

# Βελτιστοποίηση Συναρτήσεων με Άμεσες Μεθόδους

Μαρία – Νίκη Στάμου



## Περιεχόμενα

Πρόλογος	5
Κεφάλαιο 1. Εισαγωγή	7
1.1. Βελτιστοποίηση χωρίς παραγώγους	7
1.2. Φορμαλισμός του προβλήματος και συμβολισμός	8
1.3. Συζήτηση για την παραγωγίσιμη αντικειμενική συνάρτηση	9
Κεφάλαιο 2. Μέθοδοι αναζήτησης με παράγοντα σύνολα	15
2.1. Η μέθοδος Compass	15
2.2. Βασικές έννοιες και ορισμοί	19
2.3. Αναζήτηση με παράγοντα σύνολα	21
2.4. Αλγόριθμος της ΑΜΠΣ	23
2.5. Αναζήτηση με μοτίβα	24
2.6. Αναζήτηση με πλέγματα	25
2.7. Διερεύνηση σύγκλισης για την ΑΜΠΣ	29
2.8. Ύπαρξη της υπακολουθίας	30
Κεφάλαιο 3. Αλγόριθμοι ΑΜΠΣ	33
3.1. Η ενισχυμένη μέθοδος Compass	33
3.2. Η εξελικτική μέθοδος του Box	36
3.3. Η μέθοδος Hooke – Jeeves	39
Κεφάλαιο 4. Αριθμητικά πειράματα	43
4.1. Συναρτήσεις δοκιμής για προβλήματα βελτιστοποίησης	43
4.2. Μεθοδολογία	46
4.3. Βελτιστοποίηση της σφαιρικής συνάρτησης	47
4.4. Βελτιστοποίηση της συνάρτησης Rosenbrock	48
4.5. Βελτιστοποίηση της συνάρτησης Trid	50
4.6. Βελτιστοποίηση της συνάρτησης Ackley	51
4.7. Εξάρτηση των μεθόδων από τις επαναλήψεις	52
4.8. Εξάρτηση των μεθόδων από την ακτίνα	53
Παράρτημα Α'. Κώδικες Matlab	55
Α'.1. Το κύριο script all_methods	55
Α'.2. Το script all_methods_iter	60
Α'.3. Το script all_methods_radius	65
Παράρτημα. Βιβλιογραφία	67



## Πρόλογος

Το παρόν κείμενο αποτελεί τη διπλωματική εργασία της γράφουσας για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης στα Εφαρμοσμένα Μαθηματικά από το Τμήμα Μαθηματικών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών. Πραγματεύεται έναν σημαντικό κλάδο της Αριθμητικής Ανάλυσης που αφορά τη βελτιστοποίηση συναρτήσεων, με το ενδιαφέρον να εντοπίζεται σε μεθόδους που δεν κάνουν χρήση τεχνικών του Διαφορικού Λογισμού. Εξετάζουμε το θέμα τόσο από θεωρητικής, όσο και από υπολογιστικής άποψης, με υλοποίηση των σχετικών αλγορίθμων στο Matlab.

Η εργασία είναι χωρισμένη σε τέσσερα κεφάλαια. Το Κεφάλαιο 1 έχει εισαγωγικό χαρακτήρα και προετοιμάζει τον αναγνώστη για την ανάλυση που ακολουθεί. Δίνεται ο σχετικός συμβολισμός και προτείνεται ο κατάλληλος φορμαλισμός για το πρόβλημα που μας ενδιαφέρει. Το Κεφάλαιο 2 είναι κεντρικής σημασίας, αφού περιέχει το θεωρητικό υπόβαθρο για τις μεθόδους άμεσης αναζήτησης που εξετάζουμε. Ως διαφωτιστικό παράδειγμα, αναλύουμε με λεπτομέρεια τη σχετικά απλή μέθοδο Compass και δίνουμε τον κώδικα Matlab που την υλοποιεί. Στο Κεφάλαιο 3 εξετάζουμε τις άλλες τρεις μεθόδους (ενισχυμένη Compass, Box και Hooke – Jeeves) που θα χρησιμοποιήσουμε, αντιμετωπίζοντάς τις ως ειδικές περιπτώσεις της γενικής θεωρίας. Για κάθε μέθοδο, δίνουμε και πάλι τον κώδικα Matlab. Τέλος, στο Κεφάλαιο 4 εκτελούμε κάποια αριθμητικά πειράματα πάνω σε τέσσερις συναρτήσεις δοκιμής, προκειμένου να ελέγξουμε την αποτελεσματικότητα των μεθόδων, τόσο μεμονωμένα, όσο και συγκριτικά.

Θα ήθελα να ευχαριστήσω θερμά, και από τη θέση αυτή, τον επιβλέποντα καθηγητή μου κ. Μιχάλη Δρακόπουλο που μου εμπιστεύτηκε αυτό το τόσο ενδιαφέρον, αλλά και απαιτητικό, θέμα για την υλοποίηση της διπλωματικής μου εργασίας. Χωρίς την πολύτιμη καθοδήγησή του θα ήταν αδύνατο να φέρω εις πέρας το έργο αυτό. Επίσης, θα ήθελα να εκφράσω τις ευχαριστίες μου στα άλλα δύο μέλη της τριμελούς μου επιτροπής, κ.κ. Βασίλη Δουγαλή και Αντώνη Οικονόμου, για τις εύστοχες παρατηρήσεις τους.

Αθήνα 2017



## ΚΕΦΑΛΑΙΟ 1

### Εισαγωγή

#### 1.1. Βελτιστοποίηση χωρίς παραγώγους

Στη γενική του διατύπωση ένα πρόβλημα βελτιστοποίησης, είναι ένα πρόβλημα εύρεσης της καλύτερης δυνατής λύσης από ένα δεδομένο σύνολο διαθέσιμων επιλογών. Στην απλούστερη περίπτωση, συνίσταται στην ελαχιστοποίηση (ή μεγιστοποίηση) μιας πραγματικής συνάρτησης. Η θεωρία βελτιστοποίησης και οι σχετικές τεχνικές καταλαμβάνουν μια μεγάλη περιοχή των εφαρμοσμένων μαθηματικών. Πριν παρουσιάσουμε πως παριστάνεται ένα πρόβλημα βελτιστοποίησης, θα ήταν χρήσιμο να υπενθυμίσουμε τον ορισμό του τοπικού ελαχίστου (μεγίστου).

**1.1.1. ΟΡΙΣΜΟΣ.** Δίνεται μια συνάρτηση  $f : A \rightarrow \mathbb{R}$ ,  $A \subset \mathbb{R}^N$ . Λέμε ότι η  $f$  παρουσιάζει *τοπικό ελάχιστο* (αντ. *τοπικό μέγιστο*) στο  $\mathbf{x}_0 \in A$  αν υπάρχει  $\delta > 0$  τέτοιο ώστε για κάθε  $\mathbf{x} \in A$  με  $|\mathbf{x} - \mathbf{x}_0| \leq \delta$  να ισχύει ότι  $f(\mathbf{x}) \geq f(\mathbf{x}_0)$  (αντ.  $f(\mathbf{x}) \leq f(\mathbf{x}_0)$ ). Το  $\mathbf{x}_0$  ονομάζεται *θέση τοπικού ελαχίστου* (αντ. *τοπικού μεγίστου*) και η τιμή  $f(\mathbf{x}_0)$  ονομάζεται *τοπικό ελάχιστο* (αντ. *τοπικό μέγιστο*) της  $f$ .

Στο εξής θα ασχοληθούμε με προβλήματα ελαχιστοποίησης, χωρίς να περιορίζεται η γενικότητα, διότι τα τοπικά μέγιστα της συνάρτησης  $f$  είναι τα τοπικά ελάχιστα της συνάρτησης  $-f$ . Συνήθως, το σύνολο  $A$  καθορίζεται από ένα σύνολο περιορισμών και συνθηκών που πρέπει να ικανοποιούνται για να έχει νόημα η συνάρτηση  $f$ . Όταν  $A = \mathbb{R}^N$  τότε λέμε ότι έχουμε πρόβλημα ελαχιστοποίησης χωρίς περιορισμούς. Η συνάρτηση  $f$  εμφανίζεται με διάφορες ονομασίες στη βιβλιογραφία, με επικρατέστερες αυτές της *αντικειμενικής συνάρτησης* και της *συνάρτησης απώλειας/κόστους*. Στα προβλήματα με τα οποία θα ασχοληθούμε, υποθέτουμε ότι η παράγωγος της  $f$  δεν είναι ούτε συμβολικά ούτε αριθμητικά διαθέσιμη και επομένως δεν είναι διαθέσιμες και οι τεχνικές του διαφορικού λογισμού. Αυτή είναι μια πολύ εύλογη υπόθεση, αφού σε αρκετές περιπτώσεις το υπολογιστικό κόστος υπολογισμού της κλίσης  $\nabla f$  είναι εξαιρετικά μεγάλο, ακόμη και αν είναι γνωστός ο τύπος της.

Παρ' όλα αυτά, για την απόδειξη σύγκλισης των διάφορων μεθόδων, η υπόθεση της παραγωγισιμότητας είναι αναγκαία. Ο λόγος είναι ότι χρειαζόμαστε το διάνυσμα κλίσης που μας «δείχνει προς τα που» μειώνεται κατά βέλτιστο τρόπο η αντικειμενική συνάρτηση. Θα πρέπει όμως να τονιστεί ότι σε αυτές τις αποδείξεις, η κλίση χρησιμοποιείται *θεωρητικά* και σε καμία από τις μεθόδους που θα δούμε δεν απαιτείται ο υπολογισμός κάποιας παραγώγου.

Η βελτιστοποίηση χωρίς τη χρήση παραγώγων είναι μια επιστημονική περιοχή μακράς ιστορίας και τρέχουσας γρήγορης ανάπτυξης, που τροφοδοτείται από έναν αυξανόμενο αριθμό εφαρμογών. Η εμφάνιση των πρώτων αλγορίθμων χρονολογείται πίσω στα 1950. Μια ανασκόπηση της σχετικής έρευνας γίνεται στο [7].

Οι μέθοδοι άμεσης αναζήτησης ήταν και παραμένουν δημοφιλείς στους επιστημονικούς κύκλους της πληροφορικής και των μηχανικών, οι οποίοι ήθελαν, γενικά, να αποφεύγουν τον υπολογισμό παραγώγων. Η κριτική που δέχτηκαν οι πρώιμες προσπάθειες (μέχρι και τη δεκαετία του 1970) ήταν ότι οι μέθοδοι αυτές στερούνταν ισχυρού μαθηματικού υποβάθρου. Πράγματι, οι επαναληπτικές μέθοδοι που είχαν προταθεί βασιζόταν μόνο σε εμπειρικά επιχειρήματα και δεν υπήρχε κάποια απόδειξη για τη σύγκλισή τους. Επίσης δεν είχε δοθεί σαφής ορισμός του τι είναι μια «μέθοδος άμεσης αναζήτησης». Ο σχετικός όρος φαίνεται να εμφανίζεται πρώτη φορά στο άρθρο [6] και έκτοτε χρησιμοποιήθηκε αδιακρίτως, φτάνοντας στο σημείο να περιγράφει ουσιαστικά οποιαδήποτε μέθοδο δε χρησιμοποιεί παραγώγους. Η επόμενη παράγραφος είναι μια προσπάθεια να τεθεί το ζήτημα σε πιο αυστηρή βάση.

## 1.2. Φορμαλισμός του προβλήματος και συμβολισμός

Η πρώτη απόπειρα για τον τυπικό – αξιωματικό ορισμό του τι είναι μια μέθοδος άμεσης αναζήτησης γίνεται και πάλι στο άρθρο [6]. Εμείς θα χρησιμοποιήσουμε μια πιο εύχρηστη μορφή που δίνεται στο [9]. Έστω, λοιπόν,  $f : A \rightarrow \mathbb{R}$ ,  $A \subseteq \mathbb{R}^N$ ,  $n$  αντικειμενική συνάρτηση. Ορίζουμε μια σχέση  $\succeq$  στα σημεία του χώρου  $A$  ως εξής:

$$\mathbf{x} \succeq \mathbf{y} \Leftrightarrow f(\mathbf{x}) \leq f(\mathbf{y}).$$

Η σχέση  $\succeq$  είναι μια προτίμηση στα σημεία του  $A$  και έχει τις ιδιότητες μιας σχέσης διάταξης εκτός από αυτή της αντισυμμετρικότητας. Η ερμηνεία είναι η εξής:  $\mathbf{x} \succeq \mathbf{y}$ , δηλαδή το  $\mathbf{x} \in A$  είναι εξίσου ή περισσότερο προτιμώτερο από το  $\mathbf{y} \in A$ , εφόσον το πρώτο δε δίνει μεγαλύτερη τιμή στην αντικειμενική συνάρτηση από ότι το δεύτερο. Αν  $\mathbf{x} \succeq \mathbf{y}$  και  $\mathbf{x} \preceq \mathbf{y}$ , τότε  $f(\mathbf{x}) = f(\mathbf{y})$  και είμαστε αδιάφοροι στην επιλογή μας ανάμεσα στα  $\mathbf{x}$ ,  $\mathbf{y}$ . Στην περίπτωση αυτή γράφουμε  $\mathbf{x} \sim \mathbf{y}$  και  $\sim$  είναι μια σχέση ισοδυναμίας στο  $A$ . Γράφουμε επίσης  $\mathbf{x} \succ \mathbf{y}$  και λέμε ότι το  $\mathbf{x}$  είναι προτιμότερο του  $\mathbf{y}$  αν  $\mathbf{x} \succeq \mathbf{y}$  και  $\mathbf{x} \not\sim \mathbf{y}$ . Στην περίπτωση αυτή, ισχύει  $f(\mathbf{x}) < f(\mathbf{y})$  και άρα το  $\mathbf{x}$  δίνει μικρότερη τιμή στην αντικειμενική συνάρτηση απ' ότι το  $\mathbf{y}$ .

Στα πλαίσια αυτής της εργασίας, θα ασχοληθούμε με  $N$ -διάστατα προβλήματα χωρίς περιορισμούς. Συνεπώς, η αντικειμενική μας συνάρτηση είναι της μορφής  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ . Θα συμβολίζουμε με  $\mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(N)})$  ένα τυπικό διάνυσμα (σημείο) του  $\mathbb{R}^N$ . Στόχος μας είναι η κατασκευή μιας ακολουθίας

$$(\mathbf{x}_n) = ((x_n^{(1)}, x_n^{(2)}, \dots, x_n^{(N)})) \quad , \quad n = 0, 1, 2, \dots$$

σημείων του  $\mathbb{R}^N$ , για την οποία ισχύει

$$\mathbf{x}_0 \prec \mathbf{x}_1 \prec \mathbf{x}_2 \prec \dots \prec \mathbf{x}_n \prec \mathbf{x}_{n+1} \prec \dots$$



Το προφανές επιθυμητό αποτέλεσμα είναι η  $(\mathbf{x}_n)$  να συγκλίνει σε κάποιο τοπικό ελάχιστο  $\mathbf{x}_*$  της  $f$ . Το διάνυσμα  $\mathbf{x}_0$  θα ονομάζεται *αρχική εκτίμηση* και το  $\mathbf{x}_n$  *τρέχουσα επανάληψη*. Η ακολουθία  $(\mathbf{x}_n)$  ορίζεται αναδρομικά μέσω ενός αλγορίθμου, δηλαδή η *επόμενη επανάληψη* υπολογίζεται με τη βοήθεια της τρέχουσας:

$$\mathbf{x}_{n+1} = \varphi(\mathbf{x}_n),$$

όπου  $\varphi$  είναι συνάρτηση που υπολογίζεται χρησιμοποιώντας τις τέσσερις πράξεις της αριθμητικής, σε πεπερασμένο αριθμό βημάτων. Αυτό που διαφοροποιεί, λοιπόν, τη μία μέθοδο από την άλλη είναι ο τρόπος με τον οποίο καθορίζεται η  $\varphi$ .

Με  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$  συμβολίζουμε την κανονική βάση του  $\mathbb{R}^N$ , δηλ.  $\mathbf{e}_i$  είναι το διάνυσμα που έχει μονάδα στην  $i$ -οστή συντεταγμένη και μηδέν σε όλες τις υπόλοιπες. Κατά τα γνωστά, κάθε  $\mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(N)}) \in \mathbb{R}^N$  γράφεται

$$\mathbf{x} = x^{(1)}\mathbf{e}_1 + x^{(2)}\mathbf{e}_2 + \dots + x^{(N)}\mathbf{e}_N.$$

Γνωρίζουμε ότι κάθε μοναδιαίο (ήτοι με μοναδιαίο μέτρο) διάνυσμα  $\mathbf{d} \in \mathbb{R}^N$  ονομάζεται *κατεύθυνση* και τα  $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N$  ορίζουν τις θεμελιώδεις κατευθύνσεις στον  $\mathbb{R}^N$ . Θα συμβολίζουμε το σύνολο των κατευθύνσεων του  $\mathbb{R}^N$  με  $\mathbb{S}^N$ .

