόβλημα είναι 3 μεταβλητών με 2 περιορισμούς. Είναι το παράδειγμα 21 με την προσθήκη ενός ανισοτικού περιορισμού. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

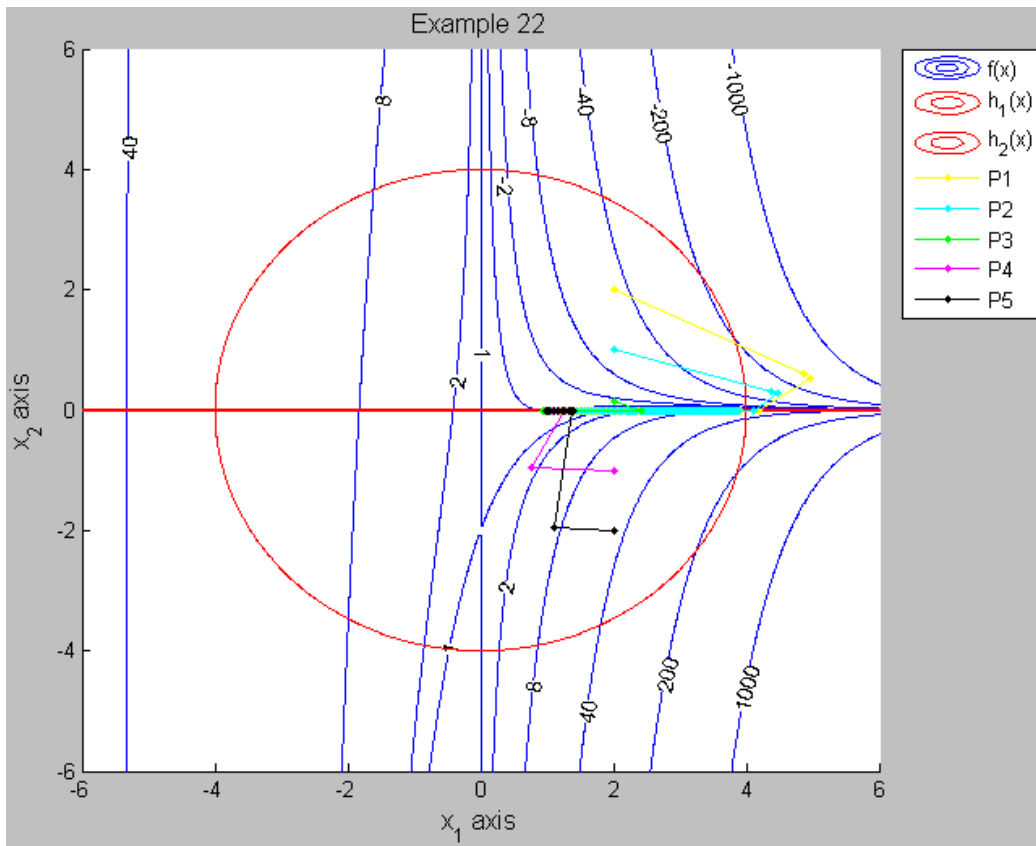
Για τα αρχικά σημεία $x^0 = (2,2)$, $x^0 = (2,1)$, $x^0 = (4,3)$, $x^0 = (1,1)$ και $x^0 = (3,1)$ του παραδείγματος 21 η SQP απέκλινε. Τώρα με την προσθήκη ανισοτικού περιορισμού η SQP επιστρέφει επιθυμητή λύση.

Για τα αρχικά σημεία $x^0 = (2,2,0.1)$, $x^0 = (3,1,0.3)$ και $x^0 = (4,3,0.3)$ η συνάρτηση `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα, σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση σε 444, 162 και 16 επαναλήψεις αντίστοιχα.

Για το αρχικό σημείο $x^0 = (2,3,1)$ η SQP αποκλίνει καθώς όπως φαίνεται στον παρακάτω πίνακα η τιμή του x^k ήδη στην 2^η επανάληψη γίνεται πολύ μεγάλη και γίνονται πράξεις με τιμές εκτός των ορίων του MATLAB. Έγιναν αρκετές δοκιμές για τις σταθερές c και ε αλλά δεν υπήρχε επιθυμητή λύση. Προφανώς και η `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα για τα σημεία αυτά.

x^0	k	x^k	c^k
(2,3,1)	0	(2.0000, 3.0000, 1.0000)	10
	1	(13.8, 0.9000, -15.6)	10
			192.5×10^3
	2	$(722.9 \times 10^3, 93.1 \times 10^{-12}, 638.4 \times 10^3)$	192.5×10^3

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Παράδειγμα 23

$$\min_{x \in \mathbb{R}^2} \left\{ \frac{1}{2}x_1^2 - \frac{1}{3}x_2x_1^3 \right\}$$

subject to $x_2 = 0$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (0, 0), \hat{\lambda} = 0, f(\hat{x}) = 0$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (-2, 10)	-	Not Converge
P2: (-2,5)	8	(0.0000, 0.0000)
P3: (-2, 1)	3	(0.0000, 0.0000)
P4: (-2, -4)	23	(0.0000, 0.0000)
(3, 3)	19	(0.0000, 0.0000)
(-11, 9)	17	(0.0000, 0.0000)
(19, 9)	30	(0.0000, 0.0000)
(-12, -12)	16	(0.0000, 0.0000)
(-3, -2)	-	Not Converge
(-40,130)	-	Not Converge
(6, 12)	-	Not Converge
(50, 50)	-	Not Converge

fmincon		
x^0	iterations	\hat{x}
P1: (-2, 10)	-	-
P2: (-2,5)	-	-
P3: (-2, 1)	5	(0.0000, 0.0000)
P4: (-2, -4)	-	-
(3, 3)	-	-
(-11, 9)	-	-
(19, 9)	-	-
(-12, -12)	-	-
(-3, -2)	-	-
(-40,130)	-	-
(6, 12)	-	-
(50, 50)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 2 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

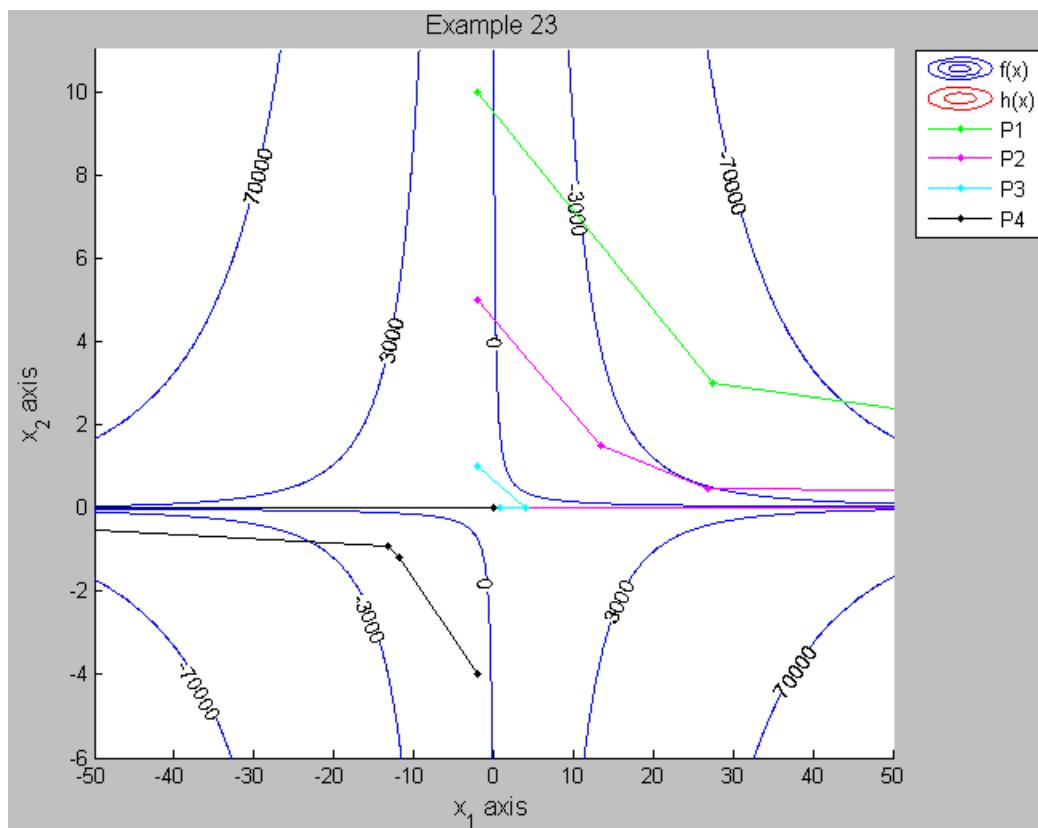
Σε αυτό το πρόβλημα η συνάρτηση fmincon επιστρέφει επιθυμητό αποτέλεσμα μόνο για το αρχικό σημείο $x^0 = (-2,1)$. Έγιναν δοκιμές για αρκετά αρχικά σημεία και για όλα η συνάρτηση σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα.

Για τα αρχικά σημεία

$$x^0 = (-2,10), x^0 = (-3,-2), x^0 = (-40,130), x^0 = (6,12), x^0 = (50,50)$$

καθώς και για πολλά ακόμη που δεν έχουν καταγραφεί στους πίνακες η SQP αποκλίνει καθώς η τιμή του x^k γίνεται πολύ μεγάλη. Αλλάζοντας την τιμή της παραμέτρου ποινής σε $c = 10^{10}$ έχουμε λύση για κάθε αρχικό σημείο. Εγιναν δοκιμές για αρκετά αρχικά σημεία και για όλα η SQP επιστρέφει επιθυμητή λύση για την καινούρια τιμή της παραμέτρου c .