**1.2.1. ΟΡΙΣΜΟΣ.** Έστω  $\mathbf{x} \in \mathbb{R}^N$  και  $\mathbf{d} \in \mathbb{S}^N$ . Λέμε ότι η  $\mathbf{d}$  είναι *κατεύθυνση καθόδου* για την  $f$  στο  $\mathbf{x}$  αν υπάρχει  $\Delta > 0$  ώστε  $f(\mathbf{x} + \Delta\mathbf{d}) < f(\mathbf{x})$ .

Ο παραπάνω ορισμός λέει απλά ότι αν κινηθούμε με κατάλληλα μικρό βήμα προς την κατεύθυνση του  $\mathbf{d}$ , τότε θα πετύχουμε μείωση της τιμής της  $f$ . Επανερχόμενοι στην κατασκευή της μεθόδου άμεσης αναζήτησης  $(\mathbf{x}_n)$ , δεδομένης της τρέχουσας επανάληψης  $\mathbf{x}_n$ , αρκεί να βρούμε μια κατεύθυνση καθόδου  $\mathbf{d}_n$  για την  $f$  στο  $\mathbf{x}_n$  και να ορίσουμε

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{d}_n, \tag{1.1}$$

όπου  $\Delta_n > 0$  είναι το βήμα. Όλες οι μέθοδοι που θα εξετάσουμε θα είναι της μορφής (1.1) και επομένως αυτό που διαφοροποιεί τις μεθόδους είναι με ποιόν τρόπο λαμβάνεται σε κάθε επανάληψη η απόφαση για τα  $\Delta_n > 0$ ,  $\mathbf{d}_n \in \mathbb{S}^N$ . Η επιλογή της αρχικής εκτίμησης  $\mathbf{x}_0$  και του αρχικού βήματος  $\Delta_0$  είναι επίσης ένα ζήτημα που θα πρέπει να ληφθεί υπόψη.

### 1.3. Συζήτηση για την παραγωγίσιμη αντικειμενική συνάρτηση

Αν μια επαναληπτική αριθμητική μέθοδος για την εύρεση τοπικού ελαχίστου της  $f$  παράγει μία ακολουθία  $(\mathbf{x}_n)$  με την ιδιότητα  $\mathbf{x}_n \prec \mathbf{x}_{n+1}$  συγκλίνει, τότε επιβεβαιώνει τη διαισθητική άποψη ότι για να είναι «στοιχειωδώς καλή» μια μέθοδος ελαχιστοποίησης, θα πρέπει σε κάθε επανάληψη να δίνει αυστηρά μικρότερη τιμή στην αντικειμενική συνάρτηση. Το πόσο «μικρότερη», όμως, είναι προφανώς ένα σημαντικό ζήτημα προς συζήτηση και θα πρέπει να θεσπιστούν κατάλληλα κριτήρια για το τι σημαίνει μια «επαρκής» ή «αποδεκτή» μείωση της τιμής.

Υποθέτουμε ότι η  $f$  είναι παραγωγίσιμη στο  $\mathbf{x} \in \mathbb{R}^N$  με  $\nabla f(\mathbf{x}) \neq \mathbf{0}$ . Τότε η

$$\mathbf{d}_S(\mathbf{x}) := -\frac{\nabla f(\mathbf{x})}{|\nabla f(\mathbf{x})|} \quad (1.2)$$

ονομάζεται *κατεύθυνση μέγιστης καθόδου* της  $f$  στο  $\mathbf{x}$ , διότι, όπως γνωρίζουμε από τον Λογισμό, η  $\mathbf{d}_S$  «δείχνει» προς τη μεγαλύτερη μείωση των τιμών της  $f$ . Έστω τώρα  $\mathbf{d} \in \mathbb{S}^N$  τέτοια ώστε

$$\mathbf{d} \cdot \mathbf{d}_S(\mathbf{x}) > 0, \quad (1.3)$$

δηλαδή η γωνία της  $\mathbf{d}_S(\mathbf{x})$  με τη  $\mathbf{d}$  είναι μικρότερη της ορθής. Από το ανάπτυγμα Taylor έχουμε

$$f(\mathbf{x} + h\mathbf{d}) = f(\mathbf{x}) + h\nabla f(\mathbf{x}) \cdot \mathbf{d} + O(h^2),$$

και άρα με κατάλληλα μικρή επιλογή του  $h > 0$  μπορούμε να πετύχουμε μείωση της τιμής της  $f$ . Άρα η  $\mathbf{d}$  είναι κατεύθυνση καθόδου.

Κατόπιν τούτου, είναι προφανές ότι επιλέγοντας  $\mathbf{d}_n \in \mathbb{S}^N$  η οποία ικανοποιεί την (1.3), η (1.1) δίνει μια μέθοδο άμεσης αναζήτησης. Αυτού του είδους οι μέθοδοι ονομάζονται *μέθοδοι γραμμικής αναζήτησης*. Ειδικότερα, αν θέσουμε  $\mathbf{d}_n := \mathbf{d}_S(\mathbf{x}_n)$ , τότε παίρνουμε τη *μέθοδο μέγιστης καθόδου*. Είναι προφανές ότι η επιλογή αυτή είναι η βέλτιστη δυνατή. Από την άλλη, αν μεν γνωρίζουμε ότι η συνάρτηση είναι παραγωγίσιμη, αλλά ο αναλυτικός τύπος της κλίσης δεν είναι διαθέσιμος ή είναι πολύπλοκος, τότε εμφανίζεται το πρόβλημα της επιλογής μιας «καλής» κατεύθυνσης καθόδου ώστε να πάρουμε την επόμενη προσέγγιση της μεθόδου.

Πότε, όμως, είναι «καλή» μια κατεύθυνση καθόδου, ποιά είναι δηλαδή τα κριτήρια που πρέπει να βάλουμε για την επιλογή της; Μια επιλογή που απλώς μειώνει την τιμή της συνάρτησης, δηλ.  $x_{n+1} < x_n$  δεν εξασφαλίζει τη σύγκλιση σε ελάχιστο, έστω και αν για την ακολουθία των βημάτων  $(\Delta_n)$  ισχύει  $\Delta_n \rightarrow 0$ , και αυτό φαίνεται στα ακόλουθα στοιχειώδη παραδείγματα.

**1.3.1. ΠΑΡΑΔΕΙΓΜΑ.** Θεωρούμε τη συνάρτηση μιας μεταβλητής  $f(x) = x^2$ ,  $x \in \mathbb{R}$ , η οποία έχει ολικό ελάχιστο στο 0. Θεωρούμε την ακολουθία

$$x_n := a + \frac{1}{n},$$

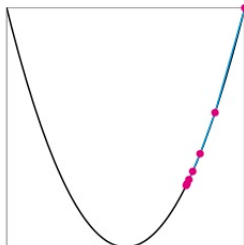
όπου  $a > 0$ . Επειδή η  $(x_n)$  είναι φθίνουσα ακολουθία και η  $f$  είναι αύξουσα στο  $[0, +\infty)$ , έχουμε ότι  $x_n < x_{n+1}$ . Όμως  $x_n \rightarrow a \neq 0$ . Ο λόγος της αποτυχίας αυτής της μεθόδου είναι ο εξής: αφού βρισκόμαστε στη μία διάσταση, υπάρχουν δύο κατευθύνσεις, «αριστερά» ( $d_n = -1$ ) και «δεξιά» ( $d_n = +1$ ). Ξεκινώντας με μια αρχική εκτίμηση  $x_0 = a + 2$ , έχουμε ότι ισχύει

$$x_{n+1} - x_n = \frac{1}{n+1} - \frac{1}{n} = -\left(\frac{1}{n} - \frac{1}{n+1}\right)$$

Άρα έχουμε κίνηση προς αριστερά με βήμα

$$\Delta_n = \frac{1}{n} - \frac{1}{n+1}$$

Έτσι, ενώ κινούμαστε προς τη σωστή κατεύθυνση, το βήμα είναι πολύ μικρό για να «φτάσουμε» στο επιθυμητό σημείο.



1.3.2. ΠΑΡΑΔΕΙΓΜΑ. Παίρνουμε και πάλι τη συνάρτηση  $f(x) = x^2$ ,  $x \in \mathbb{R}$  και τροποποιούμε τη μέθοδο ως ακολούθως:

$$x_n := (-1)^n \left( a + \frac{1}{n} \right).$$

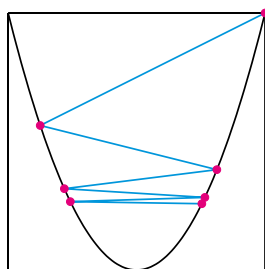
Επειδή  $|x_{n+1}| < |x_n|$  έχουμε και πάλι ότι  $x_n < x_{n+1}$ . Η ακολουθία  $(x_n)$  δε συγκλίνει (ταλαντεύεται μεταξύ των οριακών σημείων  $-a$ ,  $a$ ). Ισχύει τώρα

$$x_{n+1} - x_n = (-1)^{n+1} \left( 2a + \frac{1}{n} + \frac{1}{n+1} \right),$$

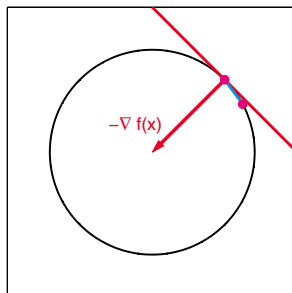
οπότε η μέθοδος κινείται δεξιά όταν  $n$  περιττός (σωστά, διότι τότε  $x_n < 0$ ), αριστερά όταν  $n$  περιττός (επίσης σωστά, διότι τότε  $x_n > 0$ ) και με βήμα

$$\Delta_n = 2a + \frac{1}{n} + \frac{1}{n+1}.$$

Τώρα ο λόγος αποτυχίας της μεθόδου εντοπίζεται στο γεγονός ότι το βήμα είναι πολύ μεγάλο.



Ένα άλλο πρόβλημα που μπορεί να δημιουργηθεί σε μεγαλύτερες διαστάσεις παρουσιάζεται στο ακόλουθο σχήμα (όπου  $N = 2$ ):



Με μαύρο χρώμα έχουμε σημειώσει μία ισοσταθμική καμπύλη της αντικειμενικής συνάρτησης  $f$  και με κόκκινο χρώμα είναι η κατεύθυνση μέγιστης καθόδου (η κλίση, κάθετη στην ισοσταθμική καμπύλη). Η κατεύθυνση που σημειώνεται με μπλε χρώμα είναι μεν κατεύθυνση καθόδου, αλλά προκαλεί πάρα πολύ μικρή κάθοδο στην τιμή της  $f$ , καθώς είναι σχεδόν εφαπτόμενη στην ισοσταθμική καμπύλη. Με άλλα λόγια, κατευθύνσεις καθόδου που είναι σχεδόν κάθετες στην κατεύθυνση μέγιστης καθόδου θα πρέπει να αποφεύγονται.

Προκειμένου να αντιμετωπιστεί το πρόβλημα της επιλογής πολύ μικρού ή πολύ μεγάλου βήματος, μπορεί να χρησιμοποιηθεί το κριτήριο των Armijo – Goldstein – Wolfe [1, 4, 11]. Σύμφωνα με το κριτήριο αυτό, με δεδομένα την τρέχον σημείο  $\mathbf{x}_n$  και την κατεύθυνση καθόδου  $\mathbf{d}_n$ , το βήμα  $\Delta_n$  επιλέγεται κατά τρόπον ώστε

$$f(\mathbf{x}_{n+1}) = f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) \leq f(\mathbf{x}_n) + c_1 \Delta_n \mathbf{d}_S(\mathbf{x}_n) \cdot \mathbf{d}_n \quad (1.4)$$

και

$$\mathbf{d}_S(\mathbf{x}_{n+1}) \cdot \mathbf{d}_n \geq c_2 \mathbf{d}_S(\mathbf{x}_n) \cdot \mathbf{d}_n, \quad (1.5)$$

όπου  $0 < c_1 < c_2 < 1$  είναι κατάλληλες σταθερές. Ακριβέστερα, η (1.4) εισάγει ένα άνω φράγμα για τη μείωση και αποθαρρύνει ένα πολύ μεγάλο βήμα, ενώ η (1.5) είναι μια συνθήκη καμπυλότητας, η οποία αποθαρρύνει ένα πολύ μικρό βήμα.

Στη συνέχεια, θα θέλαμε να εξασφαλίσουμε ότι η κατεύθυνση καθόδου δεν είναι «πολύ κάθετη» στην κατεύθυνση της μέγιστης καθόδου. Αυτό εξασφαλίζεται με σχετικά απλό τρόπο απαιτώντας να ισχύει

$$\mathbf{d}_S(\mathbf{x}_n) \cdot \mathbf{d}_n \geq c \quad (1.6)$$

όπου  $c > 0$  είναι κατάλληλη σταθερά. Υπενθυμίζεται ότι το εσωτερικό γινόμενο αριστερά είναι ακριβώς το συνημίτονο της γωνίας που σχηματίζουν τα διανύσματα  $\mathbf{d}(\mathbf{x}_n)$  και  $\mathbf{d}_n$ . Για τον λόγο αυτό, η συνθήκη (1.6) μπορεί, και μάλλον είναι προτιμότερο, να τεθεί στη μορφή

$$\mathbf{d}_S(\mathbf{x}_n) \cdot \mathbf{d}_n \geq \cos \vartheta, \quad (1.7)$$

όπου  $\vartheta$  είναι κατάλληλη γωνία. Θα πρέπει να τονίσουμε εδώ ότι όσο πιο κοντά στο 1 είναι το  $\cos \vartheta$ , τόσο πιο κοντά στο 0 είναι το  $\vartheta$  και έτσι εξασφαλίζουμε μικρές γωνίες με την κατεύθυνση μέγιστης καθόδου.

Τέλος, παραθέτουμε ένα αποτέλεσμα που αφορά τη σύγκλιση της μεθόδου (1.1). Προφανώς, αυτό που επιθυμούμε είναι η ακολουθία  $(\mathbf{x}_n)$  να συγκλίνει σε θέση ελαχίστου  $\mathbf{x}_*$  της αντικειμενικής συνάρτησης  $f$ .

**1.3.3. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  είναι παραγωγίσιμη και η κλίση  $\nabla f$  είναι Lipschitz συνεχής με σταθερά  $M > 0$ , δηλαδή

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq M \|\mathbf{x} - \mathbf{y}\|,$$

για κάθε  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ . Αν ικανοποιούνται οι συνθήκες βήματος (1.4), (1.5) και η συνθήκη κατεύθυνσης (1.6), τότε για την ακολουθία  $(\mathbf{x}_n)$  που παράγει η μέθοδος (1.1) ισχύει

$$\lim_{n \rightarrow \infty} \nabla f(\mathbf{x}_n) = 0.$$

1.3.4. ΠΑΡΑΤΗΡΗΣΕΙΣ. Κάποια σχόλια είναι απαραίτητα σχετικά με το συμπέρασμα της Πρότασης 1.3.3.

1) Η σύγκλιση της  $(x_n)$  δεν εξασφαλίζεται. Αν όμως η  $(x_n)$  συγκλίνει, τότε το όριο  $x_*$  αυτής είναι κρίσιμο σημείο της  $f$ .

2) Για την ακολουθία  $(x_n)$  μπορεί να εξασφαλιστεί ότι είναι φραγμένη, επομένως, λόγω του Θεωρήματος Bolzano – Weierstrass, θα έχει συγκλίνουσα υπακολουθία, έστω στο  $x_*$ . Έπεται ότι το  $x_*$  είναι κρίσιμο σημείο της  $f$ .

3) Εφόσον εξασφαλιστεί με κάποιο τρόπο η ύπαρξη του κρίσιμου σημείου  $x_*$ , δεν είναι σίγουρο ότι αυτό αποτελεί θέση τοπικού ή ολικού ακρότατου, πόσο μάλλον ελαχίστου. Γι' αυτό χρειάζεται μία συνθήκη δεύτερης τάξης (που εμπλέκει, δηλαδή, μερικές παραγώγους δεύτερης τάξης). Μια τέτοια συνθήκη είναι ο Hessian πίνακας της  $f$  στο  $x_*$  να είναι θετικά ορισμένος. Στην περίπτωση αυτή, το γράφημα της  $f$  είναι κυρτό κοντά στο  $x_*$  και η ακολουθία  $(x_n)$  συγκλίνει στο  $x_*$ .