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Παράδειγμα 24

$$\min_{x \in \mathbb{R}^2} \{ (x_1 - 0.1)^2 - 100x_1^2x_2e^{2x_1^2} + 0.98 \}$$

subject to $x_2 = 0$

- $n = 2, m = 1$
- Αναλυτική λύση:

$$\hat{x} = (0.1, 0), \hat{\lambda} = 1.0202, f(\hat{x}) = 0.98$$

- Αλγοριθμική λύση:

SQP		
x^0	iterations	\hat{x}
P1: (1,3)	-	Not Converge
P2: (1,1.5)	-	Not Converge
P3: (1,0.0001)	3	(0.1000, 0.0000)
P4: (1, -1.5)	8	(0.1000, 0.0000)
P5: (1, -3)	9	(0.1000, 0.0000)
(-1, -1)	8	(0.1000, 0.0000)
(0, 1)	4	(0.1000, 0.0000)
(-2, -2)	15	(0.1000, 0.0000)
(-10,10)	-	Not Converge
(-5,5)	-	Not Converge

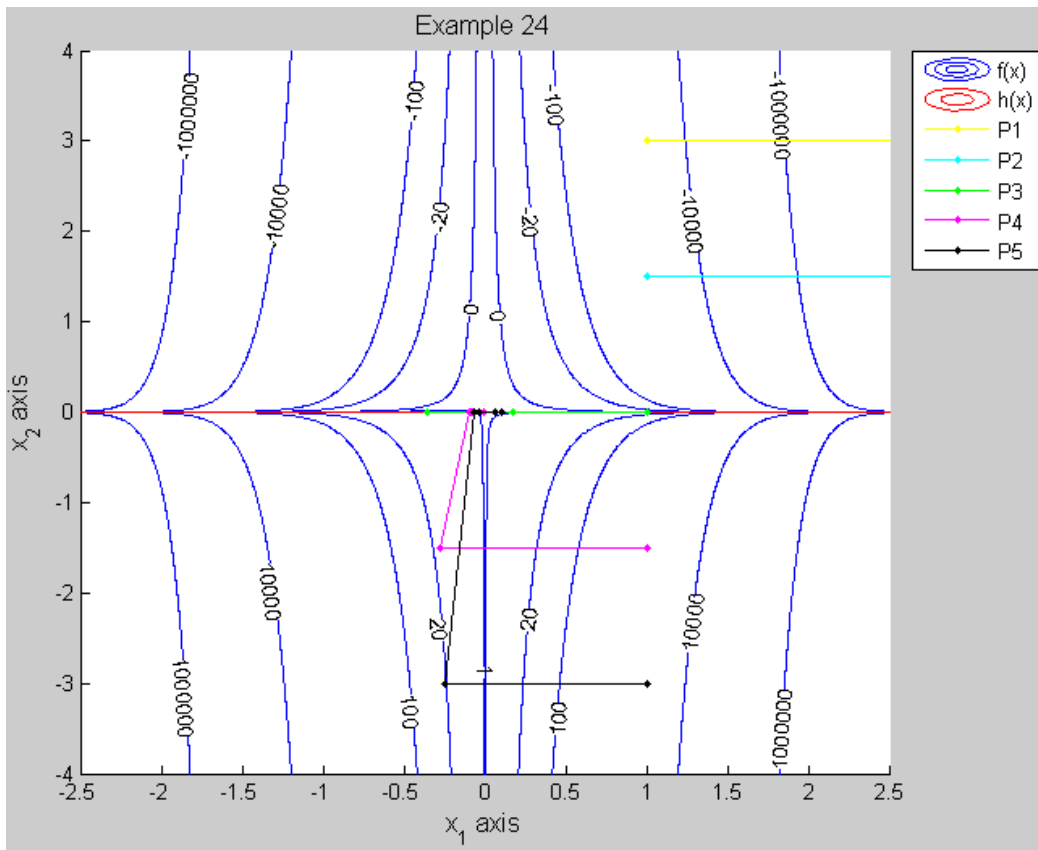
fmincon		
x^0	iterations	\hat{x}
P1: (1,3)	-	Not Converge
P2: (1,1.5)	-	Not Converge
P3: (1,0.0001)	4	(0.1000, 0.0000)
P4: (1, -1.5)	9	(0.1000, 0.0000)
P5: (1, -3)	9	(0.1000, 0.0000)
(-1, -1)	9	(0.1000, 0.0000)
(0, 1)	4	(0.1000, 0.0000)
(-2, -2)	12	(0.1000, 0.0000)
(-10,10)	-	-
(-5,5)	-	-

Στο παράδειγμα αυτό παρουσιάζεται ένα πρόβλημα 2 μεταβλητών με 1 γραμμικό περιορισμό. Η συνάρτηση SQP επιστρέφει επιθυμητή λύση για αρκετά αρχικά σημεία x^0 σε λίγες επαναλήψεις.