## ΚΕΦΑΛΑΙΟ 2

### Μέθοδοι αναζήτησης με παράγοντα σύνολα

#### 2.1. Η μέθοδος Compass

Προκειμένου να κάνουμε μια ενδελεχή εισαγωγή στις μεθόδους άμεσης αναζήτησης, θα συζητήσουμε μια πολύ απλή μέθοδο, που θα μας δώσει όμως την κεντρική ιδέα για την έννοια των παραγόντων συνόλων και πώς μπορούν αυτά να χρησιμοποιηθούν στη βελτιστοποίηση.

**2.1.1. Περιγραφή.** Η μέθοδος Compass είναι η μέθοδος άμεσης αναζήτησης (1.1), όπου έχοντας ως δεδομένα την τρέχουσα επανάληψη  $\mathbf{x}_n$  και το βήμα  $\Delta_n$ , η κατεύθυνση καθόδου  $\mathbf{d}_n$  επιλέγεται από το σύνολο που περιέχει τις βασικές κατευθύνσεις, μαζί με τις αντίθετες αυτών,

$$\mathcal{D}_\oplus = \mathcal{D}_\oplus := \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N, -\mathbf{e}_1, -\mathbf{e}_2, \dots, -\mathbf{e}_N\}.$$

Η επιλογή της  $\mathbf{d}_n \in \mathcal{D}_\oplus$  μπορεί να γίνει με δύο τρόπους:

- Υπολογίζουμε όλες τις τιμές  $f(\mathbf{x}_n + \Delta_n \mathbf{d})$  όταν η κατεύθυνση  $\mathbf{d}$  διατρέχει το  $\mathcal{D}_\oplus$  και ως  $\mathbf{d}_n$  επιλέγουμε εκείνη που επιφέρει τη μεγαλύτερη μείωση σε σχέση με την  $f(\mathbf{x}_n)$ . Ενδεικνύται όταν το  $\mathcal{D}_\oplus$  είναι μικρό σύνολο.
- Υπολογίζουμε με σειριακό τρόπο τις τιμές  $f(\mathbf{x}_n + \Delta_n \mathbf{d})$  όταν η κατεύθυνση  $\mathbf{d}$  διατρέχει το  $\mathcal{D}_\oplus$  και ως  $\mathbf{d}_n$  επιλέγουμε την πρώτη από αυτές για την οποία ισχύει  $f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) < f(\mathbf{x}_n)$ . Ενδεικνύται όταν το  $\mathcal{D}_\oplus$  είναι μεγάλο σύνολο.

Εφόσον είναι δυνατή η επιλογή του  $\mathbf{d}_n$ , θέτουμε

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \Delta_n \mathbf{d}_n, \quad \Delta_{n+1} := \Delta_n,$$

και λέμε ότι έχουμε μια *επιτυχημένη* επανάληψη. Συμβολίζουμε το σύνολο των δεικτών που δίνουν επιτυχημένη επανάληψη με  $\mathcal{S}$ .

Αν δεν πετυχαίνουμε μείωση της τιμής της  $f$  για κανένα  $\mathbf{d} \in \mathcal{D}_\oplus$ , τότε θέτουμε

$$\mathbf{x}_{n+1} := \mathbf{x}_n, \quad \Delta_{n+1} := \frac{\Delta_n}{2},$$

και λέμε ότι έχουμε μια *αποτυχημένη* επανάληψη. Συμβολίζουμε το σύνολο των δεικτών που δίνουν αποτυχημένη επανάληψη με  $\mathcal{U}$ .

**2.1.2. Σύγκλιση.** Θα εξετάσουμε την περίπτωση  $N = 2$  αλλά τα παρακάτω ισχύουν και για μεγαλύτερη διάσταση. Οποιοδήποτε συμπέρασμα για τη σύγκλιση εμπλέκει συνθήκες ομαλότητας και έτσι στη συνέχεια θα υποθέσουμε ότι η  $f$  είναι παραγωγίσιμη και μάλιστα ότι η  $\nabla f$  είναι συνάρτηση Lipschitz, υπάρχει δηλαδή  $M > 0$  ώστε

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq M \|\mathbf{x} - \mathbf{y}\|. \quad (2.1)$$

Αν  $n \in U$ , τότε

$$f(\mathbf{x}_n) \leq f(\mathbf{x}_n + \Delta_n \mathbf{d})$$

για όλες τις κατευθύνσεις  $\mathbf{d} \in \mathcal{D}_\oplus$ . Από το Θεώρημα Μέσης Τιμής θα έχουμε ότι υπάρχει  $a_n \in [0, 1]$ , ώστε

$$0 \leq f(\mathbf{x}_n + \Delta_n \mathbf{d}) - f(\mathbf{x}_n) = \nabla f(\mathbf{x}_n + a_n \Delta_n \mathbf{d}) \cdot \Delta_n \mathbf{d},$$

και, απλοποιώντας το  $\Delta_n$ ,

$$0 \leq \mathbf{d} \cdot \nabla f(\mathbf{x}_n + a_n \Delta_n \mathbf{d}).$$

Άρα

$$-\mathbf{d} \cdot \nabla f(\mathbf{x}_n) \leq \mathbf{d} \cdot (\nabla f(\mathbf{x}_n + a_n \Delta_n \mathbf{d}) - \nabla f(\mathbf{x}_n)),$$

συνεπώς

$$-\mathbf{d} \cdot \nabla f(\mathbf{x}_n) \leq \|\nabla f(\mathbf{x}_n + a_n \Delta_n \mathbf{d}) - \nabla f(\mathbf{x}_n)\|. \quad (2.2)$$

Είναι προφανές ότι κάθε κατεύθυνση του  $\mathbb{R}^2$  σχηματίζει γωνία μικρότερη ή ίση των  $45^\circ$  με κάποια από τις κατευθύνσεις του  $\mathcal{D}_\oplus$ . Υπάρχει, δηλαδή,  $\mathbf{d} \in \mathcal{D}_\oplus$  ώστε να ισχύει

$$\mathbf{d}_S(\mathbf{x}_n) \cdot \mathbf{d} \geq \cos 45^\circ,$$

όπου  $\mathbf{d}_S(\mathbf{x}_n)$  είναι η κατεύθυνση μέγιστης καθόδου, όπως ορίστηκε στην (1.2). Άρα

$$-\frac{\nabla f(\mathbf{x}_n)}{\|\nabla f(\mathbf{x}_n)\|} \cdot \mathbf{d} \geq \frac{1}{\sqrt{2}}.$$

Ισχύει επομένως

$$\frac{1}{\sqrt{2}} \|\nabla f(\mathbf{x}_n)\| \leq -\mathbf{d} \cdot \nabla f(\mathbf{x}_n),$$

και άρα η (2.2) γράφεται

$$\frac{1}{\sqrt{2}} \|\nabla f(\mathbf{x}_n)\| \leq \|\nabla f(\mathbf{x}_n + a_n \Delta_n \mathbf{d}) - \nabla f(\mathbf{x}_n)\|. \quad (2.3)$$

Όστε η (2.3) δίνει, λόγω της (2.1),

$$\frac{1}{\sqrt{2}} \|\nabla f(\mathbf{x}_n)\| \leq M \|a_n \Delta_n \mathbf{d}\| = M a_n \Delta_n \|\mathbf{d}\| \leq M \Delta_n,$$

και άρα

$$\|\nabla f(\mathbf{x}_n)\| \leq M \sqrt{2} \Delta_n. \quad (2.4)$$

Η ανισότητα (2.4) εξασφαλίζει ότι αν ισχύει

$$\lim_{\substack{n \in U \\ n \rightarrow \infty}} \Delta_n = 0, \quad (2.5)$$

τότε ισχύει και

$$\lim_{\substack{n \in U \\ n \rightarrow \infty}} \nabla f(\mathbf{x}_n) = 0, \quad (2.6)$$

και επομένως η μέθοδος Compass συγκλίνει (με την έννοια που συζητήθηκε στην 1.3.4). Παρατηρούμε, λοιπόν, ότι ενώ οι επιτυχημένες επαναλήψεις προκαλούν τη μείωση στις τιμές της  $f$ , είναι οι αποτυχημένες που μας επιτρέπουν να βγάλουμε συμπεράσματα για τη σύγκλιση!



**2.1.3. Κριτήριο τερματισμού.** Πότε είμαστε ικανοποιημένοι με την προσέγγιση που δίνει η μέθοδος Compass, δηλαδή σε ποιά επανάληψη σταματάει ο αλγόριθμος; Το κριτήριο που θέτουμε είναι το εξής: ο αλγόριθμος σταματάει όταν το βήμα  $\Delta_n$  γίνει μικρότερο από κάποιο προκαθορισμένο κατώφλι  $\Delta_{\text{tol}} > 0$ . Θα πρέπει, όμως, να εξασφαλίσουμε ότι αυτό πράγματι συμβαίνει κάποια στιγμή, ότι δηλαδή υπάρχει  $n$  ώστε  $\Delta_n < \Delta_{\text{tol}}$ . Αυτό μπορεί να εξασφαλιστεί με την υπόθεση ότι οι καμπύλες στάθμης της  $f$  είναι φραγμένες [5]. Τότε οι επιμέρους επιτυχημένες επαναλήψεις μπορούν να έχουν μόνο πεπερασμένο πλήθος και άρα, αργά ή γρήγορα, το βήμα υποδιπλασιάζεται. Με άλλα λόγια, η ακολουθία  $(\Delta_n)$  είναι φθίνουσα και υπάρχει υπακολουθία  $(\Delta_{k_n})$  ώστε

$$\Delta_{k_n} = \frac{\Delta_0}{2^n}$$

Αυτό έχει τρεις θετικές συνέπειες:

- ο αλγόριθμος πράγματι τερματίζει σε πεπερασμένο χρόνο,
- ισχύει η (2.5), και
- η μέθοδος Compass συγκλίνει.

**2.1.4. Αλγόριθμος.** Είναι ο ακόλουθος:

Αρχικοποίηση

Δώσε τη συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ .

Δώσε την αρχική εκτίμηση  $\mathbf{x}_0 \in \mathbb{R}^N$ .

Δώσε το βήμα για το κριτήριο τερματισμού  $\Delta_{\text{tol}} > 0$ .

Δώσε το αρχικό βήμα  $\Delta_0 > \Delta_{\text{tol}}$ .

Όρισε το σύνολο κατευθύνσεων  $\mathcal{D}_{\oplus}$ .

Θέσε  $n = 0$ .

Κύριος βρόχος

Όσο ισχύει  $\Delta_n > \Delta_{\text{tol}}$ ,

Δευτερεύων βρόχος

Δοκίμασε στη σειρά  $\mathbf{d}_n \in \mathcal{D}_{\oplus}$ .

Για την πρώτη που ισχύει  $f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) < f(\mathbf{x}_n)$ , θέσε

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{d}_n, \quad \Delta_{n+1} = \Delta_n,$$

πρόσθεσε το  $n$  στο σύνολο  $S$  και βγες.

Αν δεν υπάρχει τέτοια  $\mathbf{d}_n$ , θέσε

$$\mathbf{x}_{n+1} := \mathbf{x}_n, \quad \Delta_{n+1} := \frac{\Delta_n}{2},$$

πρόσθεσε το  $n$  στο σύνολο  $U$  και βγες.

Θέσε  $n = n + 1$ .

Έξοδος

Δώσε το  $\mathbf{x}_n$  ως λύση.

**2.1.5. Κώδικας Matlab.** Η μέθοδος Compass υλοποιείται στο Matlab με τον ακόλουθο κώδικα.

```
f=input('Give objective function: ');
x_0=input('Give initial approximation: ');
Delta_0=input('Give initial step: ');
Delta_tol=input('Give step tolerance: ');
nmax=input('Give maximum number of iterations: ');

N=length(x_0);
I=eye(N);

x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current);
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;

    for i=1:N
        x_trial=x_current+Delta_current*I(i,:);
        fvplus=feval(f,x_trial);
        if fvplus<fmin
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current-Delta_current*I(i,:);
        fvminus=feval(f,x_trial);
        if fvminus<fmin
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end
```

```

display(' ');
display('Current iteration is: ');
fprintf('%10f \n',x_current);
display('Value of objective function is: ');
fprintf('%10f \n',fmin);
display('Current Delta is: ');
fprintf('%10f\n',Delta_current);
display('Number of iterations: ');
fprintf('%d\n',n-1);
display('Unsuccessful iterations: ');
fprintf('%d\n',nu);

```

## 2.2. Βασικές έννοιες και ορισμοί

2.2.1. ΟΡΙΣΜΟΣ. Έστω  $\mathcal{G} := \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k\} \subseteq \mathbb{S}^N$ . Λέμε ότι το  $\mathcal{G}$  είναι παράγον σύνολο του χώρου  $\mathbb{R}^N$ , αν κάθε  $\mathbf{x} \in \mathbb{R}^N$  γράφεται

$$\mathbf{x} = \lambda^{(1)}\mathbf{d}_1 + \lambda^{(2)}\mathbf{d}_2 + \dots + \lambda^{(k)}\mathbf{d}_k,$$

όπου  $\lambda^{(i)} \geq 0, i = 1, 2, \dots, k$ .

Θα μπορούσαμε, λοιπόν, να πούμε ότι ένα παράγον σύνολο  $\mathcal{G}$  είναι μια «βάση με μη αρνητικές συντεταγμένες». Για το πλήθος  $k$  των διανυσμάτων του  $\mathcal{G}$ , ισχύει πάντα ότι  $k \geq N + 1$ , [3]. Επομένως, το ελάχιστο πλήθος διανυσμάτων που μπορεί να περιέχει ένα παράγον σύνολο είναι  $N + 1$ .

2.2.2. ΠΡΟΤΑΣΗ. Αν το  $\mathcal{G} := \{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k\}$  είναι παράγον σύνολο του  $\mathbb{R}^N$ , τότε περιέχει μια βάση ως γνήσιο υποσύνολό του.

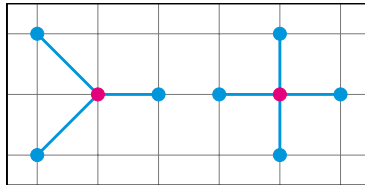
2.2.3. ΠΑΡΑΔΕΙΓΜΑ. Έστω  $N = 2$ . Τότε κάθε παράγον σύνολο του  $\mathbb{R}^2$  θα περιέχει τουλάχιστον 3 διανύσματα. Ένα παράγον σύνολο με ακριβώς  $k = N + 1 = 3$  διανύσματα είναι το

$$\mathcal{G}_1 := \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\},$$

ενώ ένα παράγον σύνολο με  $k = 2N = 4$  διανύσματα είναι το

$$\mathcal{G}_2 := \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\} = \{\mathbf{e}_1, \mathbf{e}_2, -\mathbf{e}_1, -\mathbf{e}_2\}.$$

Τα σύνολα αυτά παριστάνονται στο ακόλουθο σχήμα:



Παρατηρούμε ότι το  $\mathcal{G}_2$  είναι ουσιαστικά το σύνολο της μεθόδου Compass για  $N = 2$ . Γενικότερα, το

$$\mathcal{D}_\oplus := \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N, -\mathbf{e}_1, -\mathbf{e}_2, \dots, -\mathbf{e}_N\}$$

είναι παράγον σύνολο του  $\mathbb{R}^N$ .

Η ακόλουθη πρόταση είναι εξαιρετικά σημαντική, [3].

**2.2.4. ΠΡΟΤΑΣΗ.** Έστω  $\mathcal{G} \subseteq \mathbb{S}^N$  πεπερασμένο. Τα ακόλουθα είναι ισοδύναμα:

α) Το  $\mathcal{G}$  είναι παράγον σύνολο του  $\mathbb{R}^N$ .

β) Για κάθε  $\mathbf{x} \in \mathbb{R}^N$ ,  $\mathbf{x} \neq \mathbf{0}$ , υπάρχει  $\mathbf{d} \in \mathcal{G}$  ώστε  $\mathbf{x} \cdot \mathbf{d} > 0$ .

Αντιστρόφως, ξεκινώντας από μια βάση του  $\mathbb{R}^N$  μπορούμε να κατασκευάσουμε ένα παράγον σύνολο.

**2.2.5. ΠΡΟΤΑΣΗ.** Έστω  $B := \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$  μια βάση μοναδιαίων διανυσμάτων του  $\mathbb{R}^N$ . Τότε το  $\mathcal{G} := \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N, -\mathbf{y}_1, -\mathbf{y}_2, \dots, -\mathbf{y}_N\}$  είναι παράγον σύνολο του  $\mathbb{R}^N$ .

**ΑΠΟΔΕΙΞΗ.** Έστω  $\mathbf{x} \in \mathbb{R}^N$ . Επειδή  $\mathbf{x} \neq \mathbf{0}$  και το  $B$  είναι βάση, θα έχουμε ότι  $\mathbf{x} \cdot \mathbf{y}_i \neq 0$  για κάποιο  $i = 1, 2, \dots, N$ . Άρα θα ισχύει είτε  $\mathbf{x} \cdot \mathbf{y}_i > 0$ , είτε  $-\mathbf{x} \cdot \mathbf{y}_i > 0$ .  $\square$

**2.2.6. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f$  είναι παραγωγίσιμη στο  $\mathbf{x} \in \mathbb{R}^N$  με  $\nabla f(\mathbf{x}) \neq \mathbf{0}$ . Έστω ακόμη  $\mathcal{G}$  παράγον σύνολο του  $\mathbb{R}^N$ . Υπάρχει τότε κατεύθυνση καθόδου  $\mathbf{d} \in \mathcal{G}$  της  $f$  στο  $\mathbf{x}$ .

**ΑΠΟΔΕΙΞΗ.** Λόγω της Πρότασης 2.2.4, υπάρχει  $\mathbf{d} \in \mathcal{G}$  ώστε να ισχύει η ανισότητα (1.3) και άρα, από τη συζήτηση που έπεται αυτής, έχουμε ότι η  $\mathbf{d}$  είναι πράγματι κατεύθυνση καθόδου.  $\square$

**2.2.7. ΟΡΙΣΜΟΣ.** Έστω  $\mathcal{G}$  παράγον σύνολο του  $\mathbb{R}^N$ . Ονομάζουμε μέτρο συνημιτόνου του  $\mathcal{G}$  αριθμό

$$\kappa(\mathcal{G}) := \min_{\mathbf{x} \in \mathbb{R}^N / \{\mathbf{0}\}} \max_{\mathbf{d} \in \mathcal{G}} \frac{\mathbf{x} \cdot \mathbf{d}}{\|\mathbf{x}\|}$$

Ισχύει, μάλιστα, ότι

$$\kappa(\mathcal{G}) = \min_{\mathbf{x} \in \mathbb{S}^N} \max_{\mathbf{d} \in \mathcal{G}} \mathbf{x} \cdot \mathbf{d}.$$

Το  $\kappa(\mathcal{G})$  είναι ένα κάτω φράγμα για το πόσο μεγάλο μπορεί να γίνει το συνημίτονο της γωνίας που σχηματίζουν οι κατευθύνσεις του  $\mathcal{G}$  με οποιαδήποτε άλλη κατεύθυνση του  $\mathbb{R}^N$ . Υπό το πρίσμα της ανισότητας (1.7), λοιπόν, παρέχει το «χειρότερο δυνατό σενάριο» κατά την επιλογή της κατεύθυνσης καθόδου. Κατόπιν τούτου, είμαστε τώρα σε θέση να εκλεπτύνουμε το συμπέρασμα της Πρότασης 2.2.6.

**2.2.8. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f$  είναι παραγωγίσιμη στο  $\mathbf{x} \in \mathbb{R}^N$  με  $\nabla f(\mathbf{x}) \neq \mathbf{0}$  και  $\mathcal{G}$  παράγον σύνολο του  $\mathbb{R}^N$ . Υπάρχει τότε κατεύθυνση καθόδου  $\mathbf{d} \in \mathcal{G}$  της  $f$  στο  $\mathbf{x}$  ώστε να ισχύει

$$\|\nabla f(\mathbf{x})\| \leq -\frac{1}{\kappa(\mathcal{G})} \nabla f(\mathbf{x}) \cdot \mathbf{d}. \quad (2.7)$$

**ΑΠΟΔΕΙΞΗ.** Έστω  $\mathbf{d} \in \mathcal{G}$  η κατεύθυνση που δίνει μέγιστη τιμή στο εσωτερικό γινόμενο  $\mathbf{d}_S(\mathbf{x}) \cdot \mathbf{d}$ . Προφανώς τότε  $\mathbf{d}_S(\mathbf{x}) \cdot \mathbf{d} \geq \kappa(\mathcal{G}) > 0$  και

άρα η  $\mathbf{d}$  είναι κατεύθυνση καθόδου της  $f$  στο  $\mathbf{x}$ . Όστε είναι

$$-\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} \cdot \mathbf{d} \geq \kappa(\mathcal{G}),$$

απ' όπου προκύπτει η ζητούμενη.  $\square$

**2.2.9. ΠΑΡΑΔΕΙΓΜΑ.** Θεωρούμε και πάλι το

$$\mathcal{D}_{\oplus} := \{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_N, -\mathbf{e}_1, -\mathbf{e}_2, \dots, -\mathbf{e}_N\}.$$

Για οποιοδήποτε  $\mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(N)}) \in \mathbb{R}^N$  έχουμε ότι  $\mathbf{x} \cdot \mathbf{e}_i = x^{(i)}$  για κάθε  $i = 1, 2, \dots, N$ , και συνεπώς

$$\begin{aligned} m(\mathbf{x}) &:= \max_{\mathbf{d} \in \mathcal{D}_{\oplus}} \mathbf{x} \cdot \mathbf{d} = \max \{x^{(1)}, x^{(2)}, \dots, x^{(N)}, -x^{(1)}, -x^{(2)}, \dots, -x^{(N)}\} = \\ &= \max \{|x^{(1)}|, |x^{(2)}|, \dots, |x^{(N)}|\} = \|\mathbf{x}\|_{\infty}. \end{aligned}$$

Ισχύει, τώρα, ότι

$$\|\mathbf{x}\|^2 = (x^{(1)})^2 + (x^{(2)})^2 + \dots + (x^{(N)})^2 \leq N \|\mathbf{x}\|_{\infty}^2,$$

και άρα

$$\|\mathbf{x}\|_{\infty} \geq \frac{1}{\sqrt{N}} \|\mathbf{x}\|.$$

Ειδικότερα, για  $\mathbf{x} \in \mathbb{S}^N$ , θα ισχύει  $\|\mathbf{x}\| = 1$  και συνεπώς

$$\|\mathbf{x}\|_{\infty} \geq \frac{1}{\sqrt{N}}.$$

Θεωρούμε, επίσης, το διάνυσμα

$$\mathbf{x}_0 = \left( \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \dots, \frac{1}{\sqrt{N}} \right).$$

Τότε  $\mathbf{x}_0 \in \mathbb{S}^N$  και

$$\|\mathbf{x}_0\|_{\infty} = \frac{1}{\sqrt{N}}.$$

Θα είναι, λοιπόν,

$$\kappa(\mathcal{D}_{\oplus}) = \min_{\mathbf{x} \in \mathbb{S}^N} m(\mathbf{x}) = \frac{1}{\sqrt{N}}.$$

### 2.3. Αναζήτηση με παράγοντα σύνολα

Θα περιγράψουμε τώρα μια γενική μέθοδο άμεσης αναζήτησης της μορφής (1.1), η οποία μπορεί να συμπεριλάβει πολλές γνωστές μεθόδους ως ειδικές περιπτώσεις της. Ονομάζεται μέθοδος Αναζήτησης με Παράγοντα Σύνολα ή ΑΜΠΣ για συντομία. Η γενικότητα της μεθόδου έγκειται στο γεγονός ότι δοθείσης της τρέχουσας επανάληψης  $\mathbf{x}_n$ , η κατεύθυνση καθόδου επιλέγεται από ένα πεπερασμένο σύνολο κατευθύνσεων  $\mathcal{D}_n$ , το οποίο, γενικά, εξαρτάται από το βήμα  $n$ . Κάθε  $\mathcal{D}_n$  γράφεται με τη μορφή

$$\mathcal{D}_n = \mathcal{G}_n \cup \mathcal{H}_n,$$

όπου  $\mathcal{G}_n$  είναι παράγον σύνολο του  $\mathbb{R}^N$  και το  $\mathcal{H}_n$  είναι πιθανώς κενό.

Για την επιλογή της κατεύθυνσης καθόδου  $\mathbf{d}_n \in \mathcal{D}_n$ , χρησιμοποιούμε μια συνάρτηση  $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ , η οποία ονομάζεται *συνάρτηση εξαναγκασμού*, και έχει τις ακόλουθες ιδιότητες:

- (1) είναι συνεχής,
- (2) είναι φθίνουσα, και
- (3) ισχύει

$$\lim_{t \rightarrow 0^+} \frac{\rho(t)}{t} = 0.$$

Προφανώς και η  $\rho(t) = 0$  είναι συνάρτηση εξαναγκασμού. Η επιλογή της  $\mathbf{d}_n$ , αυτή μπορεί να γίνει με δύο τρόπους:

- Υπολογίζουμε όλες τις τιμές  $f(\mathbf{x}_n + \Delta_n \mathbf{d})$  όταν η κατεύθυνση  $\mathbf{d}$  διατρέχει το  $\mathcal{D}_n$  και ως  $\mathbf{d}_n$  επιλέγουμε εκείνη που επιφέρει τη μεγαλύτερη μείωση σε σχέση με την  $f(\mathbf{x}_n) - \rho(\Delta_n)$ . Ενδείκνυται όταν το  $\mathcal{D}_n$  είναι μικρό σύνολο.
- Υπολογίζουμε με σειριακό τρόπο τις τιμές  $f(\mathbf{x}_n + \Delta_n \mathbf{d})$  όταν η κατεύθυνση  $\mathbf{d}$  διατρέχει το  $\mathcal{D}_n$  και ως  $\mathbf{d}_n$  επιλέγουμε την πρώτη για την οποία ισχύει  $f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) < f(\mathbf{x}_n) - \rho(\Delta_n)$ . Ενδείκνυται όταν το  $\mathcal{D}_n$  είναι μεγάλο σύνολο.

Βλέπουμε, δηλαδή, ότι η συνάρτηση εξαναγκασμού  $\rho$  καθορίζει πόσο μεγάλο πρέπει να είναι το βήμα  $\Delta_n$  ώστε να επιτύχουμε μια αποδεκτή μείωση της τιμής της  $f$ . Αυτό μπορεί να συγκριθεί με το κριτήριο Armijo – Goldstein – Wolfe για την περίπτωση της παραγωγίσιμης συνάρτησης. Μάλιστα, αποδεικνύεται η ακόλουθη πρόταση [9].

**2.3.1. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f$  είναι κάτω φραγμένη και η συνάρτηση εξαναγκασμού  $\rho$  ικανοποιεί τις τρεις συνθήκες που δόθηκαν παραπάνω. Τότε

$$\liminf \Delta_n = 0.$$

Εφόσον είναι δυνατή η επιλογή του  $\mathbf{d}_n$ , θέτουμε

$$\mathbf{x}_{n+1} := \mathbf{x}_n + \Delta_n \mathbf{d}_n, \quad \Delta_{n+1} := \sigma_n \Delta_n,$$

όπου  $\sigma_n \geq 1$ . Στην περίπτωση αυτή λέμε ότι έχουμε μια *επιτυχημένη επανάληψη*. Συμβολίζουμε με  $S$  το σύνολο των δεικτών  $n$  που δίνουν επιτυχημένη επανάληψη. Βλέπουμε τώρα ότι δίνεται η δυνατότητα, επιλέγοντας  $\sigma_n > 1$  κατόπιν μιας επιτυχημένης επανάληψης, το βήμα να αυξηθεί.

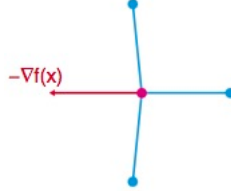
Αν δεν πετυχαίνουμε μείωση της τιμής της  $f$  για κανένα  $\mathbf{d} \in \mathcal{D}_n$ , τότε θέτουμε

$$\mathbf{x}_{n+1} := \mathbf{x}_n, \quad \Delta_{n+1} := \nu_n \Delta_n,$$

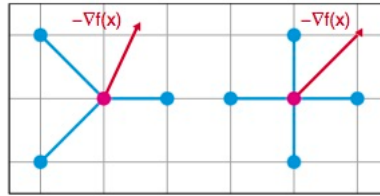
όπου  $0 < \nu_n \leq \nu_{\max} < 1$ . Στην περίπτωση αυτή λέμε ότι έχουμε μια *αποτυχημένη επανάληψη*. Συμβολίζουμε με  $U$  το σύνολο των δεικτών  $n$  που δίνουν αποτυχημένη επανάληψη. Για να μπορεί η μέθοδος να συνεχίσει, θα πρέπει τώρα το βήμα να μειωθεί.

Το παράγον σύνολο  $\mathcal{G}_n$  αποτελεί την κρίσιμη συνιστώσα του συνόλου κατευθύνσεων  $\mathcal{D}_n$  και θα πρέπει να εξασφαλίζει μια αποδεκτή επιλογή

της κατεύθυνσης καθόδου  $d_n$ . Όπως συζητήσαμε στο προηγούμενο κεφάλαιο, θα πρέπει, στην περίπτωση της παραγωγίσιμης συνάρτησης, να αποφεύγονται κατευθύνσεις σχεδόν κάθετες στην κλίση. Με άλλα λόγια, θα πρέπει να αποφεύγονται παράγοντα σύνολα όπως το ακόλουθο:



Η παθογένεια του παραπάνω παράγοντος συνόλου είναι ότι έχει πολύ μικρό μέτρο συνημιτόνου. Αντίθετα, θα θέλαμε καταστάσεις σαν τις ακόλουθες:



Συμπεραίνουμε ότι θα πρέπει να επιβάλλουμε ένα κάτω φράγμα για το μέτρο συνημιτόνου των  $\mathcal{G}_n$ ,

$$\kappa(\mathcal{G}_n) \geq \kappa_{\min} > 0.$$

2.3.2. ΠΑΡΑΤΗΡΗΣΗ. Είναι προφανές ότι η μέθοδος Compass είναι ΑΜΠΣ με  $\mathcal{G}_n := \mathcal{D}_{\oplus}$ ,  $\mathcal{H}_n := \emptyset$ ,  $\sigma_n := 1$ ,  $\nu_n := \frac{1}{2}$ .

2.3.3. ΠΑΡΑΤΗΡΗΣΕΙΣ. α) Ας υποθέσουμε ότι το σύνολο δεικτών  $U$  των αποτυχημένων επαναλήψεων είναι άπειρο. Τότε η ακολουθία των βημάτων  $(\Delta_n)_{n \in U}$  είναι γνησίως φθίνουσα, δεδομένου ότι οι συντελεστές  $\nu_n$  φράσσονται από αριθμό μικρότερο της μονάδας. Αυτή είναι προφανώς και κάτω φραγμένη (από το 0), και επομένως συγκλίνουσα. Μάλιστα, το όριο αυτής της ακολουθίας είναι το 0 επειδή

$$0 < \Delta_n \leq \nu_{\max}^{k(n)} \Delta_0,$$

όπου  $k(n)$  είναι ο αριθμός των αποτυχημένων επαναλήψεων μέχρι και τη  $n$ -οστή. Από υπόθεση,  $0 < \nu_{\max} < 1$  και  $\lim k(n) = \infty$ .

β) Αντίστοιχα, ας υποθέσουμε ότι το σύνολο δεικτών  $S$  των επιτυχημένων επαναλήψεων είναι άπειρο. Τότε η ακολουθία  $(f(\mathbf{x}_n))_{n \in S}$  είναι γνησίως φθίνουσα, επειδή σε κάθε επιτυχημένη επανάληψη παίρνουμε γνήσια μικρότερη τιμή της αντικειμενικής συνάρτησης.

## 2.4. Αλγόριθμος της ΑΜΠΣ

Ο αλγόριθμος είναι ο ακόλουθος:

### Αρχικοποίηση

Δώσε τη συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ .  
 Δώσε την αρχική εκτίμηση  $\mathbf{x}_0 \in \mathbb{R}^N$ .  
 Δώσε το βήμα για το κριτήριο τερματισμού  $\Delta_{\text{tol}} > 0$ .  
 Δώσε το αρχικό βήμα  $\Delta_0 > \Delta_{\text{tol}}$ .  
 Δώσε την συνάρτηση εξαναγκασμού  $\varrho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ .  
 Θέσε  $n = 0$ .

Κύριος βρόχος

Όσο ισχύει  $\Delta_n > \Delta_{\text{tol}}$ ,

Δευτερεύων βρόχος

Όρισε το σύνολο κατευθύνσεων  $\mathcal{D}_n = \mathcal{G}_n \cup \mathcal{H}_n$ .