Για τα αρχικά σημεία $x^0 = (1,3)$, $x^0 = (1,1.5)$, $x^0 = (-10,10)$, $x^0 = (-5,5)$ η SQP αποκλίνει καθώς όπως φαίνεται στον παρακάτω πίνακα η τιμή του x^k ήδη στην 1^η επανάληψη γίνεται πολύ μεγάλη και γίνονται πράξεις με τιμές εκτός των ορίων του MATLAB. Έγιναν αρκετές δοκιμές για τις σταθερές c και ε αλλά δεν υπήρχε επιθυμητή λύση. Προφανώς και η `fmincon` σταματάει χωρίς να επιστρέφει κάποιο αποτέλεσμα για τα σημεία αυτά.

x^0	k	x^k	c^k
(1,3)	0	(1.0, 3.0)	10
			744.1
	1	$(9.3 \times 10^3, 0.9)$	744.1
(1,1.5)	0	(1.0, 1.5)	10
			742.4
	1	$(4.7 \times 10^3, 0.5)$	742.4
(-10,10)	0	(-10.0, 10.0)	10
			7.2×10^{90}
	1	$(-2.0 \times 10^{93}, 3.0)$	7.2×10^{90}
(-5,5)	0	(-5.0, 5.0)	10
			13.0×10^{24}
	1	$(-925.0 \times 10^{24}, 1.5)$	13.0×10^{24}

Στο παρακάτω σχήμα φαίνεται η γραφική παράσταση του προβλήματος.



Συμπεράσματα

Ο αλγόριθμος SQP που παρουσιάστηκε και προγραμματίστηκε είναι πολύ σταθερός και λύνει πάρα πολλά προβλήματα της μορφής (P1) για αρκετά αρχικά σημεία x^0 . Από απλά προβλήματα με γραμμικούς περιορισμούς και λίγες μεταβλητές έως πολύ πολύπλοκα προβλήματα με αρκετές μεταβλητές και δύσκολους περιορισμούς. Ο προγραμματισμός του έγινε με πολλή προσοχή και λεπτομέρεια για να καλύψει όσο το δυνατόν περισσότερες ιδιομορφίες που μπορεί να παρουσιάσει το δοθέν πρόβλημα.

Για τα πρώτα 19 προβλήματα του προηγούμενου κεφαλαίου (τα οποία αντλήθηκαν από την βιβλιογραφία) ο αλγόριθμος συγκλίνει στην αναλυτική λύση με μεγάλη ακρίβεια για τα περισσότερα αρχικά σημεία. Για τα προβλήματα 20 έως 24 (προβλήματα της έμπνευσης του επιβλέποντα της εργασίας Ν.Μαράτου) ο αλγόριθμος αποκλίνει για πολλά αρχικά σημεία από τις πρώτες κιόλας επαναλήψεις. Εκεί παρατηρείται ότι όταν το αρχικό σημείο είναι «αρκετά» απομακρυσμένο από τον περιορισμό ο αλγόριθμος αποκλίνει καθώς η παράμετρος ποινής τείνει στο άπειρο ($c \rightarrow \infty$). Ο λόγος είναι ότι η αντικειμενική συνάρτηση κόστους μειώνεται απότομα με μεγάλο ρυθμό. Η επίδραση του κομματιού της κατεύθυνσης έρευνας d^k που τείνει να επιστρέψει την f στον περιορισμό ακυρώνεται από την μεγάλη κλίση της. Η τροχιά που διαγράφει ο αλγόριθμος μπορεί να απομακρύνεται και να επιστρέφει στο επιθυμητό σημείο, ή, για άλλα αρχικά σημεία μακριά από τον περιορισμό, να απομακρύνεται και να μην επιστρέφει στο επιθυμητό σημείο. Κάποιες φορές με κατάλληλη τροποποίηση της σταθεράς ποινής c ο αλγόριθμος μπορεί να οδηγηθεί σε σύγκλιση, άλλες φορές η σύγκλιση είναι αδύνατη. Σε άλλα προβλήματα, όχι σε αυτά που αποκλίνουν, με τροποποίηση της σταθεράς ε για αλλαγή του κριτηρίου τερματισμού μπορεί ο αλγόριθμος να πετύχει επιθυμητή λύση με μεγάλη ακρίβεια.