Δοκίμασε στη σειρά  $\mathbf{d}_n \in \mathcal{D}_n$ .

Για την πρώτη που ισχύει  $f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) < f(\mathbf{x}_n) - \varrho(\Delta_n)$

- όρισε τον συντελεστή αύξησης  $\sigma_n$ , και

- θέσε

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{d}_n, \quad \Delta_{n+1} = \sigma_n \Delta_n,$$

και βγες από τον βρόχο.

Αν δεν υπάρχει τέτοια  $\mathbf{d}_n$ ,

- όρισε τον συντελεστή απομείωσης  $\nu_n$ , και

- θέσε

$$\mathbf{x}_{n+1} := \mathbf{x}_n, \quad \Delta_{n+1} := \nu_n \Delta_n,$$

και βγες από τον βρόχο.

Θέσε  $n = n + 1$ .

Έξοδος

Δώσε το  $\mathbf{x}_n$  ως λύση.

## 2.5. Αναζήτηση με μοτίβα

Θα εξετάσουμε τώρα μια ευρεία κατηγορία άμεσων μεθόδων, όπου τα παράγωγα (όπως και τα βοηθητικά) σύνολα καθορίζονται μέσω μοτίβων (patterns) ή/και πλεγμάτων (lattices). Η γενική θεωρία για τα μοτίβα αναπτύσσεται με λεπτομέρεια στο άρθρο [10].

2.5.1. ΟΡΙΣΜΟΣ. Ένας πίνακας  $P \in \mathbb{R}^{N \times p}$ , με  $p > 2N$ , ονομάζεται *μοτίβο* αν γράφεται ως γινόμενο  $P = BC$ , όπου

- Ο  $B \in \mathbb{R}^{N \times N}$  είναι μη ιδιάζων.
- Ο  $C \in \mathbb{Z}^{N \times p}$  έχει τη μορφή  $C = [M \quad -M \quad L]$ . Ο πίνακας  $M \in \mathbb{Z}^{N \times N}$  είναι μη ιδιάζων και ο πίνακας  $L \in \mathbb{Z}^{N \times (p-2N)}$  έχει τουλάχιστον μία μηδενική στήλη.

Ο  $B$  ονομάζεται *πίνακας βάσης* και ο  $C$  γεννήτορας πίνακας του μοτίβου  $P$ .

Επειδή  $P = [BM \quad -BM \quad BL]$  και οι πίνακες  $B, M$  είναι μη ιδιάζοντες, λόγω της πρότασης 2.2, οι πρώτες  $2N$  στήλες του  $P$ , αφού κανονικοποιηθούν, αποτελούν ένα παράγον σύνολο του  $\mathbb{R}^N$ . Ακόμη, οι υπόλοιπες



$2N-p$  στήλες του  $P$  γράφονται ως μη αρνητικός γραμμικός συνδυασμός των  $2N$  πρώτων στηλών, και μάλιστα οι «συντεταγμένες» είναι φυσικοί αριθμοί.

Η αναζήτηση με μοτίβα είναι μια ΑΜΠΣ, όπου η κατεύθυνση  $\mathbf{d}_n$  επιλέγεται κάθε φορά από τις κανονικοποιημένες στήλες ενός μοτίβου  $P_n = BC_n$ . Πιο συγκεκριμένα, για τη δημιουργία του γεννήτορα  $C_n = [M_n \ -M_n \ L_n]$ , ο πίνακας  $M_n$  επιλέγεται μέσα από ένα πεπερασμένο σύνολο μη ιδιζόντων πινάκων του  $\mathbb{Z}^{N \times N}$ . Οι πρώτες  $2N$  κανονικοποιημένες στήλες αποτελούν, λοιπόν, ένα παράγον σύνολο της μορφής

$$\mathcal{G}_n := \left\{ \mathbf{d}_1^{i(n)}, \mathbf{d}_2^{i(n)}, \dots, \mathbf{d}_N^{i(n)}, -\mathbf{d}_1^{i(n)}, -\mathbf{d}_2^{i(n)}, \dots, -\mathbf{d}_N^{i(n)} \right\},$$

όπου το  $B_{i(n)} = \left\{ \mathbf{d}_1^{i(n)}, \mathbf{d}_2^{i(n)}, \dots, \mathbf{d}_N^{i(n)} \right\}$  είναι βάση του  $\mathbb{R}^N$  και  $i(n) = 1, 2, \dots, \beta$ . Επειδή οι επιλογές των  $\mathcal{G}_n$  είναι πεπερασμένες το πλήθος, ο αριθμός  $\kappa_{\min}$  είναι καλά ορισμένος και ίσος με το μικρότερο μέτρο συννημιτόνου εξ' αυτών.

Το βοηθητικό σύνολο κατευθύνσεων  $\mathcal{H}_n$  ορίζεται από τις κανονικοποιημένες στήλες του πίνακα  $L_n$ , εκτός της μηδενικής και μια κατεύθυνση  $\mathbf{h} \in \mathcal{H}_n$  γράφεται με τη μορφή

$$\mathbf{h} = \frac{1}{c} \left( a^{(1)} \mathbf{d}_1^{i(n)} + \dots + a^{(N)} \mathbf{d}_N^{i(n)} - b^{(1)} \mathbf{d}_1^{i(n)} + \dots - b^{(N)} \mathbf{d}_N^{i(n)} \right),$$

όπου  $a^{(i)}, b^{(i)} \in \mathbb{N}$  και  $c > 0$  είναι το μέτρο του γραμμικού συνδυασμού στο δεξί μέλος. Μπορούμε να απαλλαγούμε από αυτήν τη διαίρεση αν χαλαρώσουμε την υπόθεση ότι τα διανύσματα του  $\mathcal{H}_n$  έχουν μοναδιαίο μέτρο. Αυτό θα πράξουμε στη συνέχεια, χωρίς ιδιαίτερη αναφορά.

**2.5.2. ΠΑΡΑΔΕΙΓΜΑ.** Η μέθοδος Compass είναι τετρωμένα μέθοδος αναζήτησης με μοτίβα, για  $B = I_N$  και σταθερό γεννήτορα  $C = [I_N \ -I_N \ 0]$ . Για  $N = 2$ , συγκεκριμένα, είναι

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \end{bmatrix}.$$

Θα δούμε κι άλλα, περισσότερο ενδιαφέροντα, παραδείγματα στο επόμενο κεφάλαιο.

## 2.6. Αναζήτηση με πλέγματα

Η έννοια του πλέγματος γενικεύει αυτή του μοτίβου κατά έναν αφηρημένο τρόπο, που όμως είναι περισσότερο βολικός για τη μεθοδολογία των ΑΜΠΣ και θα χρησιμοποιηθεί στη συνέχεια. Πιο συγκεκριμένα, θα θεωρήσουμε ότι έχουμε ένα «μεγάλο», αλλά πεπερασμένο, σύνολο κατευθύνσεων

$$\mathbf{G} := \left\{ \mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \dots, \mathbf{d}^{(p)} \right\} \subseteq \mathbb{R}^N.$$

Θα επιτρέψουμε, λοιπόν, και στα διανύσματα του  $\mathbf{G}$  να μην έχουν κατ' ανάγκη μοναδιαίο μέτρο, καθώς αυτό διευκολύνει κατά πολύ την ανάλυσή μας και αποσυμφορεί τον συμβολισμό.

Σε κάθε επανάληψη της ΑΜΠΣ, τώρα, επιλέγουμε ως παράγον σύνολο ένα υποσύνολο του  $\mathbf{G}$ , δηλαδή

$$\mathcal{G}_n \subseteq \mathbf{G}. \quad (2.8)$$

Το βοηθητικό σύνολο κατευθύνσεων  $\mathcal{H}_n$  περιέχει πεπερασμένο πλήθος διανυσμάτων της μορφής

$$\sum_{i=1}^p a^{(i)} \mathbf{d}^{(i)}, \quad (2.9)$$

όπου  $a^{(i)} \in \mathbb{N}$ . Το βήμα  $\Delta_n$  ρυθμίζεται από τις σχετικές παραμέτρους ως εξής: θεωρούμε έναν ακέραιο αριθμό  $\Lambda > 1$  και θέτουμε

$$v_n := \Lambda^{\mu_n}, \text{ όταν } n \in U, \text{ με } \mu_n \in \mathbb{Z}_-^*,$$

και

$$\sigma_n := \Lambda^{v_n}, \text{ όταν } n \in S, \text{ με } v_n \in \mathbb{N}.$$

Έπεται ότι για κάθε  $n = 1, 2, \dots$  ισχύει

$$\Delta_n = \Lambda^{\Gamma_n} \Delta_0, \quad (2.10)$$

για κάποιο  $\Gamma_n \in \mathbb{Z}$ .

**2.6.1. ΠΑΡΑΔΕΙΓΜΑ.** Στον κλασικό (και ευρέως χρησιμοποιούμενο στην πράξη) υποδιπλασιασμό του βήματος όταν προκύψει ανεπιτυχής επανάληψη, έχουμε  $\Lambda = 2$  και

$$\Gamma_n = \begin{cases} 0, & \text{αν } n \in S, \\ -k, & \text{αν } n-1 \in U. \end{cases}$$

Το  $k \in \mathbb{N}$  μετράει τον αριθμό των αποτυχημένων επαναλήψεων μέχρι τη  $n$ -οστή επανάληψη (χωρίς να συμπεριλαμβάνεται αυτή).

**2.6.2. ΛΗΜΜΑ.** Αν τα σύνολα κατευθύνσεων  $\mathcal{G}_n, \mathcal{H}_n$  ικανοποιούν τις υποθέσεις (2.8), (2.9), τότε η ΑΜΠΣ παράγει μια ακολουθία  $(\mathbf{x}_n)$  για την οποία ισχύει

$$\mathbf{x}_n = \mathbf{x}_0 + \sum_{i=1}^p \delta_n(i) \mathbf{d}^{(i)}, \quad n = 1, 2, \dots, \quad (2.11)$$

όπου οι σταθερές  $\delta_n(i)$  δίνονται από τη σχέση

$$\delta_n(i) = \sum_{j=0}^{n-1} a_j^{(i)} \Delta_j, \quad i = 1, 2, \dots, p, \quad (2.12)$$

με  $a_j^{(i)} \in \mathbb{N}$ .

**ΑΠΟΔΕΙΞΗ.** Με επαγωγή στο  $n$ . Δείχνουμε ότι ισχύει για  $n = 1$ , διακρίνοντας τρεις περιπτώσεις:

1. Έστω ότι  $0 \in U$ , δηλαδή ότι η «μηδενική» επανάληψη είναι ανεπιτυχής. Τότε  $\mathbf{x}_1 = \mathbf{x}_0$  και άρα αρκεί να πάρουμε  $\delta_1(i) = 0$ , δηλαδή  $a_0^i = 0$  για κάθε  $i = 1, 2, \dots, p$ .

2. Έστω ότι  $0 \in S$  και  $\mathbf{d}_0 \in \mathcal{G}_0$ , δηλαδή ότι η «μηδενική» επανάληψη είναι επιτυχής και η κίνηση γίνεται σε κατεύθυνση του παράγοντος συνόλου. Τότε και  $\mathbf{d}_0 = \mathbf{d}^{(\kappa)}$ , για κάποιο  $\kappa = 1, 2, \dots, p$ , οπότε

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta_0 \mathbf{d}_0 = \mathbf{x}_0 + \Delta_0 \mathbf{d}^{(\kappa)}.$$

Επομένως, αρκεί να πάρουμε  $\delta_1(\kappa) = \Delta_0$ , δηλαδή  $a_0^{(\kappa)} = 1$ , και  $\delta_1(i) = 0$ , δηλαδή  $a_0^{(i)} = 0$ , για  $i \neq \kappa$ .

3. Έστω ότι  $0 \in S$  και  $\mathbf{d}_0 \in \mathcal{H}_0$ , δηλαδή ότι η «μηδενική» επανάληψη είναι επιτυχής και η κίνηση γίνεται σε κατεύθυνση του βοηθητικού συνόλου. Τότε

$$\mathbf{d}_0 = \sum_{i=1}^p a_0^{(i)} \mathbf{d}^{(i)},$$

άρα θα είναι

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta_0 \mathbf{d}_0 = \mathbf{x}_0 + \sum_{i=1}^p a_0^{(i)} \Delta_0 \mathbf{d}^{(i)}.$$

Δηλαδή  $\delta_1(i) = a_0^{(i)} \Delta_0$ ,  $i = 1, 2, \dots, p$ .

Έστω τώρα ότι οι (2.11), (2.12) ισχύουν για κάποιο  $n \in \mathbb{N}^*$ . Θα δείξουμε ότι ισχύουν και για τον φυσικό  $n+1$ . Διακρίνουμε και πάλι τρεις περιπτώσεις:

1. Έστω ότι  $n \in U$ , δηλαδή ότι η  $n$ -οστή επανάληψη είναι ανεπιτυχής. Τότε  $\mathbf{x}_{n+1} = \mathbf{x}_n$  και το συμπέρασμα έπεται αμέσως από την υπόθεση της μαθηματικής επαγωγής.

2. Έστω ότι  $n \in S$  και  $\mathbf{d}_n \in \mathcal{G}_n$ , δηλαδή ότι η  $n$ -οστή επανάληψη είναι επιτυχής και η κίνηση γίνεται σε κατεύθυνση του παράγοντος συνόλου. Τότε  $\mathbf{d}_n = \mathbf{d}^{(\kappa)}$ , για κάποιο  $\kappa = 1, 2, \dots, p$ , οπότε

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{d}_n = \mathbf{x}_0 + \sum_{i=1}^p \delta_n(i) \mathbf{d}^{(i)} + \Delta_n \mathbf{d}^{(\kappa)}.$$

Επομένως, αρκεί να πάρουμε  $\delta_{n+1}(\kappa) = \delta_n(\kappa) + \Delta_n$  και  $\delta_{n+1}(i) = \delta_n(i)$ , για  $i \neq \kappa$ , που είναι της μορφής (2.12).

3. Έστω ότι  $n \in S$  και  $\mathbf{d}_n \in \mathcal{H}_n$ , δηλαδή ότι η  $n$ -οστή επανάληψη είναι επιτυχής και η κίνηση γίνεται σε κατεύθυνση του βοηθητικού συνόλου. Τότε

$$\mathbf{d}_n = \sum_{i=1}^p a_n^{(i)} \mathbf{d}^{(i)},$$

άρα θα είναι

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{d}_n = \mathbf{x}_0 + \sum_{i=1}^p \delta_n(i) \mathbf{d}^{(i)} + \sum_{i=1}^p a_n^{(i)} \Delta_n \mathbf{d}^{(i)}.$$

Δηλαδή  $\delta_{n+1}(i) = \delta_n(i) + a_n^{(i)} \Delta_n$ , που είναι της μορφής (2.12).  $\square$

Το παραπάνω λήμμα μας δίνει τη μορφή του  $\mathbf{x}_n$ , ενώ η (2.10) μας δίνει πληροφορία σχετικά με το  $\Delta_n$ . Αν υποθέσουμε, επιπλέον, ότι  $\mathbf{d}^i \in \mathbb{Z}^N$  για κάθε  $\mathbf{d}^{(i)} \in \mathbf{G}$ , παίρνουμε το ακόλουθο σημαντικό αποτέλεσμα.

2.6.3. ΠΡΟΤΑΣΗ. Έστω τα σύνολα κατευθύνσεων  $\mathcal{G}_n$ ,  $\mathcal{H}_n$  και το βήμα  $\Delta_n$  ικανοποιούν τις υποθέσεις (2.8), (2.9), (2.10). Έστω ακόμη ότι  $\mathbf{G} \subset \mathbb{Z}^N$ . Τότε η ΑΜΠΣ παράγει μια ακολουθία  $(\mathbf{x}_n)$ , με

$$\mathbf{x}_n = \mathbf{x}_0 + \Lambda^{\gamma_n} \Delta_0 \sum_{i=1}^p \zeta_n(i) \mathbf{d}^{(i)}, \quad (2.13)$$

όπου

$$\gamma_n := \min\{\Gamma_1, \Gamma_2, \dots, \Gamma_{n-1}\},$$

και

$$\zeta_n(i) = \sum_{j=1}^p a_j^{(i)} \Lambda^{j-\gamma_n}. \quad (2.14)$$

Τα  $a_j^{(i)} \in \mathbb{N}$  είναι αυτά του Λήμματος 2.6.2.

2.6.4. ΠΑΡΑΤΗΡΗΣΕΙΣ. α) Ο όρος «πλέγμα» προκύπτει από το γεγονός ότι τα διανύσματα  $\mathbf{d}^{(i)}$  είναι ακέραια, έχουν δηλαδή ακέραιες συντεταγμένες. Άρα όλα τα σημεία που έχουν ως διανύσματα θέσης

$$\sum_{i=1}^p a^{(i)} \mathbf{d}^{(i)},$$

$a^{(i)} \in \mathbb{N}$ , έχουν ακέραιες συντεταγμένες και λέμε ότι ανήκουν στο πλέγμα που παράγει το σύνολο  $\mathbf{G}$ . Υπό το πρίσμα της Πρότασης 2.6.3, λοιπόν, και επειδή  $\zeta_n(i) \in \mathbb{N}$ , έχουμε ότι η εκτίμηση  $\mathbf{x}_n$  είναι σημείο του πλέγματος, το οποίο όμως έχει μεταφερθεί παράλληλα κατά το διάνυσμα  $\mathbf{x}_0$  και συρρικνωθεί (ή διασταλεί) κατά τον παράγοντα  $\Lambda^{\gamma_n} \Delta_0$ . Θα συμβολίζουμε το τελευταίο αυτό σύνολο (την παράλληλη μεταφορά και συρρίκνωση του αρχικού πλέγματος) με  $\mathcal{M}_n(\mathbf{x}_0, \Delta_0)$ . Όστε δείξαμε ότι αν η ΑΜΠΣ υλοποιηθεί με τη βοήθεια πλεγμάτων, τότε  $\mathbf{x}_n \in \mathcal{M}_n(\mathbf{x}_0, \Delta_0)$  για κάθε  $n = 1, 2, \dots$

β) Αξίζει να σημειωθεί ότι το  $\mathcal{M}_n(\mathbf{x}_0, \Delta_0)$  εξαρτάται από το  $n$  μόνο μέσω της παραμέτρου  $\gamma_n$ , η οποία, με τη σειρά της, εξαρτάται από τους εκθέτες  $\Gamma_n$ . Μάλιστα, ο συντελεστής συρρίκνωσης του αρχικού πλέγματος εξαρτάται αποκλειστικά από το «πόσο αρνητικά» μπορούν να γίνουν τα  $\Gamma_n$ . Το  $\Gamma_n$  γίνεται «πιο αρνητικό» όταν  $n - 1 \in U$ .

γ) Αν  $n \in \mathcal{S}$ , τότε

$$\mathcal{M}_{n+1}(\mathbf{x}_0, \Delta_0) = \mathcal{M}_n(\mathbf{x}_0, \Delta_0).$$

Αν  $n \in U$ , τότε

$$\mathcal{M}_{n+1}(\mathbf{x}_0, \Delta_0) \subsetneq \mathcal{M}_n(\mathbf{x}_0, \Delta_0).$$

δ) Για οποιοδήποτε  $\mathbf{y} \in \mathbb{R}^N$  συμβολίζουμε με  $\mathcal{L}_f(\mathbf{y})$

$$\mathcal{L}_f(\mathbf{y}) := \left\{ \mathbf{x} \in \mathbb{R}^N : f(\mathbf{x}) = f(\mathbf{y}) \right\}$$

το σύνολο στάθμης της  $f$  που διέρχεται από το  $\mathbf{y}$ . Υποθέτουμε ότι το  $\mathcal{L}_f(\mathbf{x}_0)$  είναι φραγμένο σύνολο. Επειδή η ακολουθία  $(f(\mathbf{x}_n))$  είναι φθίνουσα, έχουμε ότι το  $\mathcal{L}_f(\mathbf{x}_n)$  είναι φραγμένο για κάθε  $n$ . Δεδομένου του  $n$ , τώρα, έχουμε ότι τα σύνολα  $\mathcal{L}_f(\mathbf{x}_n)$ ,  $\mathcal{M}_n(\mathbf{x}_0, \Delta_0)$  έχουν το πολύ πεπερασμένου πλήθους κοινά σημεία. Έπεται ότι οι όροι της ακολουθίας

$(\mathbf{x}_k)_{k \geq n}$  δεν μπορούν να μείνουν επ' άπειρον στο  $\mathcal{M}_n(\mathbf{x}_0, \Delta_0)$  και άρα οι επιτυχημένες επαναλήψεις μετά τη  $n$ -οστή έχουν πεπερασμένο πλήθος.

### 2.7. Διερεύνηση σύγκλισης για την ΑΜΠΣ

Στη συνέχεια θα εξετάσουμε τη σύγκλιση της άμεσης μεθόδου με παράγοντα σύνολα (ΑΜΠΣ), υπό την έννοια που συζητήσαμε στην 1.3.4. Θα ακολουθήσουμε την εξής στρατηγική:

- Θα δείξουμε πρώτα ότι αν  $(\mathbf{x}_{k_n})$  είναι μια υπακολουθία της  $(\mathbf{x}_n)$  τέτοια ώστε

$$\lim \Delta_{k_n} = 0, \quad (2.15)$$

τότε ισχύει και

$$\lim \nabla f(\mathbf{x}_{k_n}) = 0. \quad (2.16)$$

Θυμίζουμε ότι στην περίπτωση της μεθόδου Compass, αυτό εξασφαλίστηκε από το γεγονός ότι  $\|\nabla f(\mathbf{x}_n)\| = O(\Delta_n)$ , λόγω της (2.3). Η συνεπαγωγή (2.15)  $\Rightarrow$  (2.16) είναι το αντικείμενο της παρούσας παραγράφου και θα αποδειχθεί με την Πρόταση 2.7.1 παρακάτω.

- Σε δεύτερη φάση, θα αποδείξουμε ότι υπάρχει πράγματι υπακολουθία της  $(\mathbf{x}_n)$  ώστε να ισχύει η (2.15). Στην περίπτωση της μεθόδου Compass, αυτό προέκυψε από το γεγονός ότι μετά από μια αποτυχημένη επανάληψη ακολουθούν το πολύ πεπερασμένου πλήθους επιτυχημένες επαναλήψεις. Η εξαγωγή μιας τέτοιας υπακολουθίας για την ΑΜΠΣ θα εξεταστεί στην επόμενη παράγραφο.

Η ακόλουθη Πρόταση είναι η αντίστοιχη της 1.3.3.

**2.7.1. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$  είναι παραγωγίσιμη και η κλίση  $\nabla f$  είναι Lipschitz συνεχής με σταθερά  $M > 0$ , δηλαδή

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq M \|\mathbf{x} - \mathbf{y}\|,$$

για κάθε  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ . Τότε η ακολουθία  $(\mathbf{x}_n)$  που παράγεται με την ΑΜΠΣ ικανοποιεί, για κάθε  $n \in U$ , την ανισότητα

$$\|\nabla f(\mathbf{x}_n)\| \leq \frac{1}{\kappa_{\min}} \left( M\Delta_n + \frac{\varrho(\Delta_n)}{\Delta_n} \right). \quad (2.17)$$

**ΑΠΟΔΕΙΞΗ.** Έστω  $n \in U$ . Σύμφωνα με τη Πρόταση 2.2.8, υπάρχει κατεύθυνση καθόδου  $\mathbf{k} \in \mathcal{G}_n \subseteq \mathcal{D}_n$  της  $f$  στο  $\mathbf{x}_n$  ώστε

$$\|\nabla f(\mathbf{x}_n)\| \leq -\frac{1}{\kappa(\mathcal{G}_n)} \nabla f(\mathbf{x}_n) \cdot \mathbf{k}. \quad (2.18)$$

Από το Θεώρημα Μέσης Τιμής, επίσης, υπάρχει  $a \in [0, 1]$  ώστε

$$f(\mathbf{x}_n + \Delta_n \mathbf{k}) - f(\mathbf{x}_n) = \Delta_n \nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) \cdot \mathbf{k}.$$

Επειδή έχουμε αποτυχημένη επανάληψη, θα ισχύει

$$f(\mathbf{x}_n + \Delta_n \mathbf{k}) - f(\mathbf{x}_n) + \varrho(\Delta_n) \geq 0,$$

και άρα

$$\Delta_n \nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) \cdot \mathbf{k} + \varrho(\Delta_n) \geq 0.$$

Διαιρούμε διά  $\Delta_n$  και αφαιρούμε και από τα δύο μέλη το  $\nabla f(\mathbf{x}_n) \cdot \mathbf{k}$ , οπότε παίρνουμε

$$-\nabla f(\mathbf{x}_n) \cdot \mathbf{k} \leq (\nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) - \nabla f(\mathbf{x}_n)) \cdot \mathbf{k} + \frac{\varrho(\Delta_n)}{\Delta_n}. \quad (2.19)$$

Από την ανισότητα Cauchy – Schwarz θα έχουμε ότι

$$(\nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) - \nabla f(\mathbf{x}_n)) \cdot \mathbf{k} \leq \|\nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) - \nabla f(\mathbf{x}_n)\| \|\mathbf{k}\|,$$

και ισχύει  $\|\mathbf{k}\| = 1$ . Επιπλέον, από την υπόθεση της Lipschitz συνέχειας,

$$\|\nabla f(\mathbf{x}_n + a\Delta_n \mathbf{k}) - \nabla f(\mathbf{x}_n)\| \leq aM\Delta_n \leq M\Delta_n.$$

Τέλος, από την ανισότητα (2.18), έχουμε

$$\kappa_{\min} \|\nabla f(\mathbf{x}_n)\| \leq \kappa(\mathcal{G}_n) \|\nabla f(\mathbf{x}_n)\| \leq -\nabla f(\mathbf{x}_n) \cdot \mathbf{k},$$

και, συνδυάζοντας όλα τα παραπάνω, βρίσκουμε

$$\kappa_{\min} \|\nabla f(\mathbf{x}_n)\| \leq M\Delta_n + \frac{\varrho(\Delta_n)}{\Delta_n}.$$

□

**2.7.2. ΠΑΡΑΤΗΡΗΣΗ.** Η υπόθεση της Lipschitz συνέχειας για την κλίση  $\nabla f$  της αντικειμενικής συνάρτησης δεν είναι απαραίτητη για την απόδειξη του παραπάνω θεωρήματος και μάλιστα αρκεί μόνο η συνεχής παραγωγισιμότητα της  $f$ . Αυτό που θα αντικαταστήσει την ύπαρξη της σταθεράς Lipschitz είναι το μέτρο συνέχειας της  $\nabla f$  στο  $\mathbf{x} \in \mathbb{R}^N$ , για  $r > 0$ ,

$$\omega(\mathbf{x}, r) = \sup_{\|\mathbf{x}-\mathbf{y}\| \leq r} \|f(\mathbf{x}) - f(\mathbf{y})\|.$$

Πιο συγκεκριμένα, η ανισότητα (2.17) γράφεται τώρα

$$\|\nabla f(\mathbf{x}_n)\| \leq \frac{1}{\kappa_{\min}} \left( \omega(\mathbf{x}_n, \Delta_n) + \frac{\varrho(\Delta_n)}{\Delta_n} \right). \quad (2.17')$$

Συνοψίζοντας τα παραπάνω, φτάνουμε στο κεντρικό συμπέρασμα της παραγράφου.

**2.7.3. ΠΟΡΙΣΜΑ.** Έστω ότι η αντικειμενική συνάρτηση είναι συνεχώς παραγωγίσιμη. Αν  $(\mathbf{x}_{k_n})$  είναι μια υπακολουθία της ακολουθίας  $(\mathbf{x}_n)$  που παράγει η ΑΜΠΣ με  $k_n \in U$  τέτοια ώστε  $\lim \Delta_{k_n} = 0$ , τότε ισχύει και  $\lim \nabla f(\mathbf{x}_{k_n}) = 0$ .

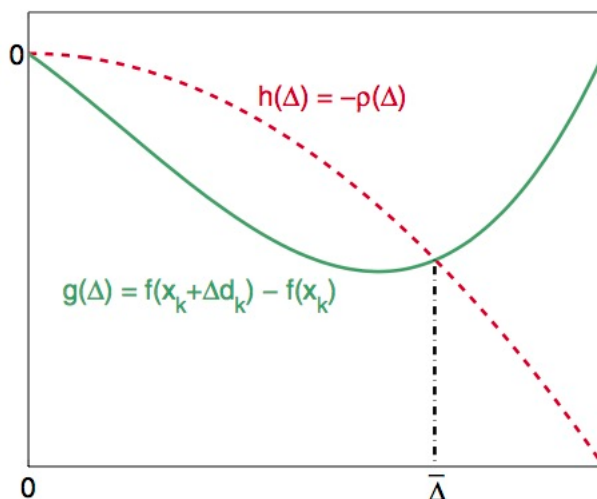
## 2.8. Ύπαρξη της υπακολουθίας

Σε αυτό το κεφάλαιο θα οριστικοποιήσουμε τη σύγκλιση της ΑΜΠΣ, μετά και την απόδειξη της Πρότασης 2.7.1. Όπως είδαμε, θα πρέπει να βρούμε μια ακολουθία δεικτών  $(k_n)$  από το σύνολο  $U$  ώστε να ισχύει  $\lim \Delta_{k_n} = 0$ . Επειδή  $\Delta_n > 0$  για κάθε  $n \in \mathbb{N}$ , αυτό είναι ισοδύναμο με το να δείξουμε ότι

$$\liminf \Delta_n = 0. \quad (2.20)$$

Ο στόχος μας, λοιπόν, είναι η εύρεση κατάλληλων συνθηκών ώστε να ισχύει η (2.20). Για την ακρίβεια, είναι εντυπωσιακό ότι η ΑΜΠΣ, από την κατασκευή της, μπορεί να εξασφαλίσει την (2.20) κάτω από πολύ

ασθενείς συνθήκες. Για να το διαπιστώσουμε αυτό, θεωρούμε τις συναρτήσεις  $g(\Delta) := f(\mathbf{x}_n + \Delta \mathbf{d}_n) - f(\mathbf{x}_n)$ ,  $h(\Delta) := -\rho(\Delta)$ . Η συμπεριφορά αυτών των δύο συναρτήσεων φαίνεται στο ακόλουθο σχήμα.



Βλέπουμε λοιπόν ότι όταν το βήμα ικανοποιεί  $0 < \Delta_n < \bar{\Delta}$ , τότε ισχύει  $g(\Delta_n) < h(\Delta_n)$  ή, ισοδύναμα

$$f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) - f(\mathbf{x}_n) < -\rho(\Delta_n)$$

και άρα έχουμε μείωση της τιμής της  $f$  στα πλαίσια της ΑΜΠΣ. Το γεγονός αυτό είναι μια ένδειξη για τη σύγκλιση.

**2.8.1. ΠΡΟΤΑΣΗ.** Έστω ότι η αντικειμενική συνάρτηση  $f$  είναι κάτω φραγμένη. Τότε η ΑΜΠΣ παράγει επαναλήψεις για τις οποίες ικανοποιείται η (2.20).

**ΑΠΟΔΕΙΞΗ.** Προς απαγωγή σε άτοπο, υποθέτουμε ότι η (2.20) δεν ισχύει. Έπεται ότι η  $(\Delta_n)$  έχει ως κάτω φράγμα της έναν θετικό αριθμό  $K$ , δηλαδή  $\Delta_n \geq K > 0$  για κάθε  $n$ . Αυτό έχει δύο συνέπειες:

- (1) Το σύνολο δεικτών  $U$  των αποτυχημένων επαναλήψεων είναι πεπερασμένο, διότι διαφορετικά το  $\Delta_n$  θα γινόταν απεριόριστα μικρό. Άρα το σύνολο δεικτών  $S$  των επιτυχημένων επαναλήψεων είναι άπειρο και, για την ακρίβεια, τελικά όλες οι επαναλήψεις είναι επιτυχημένες.
- (2) Η συνάρτηση  $\rho$  είναι φθίνουσα και άρα  $\rho(\Delta_n) \geq \rho(K)$ .

Για  $n \in S$  έχουμε ότι

$$f(\mathbf{x}_{n+1}) - f(\mathbf{x}_n) = f(\mathbf{x}_n + \Delta_n \mathbf{d}_n) - f(\mathbf{x}_n) < -\rho(\Delta_n) \leq -\rho(K) < 0$$

Όπως είδαμε, η παραπάνω ανισότητα ισχύει τελικά για κάθε  $n$  και άρα  $\lim f(\mathbf{x}_n) = -\infty$ , άτοπο γιατί υποθέσαμε ότι η  $f$  είναι κάτω φραγμένη.  $\square$

Βλέπουμε, λοιπόν, ότι και μόνο η απλή μείωση των τιμών της  $f$  δίνει σύγκλιση στην ΑΜΠΣ, αρκεί η αντικειμενική συνάρτηση να είναι κάτω φραγμένη.