Συνολικά προκύπτει ότι η συνάρτηση SQP είναι πιο αξιόπιστη από την συνάρτηση `fmincon` του MATLAB όπως φαίνεται ξεκάθαρα από τους πίνακες με την σύγκριση των αποτελεσμάτων που επιστρέφουν. Τα αποτελέσματα που επιστρέφει η `fmincon` πολλές φορές δεν έχουν μεγάλη αριθμητική ακρίβεια σε αντίθεση με την SQP που επιστρέφει επιθυμητή λύση με μεγάλη ακρίβεια. Ένα άλλο πλεονέκτημα της SQP είναι ότι για ιδιόμορφα προβλήματα ο χρήστης μπορεί να επιτύχει επιθυμητή λύση με κατάλληλες τροποποιήσεις των παραμέτρων (κυρίως των παραμέτρων c και ϵ) κάτι που δεν μπορεί να γίνει στην `fmincon` καθώς ο κώδικας της είναι δυσνόητος, πολύπλοκος και δύσκολα τροποποιήσιμος για απλούς χρήστες του MATLAB (μη προγραμματιστές). Συνεπώς, όπως φάνηκε από τα παραδείγματα και για τους παραπάνω λόγους, η SQP μπορεί να αντικαταστήσει την `fmincon`.

Μπορεί κάλλιστα όποιος θέλει να λύσει προβλήματα βελτιστοποίησης με ισοτικούς περιορισμούς απλά να πάρει τον κώδικα από την παρούσα εργασία (υπάρχει στο παράρτημα) και να λύσει απλά και εύκολα τα προβλήματα του αρκεί να έχει στις αναφορές του την εργασία αυτή.

Παράρτημα

Κώδικας MATLAB και Οδηγίες

Στο παράρτημα θα παρουσιαστεί ο κώδικας που χρησιμοποιήθηκε στα πλαίσια της παρούσας εργασίας καθώς και μερικές χρήσιμες οδηγίες για το MATLAB. Επειδή τα προγράμματα και οι εντολές που πρέπει να δοθούν στο MATLAB για να τρέξουν δεν είναι προφανή είναι χρήσιμο να δοθούν αναλυτικές οδηγίες για όλη τη διαδικασία που ακολουθήθηκε. Τα προγράμματα υλοποιήθηκαν με το MATLAB 7.13.0.564 (R2011b) που σημαίνει ότι τυχόν εκτέλεσή τους με παλαιότερες εκδόσεις μπορεί να έχει ανεπιθύμητα αποτελέσματα. Θα καταγραφεί όλη η διαδικασία που πρέπει να ακολουθήσει κανείς για να βρει τη λύση στο MATLAB με τις 2 συναρτήσεις και τη γραφική παράσταση όπως ακριβώς παρουσιάστηκε στο κεφάλαιο 5.

Αρχικά ανοίγει το MATLAB και ορίζει ως Current Folder το φάκελο που θέλει να εργαστεί και να αποθηκεύσει τα αρχεία του. Έστω ότι θέλει να λύσει το ακόλουθο πρόβλημα (παράδειγμα 3, κεφάλαιο 5)

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \{ x_1^2 + x_2^2 \} \\ \text{subject to } (x_1 + 1)^2 + x_2^2 - 4 = 0 \end{aligned}$$

Πρέπει να γράψει την αντικειμενική συνάρτηση κόστους, τους ισοτικούς περιορισμούς καθώς και τις κλίσεις τους σε ξεχωριστά αρχεία συναρτήσεων τα οποία θα είναι είσοδοι για την συνάρτηση SQP. Πηγαίνει στη γραμμή επιλογών File > New > Function για κάθε συνάρτηση που θέλει να φτιάξει. Γράφει στον Editor που ανοίγει

```
function u = f(x)
% objective function
u = x(1)^2+x(2)^2; % Example 3
```

την αποθηκεύει με File > Save As... > f.m

```
function v = gradf(x)
% gradient of the objective function
v = [2*x(1) 2*x(2)]'; % Example 3
```

την αποθηκεύει με File > Save As... > gradf.m

```
function y = h(x)
% equality constraint
y = (x(1)+1)^2+x(2)^2-4; % Example 3
```

την αποθηκεύει με File > Save As... > h.m

```
function w = gradh(x)
% gradient of the equality constraint
w = [2*(x(1)+1) 2*x(2)]'; % Example 3
```

την αποθηκεύει με File > Save As... > gradh.m

Για την βασική συνάρτηση SQP της εργασίας γράφει

```
function [x, l, fmin, iter]...
    = SQP(obj, grad_obj, constr, grad_constr, x_0)
% Sequential Quadratic Programming Algorithm
% Tested with Matlab 7.13 (R2011b)
%
% Inputs
% obj: objective function in file f.m
% grad_obj: gradient of the objective function in
% file gradf.m
```

```

% constr: equality constraint in file h.m
% grad_constr: gradient of the constraint in file gradh.m
% x_0: starting point, x_0 is a row vector
%
% Outputs
% x: optimal solution of the problem
% l: lagrange multipliers of the problem
% fmin: minimum value of the objective function
% iter: number of iterations for optimal solution
%
% Run the algorithm with the command:
% >> [x,l,fmin,iter] = SQP(@f, @gradf, @h, @gradh, [1 1])
%
% f, gradf, h, gradh can be specified
% as anonymous functions with @
%
% Run the algorithm with the command:
% >> [x,l,fmin,iter] = SQP(@(x) x(2), @(x) [0 1]',...
% @(x) x(1)^2+x(2)^2-1, @(x) [2*x(1) 2*x(2)]', [1 1])
global X
global C
global G
global T
format short
n = length(grad_obj(x_0));
m = length(constr(x_0));
% step 0
k = 0;
x_k = x_0';
Q = eye(n);
a = 0.1;
b = 0.7;
eps = 10^(-16);
epsilon = 10^(-4);