Θα διερευνήσουμε στη συνέχεια περαιτέρω δυνατότητες σύγκλισης στα πλαίσια της αναζήτησης με πλέγματα, όπως αναπτύχθηκε στην Παράγραφο 2.6.

**2.8.2. ΠΡΟΤΑΣΗ.** Έστω ότι η ΑΜΠΣ υλοποιείται μέσω πλεγμάτων. Αν το σύνολο στάθμης  $\mathcal{L}_f(\mathbf{x}_0)$  είναι φραγμένο, τότε ισχύει η (2.15).

**ΑΠΟΔΕΙΞΗ.** Από τις Παρατηρήσεις 2.6.4 έχουμε ότι το  $U$  είναι άπειρο σύνολο και το συμπέρασμα έπεται.  $\square$



## Αλγόριθμοι ΑΜΠΣ

### 3.1. Η ενισχυμένη μέθοδος Compass

Πρόκειται για μια παραλλαγή της μεθόδου Compass όπου η αναζήτηση στις βασικές κατευθύνσεις ενέχει επιπλέον και θέση αυτού που ονομάζουμε *διερευνητική κίνηση* (exploratory move) [10]. Πιο συγκεκριμένα, ως υποθέσουμε ότι η  $n$ -οστή επανάληψη είναι επιτυχημένη και η κατεύθυνση  $\varepsilon_i \mathbf{e}_i$ , όπου  $\varepsilon_i = 1$  ή  $-1$ , δίνει μείωση της τιμής της  $f$ . Τότε δεν ορίζουμε απευθείας ως επόμενη εκτίμηση τη  $\mathbf{x}_n + \Delta_n \varepsilon_i \mathbf{e}_i$ , όπως θα κάναμε στην απλή Compass, αλλά εκτελούμε επιπλέον ελέγχους στα διανύσματα της μορφής

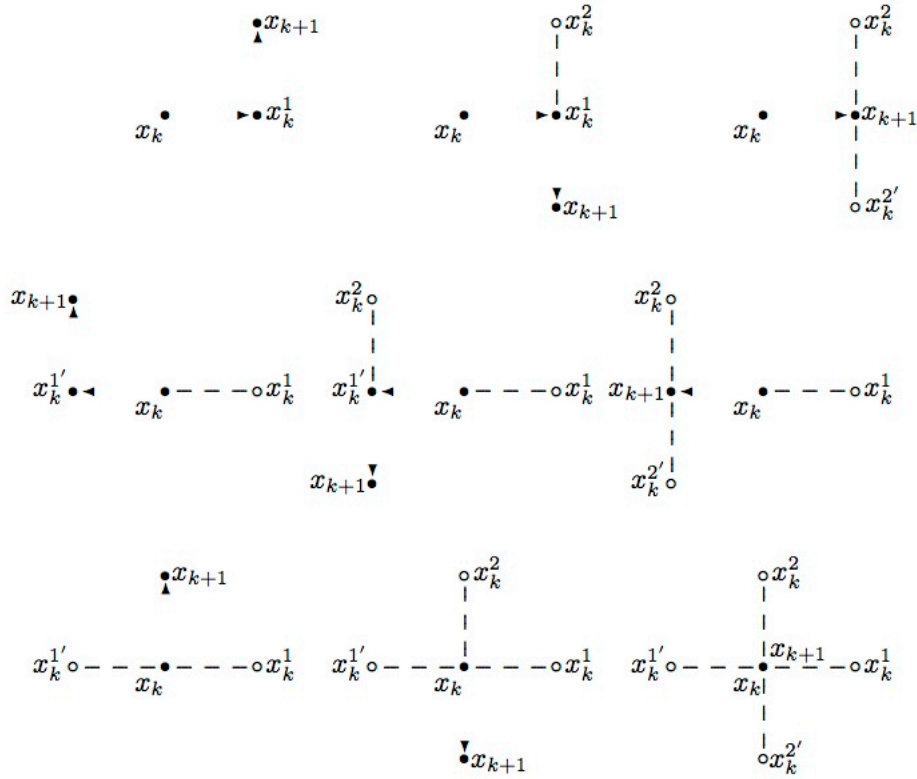
$$\mathbf{x}_n + \Delta_n \left( \varepsilon_i \mathbf{e}_i + a^{(i+1)} \mathbf{e}_{i+1} + \dots + a^{(N)} \mathbf{e}_N \right),$$

όπου  $a^{(j)} = -1, 0$  ή  $1$ . Λαμβάνουμε τότε ως  $\mathbf{x}_{n+1}$  εκείνο από αυτά τα διανύσματα που δίνει τη μικρότερη τιμή στην  $f$ . Στο σχήμα που ακολουθεί, συγκεντρώνονται όλα τα πιθανά σενάρια στην περίπτωση  $N = 2$ . Με μαύρη τελεία σημειώνεται η περίπτωση όπου μειώνεται η τιμή της  $f$ , το αντίθετο παριστάνει η άσπρη τελεία.

Η ενισχυμένη μέθοδος Compass ανήκει στις μεθόδους ΑΜΠΣ όπου η αναζήτηση γίνεται με μοτίβα. Μάλιστα, το μοτίβο που παράγει που παράγει τη συγκεκριμένη μέθοδο Compass είναι σταθερό και δίνεται από τους πίνακες  $B = I$  και τον πίνακα  $C$ , ο οποίος έχει  $N$  γραμμές και οι στήλες του αποτελούνται από όλους τους πιθανούς συνδυασμούς των  $-1, 0, 1$ . Έπεται ότι ο  $C$  έχει διάσταση  $N \times 3^N$  και μπορεί να γραφεί με τη μορφή  $C = [I \quad -I \quad L]$ , όπου ο  $L$  έχει διάσταση  $N \times (3^N - 2N)$  και περιέχει τις στήλες που έχουν δύο τουλάχιστον μη μηδενικά στοιχεία, καθώς και τη μηδενική στήλη. Στην περίπτωση  $N = 2$ , θα είναι

$$C = \begin{bmatrix} 1 & 0 & -1 & 0 & 1 & 1 & -1 & -1 & 0 \\ 0 & 1 & 0 & -1 & 1 & -1 & 1 & -1 & 0 \end{bmatrix}$$

Τα παράγον σύνολο είναι ανεξάρτητο του  $n$  και  $\mathcal{G}_n := \mathcal{D}_{\oplus}$ . Το βοηθητικό σύνολο  $\mathcal{H}_n$  είναι επίσης ανεξάρτητο του  $n$  και περιέχει τις στήλες του πίνακα  $L$ . Η εξέλιξη του βήματος  $\Delta_n$  γίνεται ακριβώς όπως και στην απλή Compass.



**3.1.1. Αλγόριθμος.** Είναι ο ακόλουθος:

Αρχικοποίηση

Δώσε τη συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ .

Δώσε την αρχική εκτίμηση  $x_0 \in \mathbb{R}^N$ .

Θέσε  $m = f(x_0)$ .

Δώσε το βήμα για το κριτήριο τερατισμού  $\Delta_{\text{tol}} > 0$ .

Δώσε το αρχικό βήμα  $\Delta_0 > \Delta_{\text{tol}}$ .

Όρισε το σύνολο κατευθύνσεων  $\mathcal{D}_{\oplus} := \{\pm \mathbf{e}_1, \pm \mathbf{e}_2, \dots, \pm \mathbf{e}_N\}$ .

Θέσε  $n := 0$ .

Κύριος βρόχος

Όσο ισχύει  $\Delta_n > \Delta_{\text{tol}}$ ,

Δευτερεύων βρόχος

Θέσε  $\mathbf{h} := 0$ .

Για  $i = 1, 2, \dots, N$ ,

Αν  $f(x_n + \Delta_n(\mathbf{h} + \mathbf{e}_i)) < m$ ,

θέσε  $\mathbf{h} = \mathbf{h} + \mathbf{e}_i$ ,

θέσε  $m = f(x_n + \mathbf{h})$ , και

πάρε επόμενο  $i$ .

Αν  $f(x_n + \Delta_n(\mathbf{h} - \mathbf{e}_i)) < m$ ,

θέσε  $\mathbf{h} = \mathbf{h} - \mathbf{e}_i$ ,

θέσε  $m = f(\mathbf{x}_n + \mathbf{h})$ , και  
 πάρε επόμενο  $i$ .  
 Θέσε  $\mathbf{x}_{n+1} := \mathbf{x}_n + \Delta_n \mathbf{h}$ .  
 Αν  $\mathbf{h} = 0$ , θέσε  $\Delta_{n+1} := \frac{\Delta_n}{2}$ .

Θέσε  $n = n + 1$ .

Έξοδος

Δώσε το  $\mathbf{x}_n$  ως λύση.

### 3.1.2. Κώδικας Matlab.

```

f=input('Give objective function: ');
x_0=input('Give initial approximation: ');
Delta_0=input('Give initial step: ');
Delta_tol=input('Give step tolerance: ');
nmax=input('Give maximum number of iterations: ');

N=length(x_0);
I=eye(N);

x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current);
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial);
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial);
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
  
```

```

end

if check==fmin
    Delta_current=0.5*Delta_current;
    nu=nu+1;
else
    x_current=x_next;
end

n=n+1;

end

display(' ');
display('Current iteration is: ');
fprintf('%10f \n',x_current);
display('Value of objective function is: ');
fprintf('%10f \n',fmin);
display('Current Delta is: ');
fprintf('%10f\n',Delta_current);
display('Number of iterations: ');
fprintf('%d\n',n);
display('Unsuccessful iterations: ');
fprintf('%d\n',nu);

```

### 3.2. Η εξελικτική μέθοδος του Box

Η μέθοδος αυτή εμφανίστηκε το 1957 στο άρθρο [2], σε μία μελέτη για την αύξηση της βιομηχανικής παραγωγής. Η αρχική ιδέα εντοπίζεται στον υπολογισμό των τιμών της συνάρτησης στις κορυφές ενός υπερκύβου με κέντρο την τρέχουσα επανάληψη  $x_n$ .

Πιο συγκεκριμένα, θεωρούμε ότι το γενικό σύνολο κατευθύνσεων  $G$  περιέχει τα  $2^N$  το πλήθος διανύσματα της μορφής

$$\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix},$$

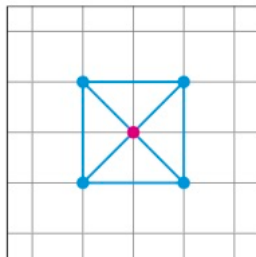
όπου  $\varepsilon_i = 1$  ή  $\varepsilon_i = -1$ . Λαμβάνουμε μία βάση  $B$  του  $\mathbb{R}^N$  που αποτελείται από  $N$  το πλήθος γραμμικά ανεξάρτητα διανύσματα εξ' αυτών. Το παράγον σύνολο  $G_n$  αποτελείται (ανεξαρτήτως του  $n$ ) από τα διανύσματα του  $B$  και του  $-B$ , ενώ το βοηθητικό σύνολο  $H_n$  από τα υπόλοιπα  $2^N - 2N$  διανύσματα. Το βήμα  $\Delta_n$  υποδιπλασιάζεται κάθε φορά που έχουμε αποτυχημένη επανάληψη.

Η μέθοδος του Box είναι ειδική περίπτωση της αναζήτησης με μοτίβα όπου για τον βασικό πίνακα ισχύει  $B = I$  και ο  $N \times (2^N + 1)$  γεννήτορας πίνακας  $C$  σταθερός και οι στήλες του περιέχουν όλους τους

δυνατούς συνδυασμούς των  $-1, 1$ , συν τη μηδενική στήλη. Ως  $N \times N$  πίνακα  $M$  μπορούμε να πάρουμε οποιονδήποτε που περιέχει  $n$  γραμμικά ανεξάρτητες στήλες του  $C$ . Στην περίπτωση  $N = 2$  θα είναι

$$C = \begin{bmatrix} 1 & 1 & -1 & -1 & 0 \\ 1 & -1 & -1 & 1 & 0 \end{bmatrix}$$

Στον  $\mathbb{R}^2$  υπάρχει μόνο παράγον σύνολο, όπως αναπαρίσταται στο ακόλουθο σχήμα.



**3.2.1. Αλγόριθμος.** Είναι ο ακόλουθος:

Αρχικοποίηση

Δώσε τη συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ .

Δώσε την αρχική εκτίμηση  $\mathbf{x}_0 \in \mathbb{R}^N$ .

Θέσε  $m = f(\mathbf{x}_0)$ .

Δώσε το βήμα για το κριτήριο τερματισμού  $\Delta_{\text{tol}} > 0$ .

Δώσε το αρχικό βήμα  $\Delta_0 > \Delta_{\text{tol}}$ .

Όρισε τον  $N \times N$  πίνακα  $A$  με στήλες συνδυασμούς των  $\pm 1$ .

Θέσε  $n := 0$ .

Κύριος βρόχος

Όσο ισχύει  $\Delta_n > \Delta_{\text{tol}}$ ,

Δευτερεύων βρόχος

Για  $i = 1, 2, \dots, 2^N$ ,

Θέσε  $\mathbf{u}_i$  την  $i$ -στήλη του πίνακα.

Αν  $f(\mathbf{x}_n + \Delta_n \mathbf{u}_i) < m$ ,

θέσε  $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta_n \mathbf{u}_i$ ,  $\Delta_{n+1} = \Delta_n$

θέσε  $m = f(\mathbf{x}_{n+1})$ ,

και βγες από το βρόχο.

Αν το  $m$  δεν έχει μειωθεί, θέσε  $\mathbf{x}_{n+1} = \mathbf{x}_n$ ,  $\Delta_{n+1} := \frac{\Delta_n}{2}$ .

Θέσε  $n = n + 1$ .

Έξοδος

Δώσε το  $\mathbf{x}_n$  ως λύση.

**3.2.2. Κώδικας Matlab.**

```
f=input('Give objective function: ');
x_0=input('Give initial approximation: ');
```

```
Delta_0=input('Give initial step: ');
Delta_tol=input('Give step tolerance: ');
nmax=input('Give maximum number of iterations: ');

N=length(x_0);
A=pernm([1 -1],N);
[r,c]=size(A);

x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current);
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;

    for i=1:r
        x_trial=x_current+Delta_current*A(i,:);
        fv=feval(f,x_trial);
        if fv<fmin
            x_next=x_trial;
            fmin=fv;
            continue
        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end

display(' ');
display('Current iteration is: ');
fprintf('%10f \n',x_current);
display('Value of objective function is: ');
fprintf('%10f \n',fmin);
display('Current Delta is: ');
fprintf('%10f\n',Delta_current);
display('Number of iterations: ');
fprintf('%d\n',n-1);
```

```
display('Unsuccessful iterations: ');
fprintf('%d\n',nu);
```

### 3.3. Η μέθοδος Hooke – Jeeves

Πρόκειται για μια βελτίωση της ενισχυμένης μεθόδου Compass που προτάθηκε με το άρθρο [6] και εισήγαγε για πρώτη φορά την έννοια του μοτίβου, αλλά και την έννοια της δοκιμαστικής κίνησης (exploratory move). Η ιδέα είναι η εξής: ας υποθέσουμε ότι εφαρμόζουμε την ενισχυμένη μέθοδο Compass, της οποίας η  $n$ -οστή επανάληψη είναι επιτυχής, οπότε από την εκτίμηση  $\mathbf{x}_n$  που βρισκόμαστε, παίρνουμε μια νέα εκτίμηση  $\tilde{\mathbf{x}}_{n+1}$ . Είναι λογικό να υποθέσουμε ότι η κατεύθυνση που ορίζει το διάνυσμα  $\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n$  μπορεί να δίνει σημαντική μείωση στην τιμή της  $f$  και προτείνεται έτσι η επόμενη διερεύνηση να γίνει όχι στην τρέχουσα επανάληψη  $\mathbf{x}_n$ , αλλά στο δοκιμαστικό σημείο

$$\mathbf{x}_p := \mathbf{x}_n + \alpha(\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n).$$

Το  $\alpha$  ονομάζεται *συντελεστής επιτάχυνσης* και οι πλέον συνήθης επιλογή είναι η  $\alpha = 2$ , οπότε

$$\mathbf{x}_p = 2\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n.$$

Η διερεύνηση στο  $\mathbf{x}_p$  γίνεται επίσης με την ενισχυμένη Compass. Αν αυτή η διερεύνηση δώσει κάποιο σημείο  $\mathbf{x}_+$  ώστε  $f(\mathbf{x}_+) < f(\mathbf{x}_p)$ , είναι δηλαδή επιτυχής, τότε παίρνουμε  $\mathbf{x}_{n+1} = \mathbf{x}_+$ . Αν η διερεύνηση είναι ανεπιτυχής, τότε  $\mathbf{x}_{n+1} = \tilde{\mathbf{x}}_{n+1}$ .