```

```

c = 10;
delta = 2;
ksi = 2;
t1 = obj(x_k);
t2 = grad_obj(x_k);
t3 = constr(x_k);
t4 = grad_constr(x_k);
[X] = [k x_k'];
[C] = [k c];
[G] = [];
[T] = [];
while true
if rank(t4) < m
    error('Not a regular point of the constraints')
end
% step 1
t = zeros(m); % t is temporary variable
T1 = [Q t4; t4' t];
T2 = [-t2; -t3];
sol = mldivide(T1, T2);
d = sol(1:n);
lamda = sol(n+1:n+m);
% step 1'
norma = norm(t3,1);
test = (c-ksi)*norma-(lamda')*t3;
if test < 0
    c = max(c+delta, ((lamda')*t3)/norma+ksi);
    [C] = [C; k c];
end
% step 2
theta = (t2')*d-c*norma;

% step 3
t = norm(d, 2)^2;

```

```

if t <= eps
    break
end
% step 4
s = 1;
gama = t1+c*norma;
while true
    t = x_k+s*d;
    t5 = obj(t);
    t7 = constr(t);
    gama_new = t5+c*norm(t7, 1);
    if gama_new - gama <= a*s*theta
        break
    end
    s = s*b;
end
% step 5
p = s*d;
x_k = t;
t6 = grad_obj(x_k);
t8 = grad_constr(x_k);
q = t6+t8*lamda-t2-t4*lamda;
ni = (p')*q;
w = (p')*Q*p;
if ni >= 0.2*w
    psi = 1;
else
    psi = (0.8*w)/(w-ni);
end
r = psi*q+(1-psi)*Q*p;
Q = Q - ((Q*p*(p')*Q)/w) + ((r*(r'))/((p')*r));
k = k+1;
t1 = t5;
t2 = t6;

```

```

t3 = t7;
t4 = t8;
[X] = [X; k x_k'];
[C] = [C; k c];
[G] = [G; k gama];
[T] = [T; k theta];
% Back to step 1
end
des = 1;
t = norm(t2+t4*lamda, 2)^2;
if t > epsilon
    des = 0;
end
t = norm(t3,2)^2;
if t > epsilon
    des = 0;
end
if rank(t4) < m
    des = 0;
end
if des
    fprintf('\nx is desirable point\n\n');
end
x = x_k;
l = lamda;
fmin = t1;
iter = k;

```

και αποθηκεύει με File > Save As... > SQP.m

Για να τρέξει τη συνάρτηση SQP δίνει στο Command Window την εντολή

```
>> [x,l,fmin,iter] = SQP(@f, @gradf, @h, @gradh, [1 1])
```

Αν κάποιος δε θέλει να έχει σε ξεχωριστά αρχεία τις συναρτήσεις `f`, `gradf`, `h`, `gradh` μπορεί να καλέσει τη συνάρτηση `SQP` με τη βοήθεια των `anonymous function` που διαθέτει το `MATLAB`. Δηλαδή για το παράδειγμα 3 αρκεί κάποιος να δώσει στο `Command Window` την εντολή

```
>> [x,l,fmin,iter] = SQP(@(x) x(1)^2+x(2)^2,...
    @(x) [2*x(1) 2*x(2)]', @(x) (x(1)+1)^2+x(2)^2-4,...
    @(x) [2*(x(1)+1) 2*x(2)]', [1 1])
```

Για να δει κανείς όλα τα ενδιάμεσα σημεία που βρίσκει ο `SQP` (χρήσιμο για τα προβλήματα που ο αλγόριθμος αποκλίνει) και τις αλλαγές της παραμέτρου ποινής `c` στην κάθε επανάληψη πρέπει να δώσει τις εντολές

```
>> global X;
>> global C;
>> X
>> C
```

Οι εντολές αυτές πρέπει να δοθούν αφού πρώτα έχει τρέξει η συνάρτηση `SQP` για να έχουν καταχωρηθεί τιμές στις μεταβλητές `X` και `C`. Οι `X` και `C` είναι πίνακες που έχουν σε κάθε γραμμή τις τιμές του τρέχοντος σημείου και της σταθεράς `C` σε κάθε επανάληψη του αλγορίθμου.

Η συνάρτηση `fmincon` υπάρχει εγκατεστημένη στο `MATLAB` (`Optimization Toolbox` [22]) και επιλύει προβλήματα της μορφής (P1). Για την αντικειμενική συνάρτηση κόστους χρησιμοποιεί την `f.m` που έχει ήδη γράψει. Για τους περιορισμούς γράφει τη συνάρτηση

```
function [cineq,ceq] = cons(x)
% cineq: inequality constraint
% ceq: equality constraint
cineq = [];
ceq = (x(1)+1)^2+x(2)^2-4; % Example 3
```

και αποθηκεύει με File > Save As... > cons.m

Για να τρέξει τη συνάρτηση fmincon δίνει στο Command Window την εντολή

```
>> options=optimset('Display','iter','Algorithm','sqp');  
  
>> [x,feval,flag,output] =...  
fmincon(@f,[1 1],[[],[],[],[],[],[],[],@cons,options)
```

Για παραδείγματα του κεφαλαίου 5 που έχουν γίνει γραφικές παραστάσεις λύσεων έχει χρησιμοποιηθεί η συνάρτηση contSQP. Για τη συνάρτηση αυτή γράφει

```
function [] = contSQP(min1, max1, n1, min2, max2, n2)  
% Plots the contours of the optimization problem  
%  
% min1: the minimum value of variable x1  
% max1: the maximum value of variable x1  
% n1: number of points in x1 axis  
% min2: the minimum value of variable x2  
% max2: the maximum value of variable x2  
% n2: number of points in x2 axis  
global X;  
hold on  
step1 = (max1-min1)/n1;  
x1 = min1:step1:max1;  
step2 = (max2-min2)/n2;  
x2 = min2:step2:max2;  
nn1 = size(x1);  
nn1 = nn1(2);  
nn2 = size(x2);  
nn2 = nn2(2);  
Z = zeros(nn1, nn2); % preallocation Z  
Z1 = zeros(nn1, nn2); % preallocation Z1  
for i = 1:nn1  
    x(1) = x1(i);
```



```

    for j = 1:nn2
        x(2) = x2(j);
        Z(i,j) = f(x);
        Z1(i,j) = h(x);
    end
end
v = [1 4.5 11 20 35]; % Example 3
vder = [0 0.001];
Z = Z';
Z1 = Z1';
[q1, q2] = contour(x1, x2, Z, v, '-b');
contour(x1, x2, Z1, vder, 'r');
clabel(q1, q2, 'manual');
SQP(@f, @gradf, @h, @gradh, [-3 4]);
plot(X(:, 2), X(:, 3), '.-g');
SQP(@f, @gradf, @h, @gradh, [-4 0.1]);
plot(X(:, 2), X(:, 3), '.-c');
SQP(@f, @gradf, @h, @gradh, [-4 -0.2]);
plot(X(:, 2), X(:, 3), '.-m');
SQP(@f, @gradf, @h, @gradh, [-3 -4]);
plot(X(:, 2), X(:, 3), '.-k');
title('Example 3', 'FontSize', 12);
xlabel('x_1 axis', 'FontSize', 12);
ylabel('x_2 axis', 'FontSize', 12);
legend('f(x)', 'h(x)', 'P1', 'P2', 'P3', 'P4', 'location',
'NorthEastOutside')
hold off

```

και αποθηκεύει με File > Save As... > contSQP.m

Για να τρέξει τη συνάρτηση contRQP δίνει στο Command Window την εντολή

```
>> contSQP(-5, 3, 500, -5, 5, 500)
```

Για παραδείγματα του κεφαλαίου 5 που έχουν έχει υπολογιστεί η μήτρα Hessian της συνάρτησης Lagrange έχει χρησιμοποιηθεί το script Hessians. Για το παράδειγμα 17 γράφει

```
% Hessians Example 17
syms x1 x2 x3 x4 x5
g = exp(x1*x2*x3*x4*x5);
hg = hessian(g);
h1 = x1^2+x2^2+x3^2+x4^2+x5^2-10;
hh1 = hessian(h1);
h2 = x2*x3-5*x4*x5;
hh2 = [hessian(h2) zeros(4,1); zeros(1,5)];
h3 = x1^3+x2^3+1;
hh3 = [hessian(h3) zeros(2,3); zeros(3,5)];
L = hg+1(1)*hh1+1(2)*hh2+1(3)*hh3;
t = subs(L, [x1 x2 x3 x4 x5], x');
[Q,R] = qr(gradh(x));
n = 5; m = 3;
Q2 = Q(1:n, (m+1):n);
tetr_morfi = (Q2')*t*Q2;
idiotimes = eig(tetr_morfi);
fprintf('\ntetragoniki_morfi=\n\n');
disp(tetr_morfi);
fprintf('\nidiotimes = \n\n');
disp(idiotimes);
```

και αποθηκεύει με File > Save As... > Hessians.m

Για να τρέξει το script δίνει στο Command Window την εντολή

```
>> Hessians
```

Επειδή το script αυτό χρησιμοποιεί την λύση που επιστρέφει η συνάρτηση SQP και τους πολλαπλασιαστές Lagrange, πρέπει να τρέξει αφού πρώτα έχει καλεστεί η

συνάρτηση SQP και έχουν καταχωρηθεί τιμές στις μεταβλητές x και l . Την ίδια διαδικασία για τον υπολογισμό των μητρών Hessian ακολουθεί κανείς και για τα άλλα παραδείγματα τροποποιώντας κάθε φορά τις συναρτήσεις και τις παραμέτρους του προβλήματος.

Βιβλιογραφία – Αναφορές

- [1] N. Μαράτος, Τεχνικές Βελτιστοποίησης, ΕΜΠ, Αθήνα, 1990
- [2] D. Luenberger, Linear and Nonlinear Programming, Third Edition, Springer, 2008
- [3] M. Barbarosou , N. Maratos, "A Non-feasible Gradient Projection Recurrent Neural Network for Equality-Constrained Optimization Problems", IEEE Trans. on Neural Networks, Vol. 19, no. 10, pp. 1665-1677, 2008
- [4] Bonnans, Gilbert, Lemarechal, Sagastizabal, Numerical Optimization, Second Edition, Springer, 2006
- [5] E. Polak, Optimization-Algorithms and Consistent Approximations, Springer, 1997
- [6] D. Bertsekas, Nonlinear Programming, Athena Scientific, Belmont, Massachusetts, 1995
- [7] R. Fletcher, Practical Methods of Optimization, Second Edition, John Wiley and Sons, 1987
- [8] N.Maratos, "Exact penalty function algorithms for finite dimensional and control optimization problems", PhD Thesis, Imperial College of Science and Technology, London, 1978
- [9] M. Bazarra, C. Shetty, Nonlinear Programming, Theory and Algorithms, Third Edition, John Wiley and Sons, 2006
- [10] E. Polak, Computational Methods in Optimization, An Unified Approach, Academic Press, 1971
- [11] P.Venkataraman, Applied Optimization with Matlab Programming, John Wiley and Sons, 2002
- [12] M.Avriel, Nonlinear Programming, Analysis and Methods, Dover, 2003
- [13] K. Schittkowski, Test Examples for Nonlinear Programming Codes, <http://www.ai7.uni-bayreuth.de/> , 2009

- [14] K. Schittkowski, The Nonlinear Programming Method of Wilson, Han and Powell with an Augmented Lagrangian Type Line Search Function, Numerische Mathematik, Springer-Verlag, 1981
- [15] D. Mayne, E. Polak, A Superlinearly Convergent Algorithm for Constrained Optimization Problems, Mathematical Programming Study 16, 45-61, 1982, North-Holland Publishing Company
- [16] A. Bondarenko, D. Bortz, A Collection of Large-Scale Nonlinearly Constrained Optimization Test Problems, <http://www.mcs.anl.gov/~more/> , 2008
- [17] D. Shanno, An Example of Numerical Nonconvergence of a Variable-Metric Method, Journal of Optimization Theory and Applications, Vol.46, No.1, 1985
- [18] R. Chamberlain, Some Examples of Cycling in Variable Metric Methods for Constrained Minimization, Mathematical Programming 16, 378-383, 1979
- [19] S.Lucidi, Recursive Quadratic Programming Algorithm That Uses an Exact Augmented Lagrangian Function, Journal of Optimization Theory and Applications, Vol.67, No.2, 1990
- [20] M. Fukushima, A Successive Quadratic Programming Algorithm with global and superlinear convergence properties, Mathematical Programming 35 (1986) 253-264
- [21] J.M.Martinez and L.T. Santos, Some New Theoretical Results on Recursive Quadratic Programming Algorithms, Journal of Optimization Theory and Applications, 1998
- [22] An Exact Penalty Method For Smooth Equality Constrained Optimization With Application To Maximum Likelihood Estimation, W. Bergasma, T. Rapcsak, EURANDOM-report 2006-001, 2004
- [23] S.P.Han, A Globally Convergent Method for Nonlinear Programming, Journal Optimization Theory and Applications, Vol.22, p.297, 1977
- [24] Ν. Καδιανάκης, Σ. Καρανάσιος, Α. Φελούρης, Ανάλυση ΙΙ – Συναρτήσεις πολλών μεταβλητών, Έκδοση 7^η, Αθήνα 2007
- [25] Optimization Toolbox Matlab
- [26] Mathematical Optimization
http://en.wikipedia.org/wiki/Mathematical_optimization
- [27] Γ.Ε. Τσιφετάκης, Το πρόβλημα του τετράγωνου προγραμματισμού, Διπλωματική Εργασία, Σχολή ΗΜΜΥ ΕΜΠ, Αθήνα, Ιούλιος 2004