Η μέθοδος Hooke – Jeeves εμπίπτει προφανώς στην αναζήτηση με μοτίβα, και μάλιστα η δοκιμαστική κίνηση που περιγράψαμε αναφέρεται ως *κίνηση μοτίβου*. Στην περίπτωση όπου  $N = 2$ , τα παράγοντα σύνολα είναι πάντοτε

$$\mathcal{G}_n = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}.$$

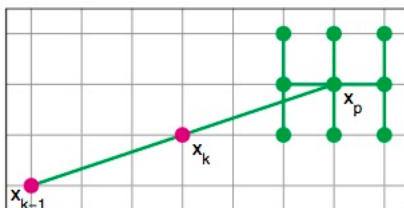
Επειδή ξεκινάμε με την ενισχυμένη Compass, θα είναι επίσης

$$\mathcal{H}_0 := \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.$$

Έστω τώρα  $n = 1, 2, \dots$ . Αν η  $(n-1)$ -επανάληψη έγινε με την ενισχυμένη Compass και ήταν επιτυχημένη, τότε το βοηθητικό σύνολο ορίζεται ως

$$\mathcal{H}_n := \left\{ \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}, \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 4 \\ 2 \end{pmatrix} \right\},$$

και προσοχή τώρα στο γεγονός ότι η αναζήτηση γίνεται στο  $\mathbf{x}_n$  σύμφωνα με το παρακάτω σχήμα



Στο επόμενο βήμα εφαρμόζεται η ενισχυμένη Compass, ανεξαρτήτως επιτυχούς ή ανεπιτυχούς έκβασης, και άρα

$$\mathcal{H}_{n+1} := \mathcal{H}_0.$$

**3.3.1. Αλγόριθμος.** Είναι ο ακόλουθος:

Αρχικοποίηση

Δώσε τη συνάρτηση  $f : \mathbb{R}^N \rightarrow \mathbb{R}$ .

Δώσε την αρχική εκτίμηση  $\mathbf{x}_0 \in \mathbb{R}^N$ .

Θέσε  $m = f(\mathbf{x}_0)$ .

Δώσε το βήμα για το κριτήριο τερματισμού  $\Delta_{\text{tol}} > 0$ .

Δώσε το αρχικό βήμα  $\Delta_0 > \Delta_{\text{tol}}$ .

Όρισε το σύνολο κατευθύνσεων  $\mathcal{D}_{\oplus} := \{\pm \mathbf{e}_1, \pm \mathbf{e}_2, \dots, \pm \mathbf{e}_N\}$ .

Θέσε  $n := 0$ .

Κύριος βρόχος

Όσο ισχύει  $\Delta_n > \Delta_{\text{tol}}$ ,

Δοκιμαστική κίνηση

Εφάρμοσε ενισχυμένη Compass στην τρέχουσα επανάληψη  $\mathbf{x}_n$ .

Αν προκύψει επιτυχής επανάληψη  $\tilde{\mathbf{x}}_{n+1}$ ,

θέσε  $\mathbf{x}_p = 2\tilde{\mathbf{x}}_{n+1} - \mathbf{x}_n$ , και

πήγαινε στον δευτερεύοντα βρόχο.

Αν η επανάληψη είναι ανεπιτυχής, αγνόησε τον δευτερεύοντα βρόχο.

Δευτερέων βρόχος

Εφάρμοσε ενισχυμένη Compass στο  $\mathbf{x}_p$ .

Θέσε  $n = n + 1$ .

Έξοδος

Δώσε το  $\mathbf{x}_n$  ως λύση.

**3.3.2. Κώδικας Matlab.**

```
f=input('Give objective function: ');
x_0=input('Give initial approximation: ');
Delta_0=input('Give initial step: ');
Delta_tol=input('Give step tolerance: ');
nmax=input('Give maximum number of iterations: ');
```

```
N=length(x_0);
```

```
I=eye(N);
```

```
x_current=x_0;
```

```
Delta_current=Delta_0;
```

```
fmin=feval(f,x_current);
```

```
n=1;
```

```
nu=0;
```



```

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial);
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial);
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
end

if check==fmin
    Delta_current=0.5*Delta_current;
    nu=nu+1;
else
    x_p=2*x_next-x_current;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_p+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial);
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_p+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial);
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
end

```

```
        end
        end

        x_current=x_next;
    end

    n=n+1;

end

display(' ');
display('Current iteration is: ');
fprintf('%10f \n',x_current);
display('Value of objective function is: ');
fprintf('%10f \n',fmin);
display('Current Delta is: ');
fprintf('%10f\n',Delta_current);
display('Number of iterations: ');
fprintf('%d\n',n);
display('Unsuccessful iterations: ');
fprintf('%d\n',nu);
```

## ΚΕΦΑΛΑΙΟ 4

### Αριθμητικά πειράματα

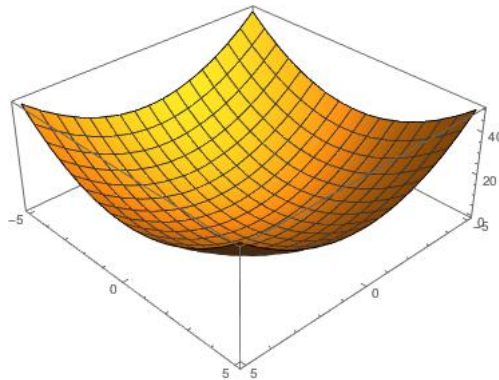
#### 4.1. Συναρτήσεις δοκιμής για προβλήματα βελτιστοποίησης

Οι μέθοδοι που αναπτύξαμε θα ελεγχθούν με τις ακόλουθες τέσσερις συναρτήσεις στον  $\mathbb{R}^N$ , οι οποίες έχουν ένα εκ των προτέρων γνωστό ολικό ελάχιστο. Οι συναρτήσεις αυτές συναντώνται συχνά στη βιβλιογραφία, π.χ. [8].

**4.1.1. Η σφαιρική συνάρτηση.** Η συνάρτηση αυτή στον  $\mathbb{R}^2$  ορίζεται ως

$$f(x^{(1)}, x^{(2)}) = (x^{(1)})^2 + (x^{(2)})^2.$$

Η  $f$  παρουσιάζει ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}) = (0, 0)$  την τιμή  $f(0, 0) = 0$  και δεν έχει άλλα τοπικά ελάχιστα.



Η σφαιρική συνάρτηση γενικεύεται με προφανή τρόπο στον  $\mathbb{R}^N$ ,

$$f(x^{(1)}, x^{(2)}, \dots, x^{(N)}) = (x^{(1)})^2 + (x^{(2)})^2 + \dots + (x^{(N)})^2 = \sum_{i=1}^N (x^{(i)})^2,$$

με ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}, \dots, x_*^{(N)}) = (0, 0, \dots, 0)$  την τιμή  $f(\mathbf{x}_*) = 0$ .

Η σφαιρική συνάρτηση υλοποιείται στο Matlab με τον ακόλουθο κώδικα:

```
function [y]=spherical(x)
```

```
N=length(x);
```

```
y=0;
```

```

for i=1:N
y=y+x(i)^2;
end

end

```

**4.1.2. Η συνάρτηση Rosenbrock.** Η συνάρτηση αυτή στον  $\mathbb{R}^2$  ορίζεται ως εξής:

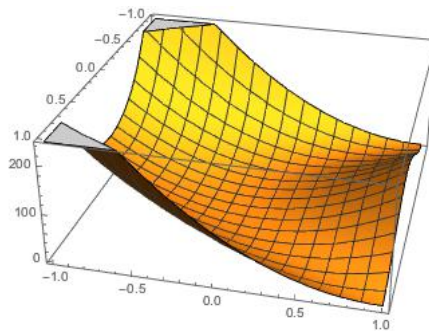
$$f(x^{(1)}, x^{(2)}) = \beta \left( x^{(2)} - \left( x^{(1)} \right)^2 \right)^2 + \left( x^{(1)} - \alpha \right),$$

όπου  $\alpha, \beta$  είναι θετικές παράμετροι. Η  $f$  παρουσιάζει ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}) = (\alpha, \alpha^2)$  την τιμή  $f(\alpha, \alpha^2) = 0$  και υπάρχουν επίσης πολλά τοπικά ελάχιστα. Το ολικό ελάχιστο είναι δύσκολο να προσεγγιστεί και αυτό καθιστά τη συνάρτηση Rosenbrock μια φιλόδοξη επιλογή για τα προβλήματα βελτιστοποίησης.

Συνήθως χρησιμοποιούμε τις τιμές  $\alpha = 1, \beta = 100$ , οπότε

$$f(x^{(1)}, x^{(2)}) = 100 \left( x^{(2)} - \left( x^{(1)} \right)^2 \right)^2 + \left( x^{(1)} - 1 \right)^2,$$

με ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}) = (1, 1)$ .



Η συνάρτηση Rosenbrock γενικεύεται στον  $\mathbb{R}^N$ ,

$$f(x^{(1)}, x^{(2)}, \dots, x^{(N)}) = \sum_{i=1}^{N-1} \left[ 100 \left( x^{(i+1)} - \left( x^{(i)} \right)^2 \right)^2 + \left( x^{(i)} - 1 \right)^2 \right],$$

με ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}, \dots, x_*^{(N)}) = (1, 1, \dots, 1)$  την τιμή  $f(\mathbf{x}_*) = 0$ .

Η συνάρτηση Rosenbrock υλοποιείται στο Matlab με τον ακόλουθο κώδικα:

```

function [y]=rosenbrock(x)

N=length(x);
y=0;

for i=1:(N-1)

```

```

y=y+100*(x(i+1)-x(i)^2)^2+(x(i)-1)^2;
end

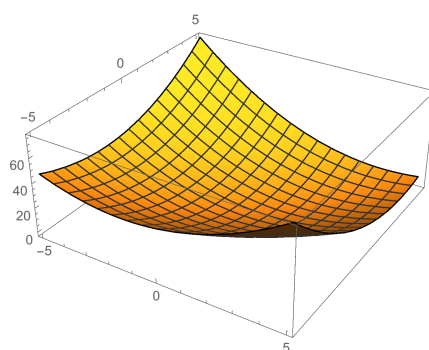
end

```

**4.1.3. Η συνάρτηση Trid.** Ορίζεται στον  $\mathbb{R}^2$  με τον τύπο

$$f(x^{(1)}, x^{(2)}) = (x^{(1)} - 1)^2 + (x^{(2)} - 1)^2 - x^{(1)}x^{(2)}.$$

Η  $f$  παρουσιάζει ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}) = (2, 2)$  την τιμή  $f(2, 2) = -2$  και δεν υπάρχουν άλλα τοπικά ελάχιστα.



Η συνάρτηση Trid γενικεύεται στον  $\mathbb{R}^N$  με τη βοήθεια του τύπου

$$f(x^{(1)}, x^{(2)}, \dots, x^{(N)}) = \sum_{i=1}^N (x^{(i)} - 1)^2 - \sum_{i=1}^{N-1} x^{(i)}x^{(i+1)},$$

και παρουσιάζει ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}, \dots, x_*^{(N)})$ , όπου

$$x^{(i)} = i(N + 1 - i), \quad i = 1, 2, \dots, N,$$

την τιμή

$$f(\mathbf{x}_*) = -\frac{N(N-1)(N+4)}{6}.$$

Η συνάρτηση Trid υλοποιείται στο Matlab με τον ακόλουθο κώδικα:

```

function[y]=trid(x)

N=length(x);
y=(x(N)-1)^2;

for i=1:(N-1)
y=y+(x(i)-1)^2-x(i)*x(i+1);
end

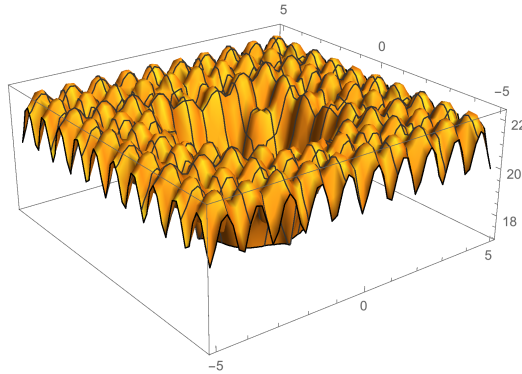
end

```

#### 4.1.4. Η συνάρτηση Ackley. Ορίζεται στον $\mathbb{R}^2$ με τύπο

$$f(x^{(1)}, x^{(2)}) = -20e^{-0.2\sqrt{\frac{1}{2}[(x^{(1)})^2 + (x^{(2)})^2]}} - e^{\frac{1}{2}[\cos(2\pi x^{(1)}) + \cos(2\pi x^{(2)})]} + e + 20.$$

Η  $f$  παρουσιάζει ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}) = (0, 0)$  την τιμή  $f(0, 0) = 0$  και έχει πάρα πολλά άλλα τοπικά ελάχιστα. Το γεγονός αυτό καθιστά δύσκολο τον εντοπισμό του ολικού ελαχίστου και για τον λόγο αυτόν η συνάρτηση Ackley χρησιμοποιείται ευρέως για τον έλεγχο μιας μεθόδου ελαχιστοποίησης.



Η συνάρτηση Ackley γενικεύεται κατά προφανή τρόπο στον  $\mathbb{R}^N$  με τον τύπο

$$f(x^{(1)}, x^{(2)}, \dots, x^{(N)}) = -20e^{-0.2\sqrt{\frac{1}{N}\sum_{i=1}^N (x^{(i)})^2}} - e^{\frac{1}{N}\sum_{i=1}^N \cos(2\pi x^{(i)})} + e + 20,$$

με ολικό ελάχιστο στο  $\mathbf{x}_* = (x_*^{(1)}, x_*^{(2)}, \dots, x_*^{(N)}) = (0, 0, \dots, 0)$  την τιμή  $f(\mathbf{x}_*) = 0$ .

Η συνάρτηση Trid υλοποιείται στο Matlab με τον ακόλουθο κώδικα:

```
function [y]=auckley(x)

N=length(x);
s=0;
c=0;

for i=1:N
    s=s+x(i)^2;
    c=c+cos(2*pi*x(i));
end

y=-20*exp(-0.2*sqrt(1/N*s))-exp(1/N*c)+exp(1)+20;
```

## 4.2. Μεθοδολογία

Χρησιμοποιούμε το κύριο script `all_methods`, το οποίο υλοποιεί στο Matlab ταυτόχρονα τις τέσσερις μεθόδους (Compass, Ενισχυμένη Compass, Box, Horoke – Jeeves). Το πρόγραμμα ζητάει ως εισόδους τα εξής:

- Την αντικειμενική συνάρτηση  $f$  (objective function).

- Το γνωστό σημείο όπου παρουσιάζεται το ολικό ελάχιστο της αντικειμενικής συνάρτησης (point of global minimum). Από αυτό το δεδομένο αποφασίζεται και η διάσταση του προβλήματος.
- Την ακτίνα (radius) εντός της οποίας θα ληφθεί τυχαία η αρχική εκτίμηση.
- Το αρχικό βήμα (initial step)  $\Delta_0$ . Επιλέγεται η τιμή 0.1.
- Το κατώφλι τερματισμού (step tolerance)  $\Delta_{tol}$ . Επιλέγεται η τιμή 0.00001.
- Τον μέγιστο αριθμό επαναλήψεων (maximum number of iterations) προκειμένου να αποφευχθεί ατέρμονας βρόχος.

Μετά τους υπολογισμούς, το πρόγραμμα εμφανίζει τα εξής:

- Την διάσταση του προβλήματος (dimension).
- Την απόσταση του αρχικού σημείου από το σημείο ελαχίστου (initial distance).
- Την ακρίβεια της μεθόδου (accuracy), εκφρασμένη ως την απόσταση της τελευταίας επανάληψης από το σημείο του ελαχίστου.
- Τον αριθμό των επαναλήψεων που χρειάστηκαν (iterations).
- Το τρέχον βήμα (delta).
- Την τιμή της αντικειμενικής συνάρτησης στο εν λόγω σημείο (value of objective function).
- Τον αριθμό των κλήσεων της συνάρτησης (calls).

Οι αποστάσεις εκφράζονται στην ευκλείδεια νόρμα. Εξετάζουμε κάθε συνάρτηση δοκιμής ξεχωριστά και για διαστάσεις  $N = 2, 5, 20$ .

### 4.3. Βελτιστοποίηση της σφαιρικής συνάρτησης

Η σφαιρική συνάρτηση είναι μια «εύκολη» περίπτωση ελαχιστοποίησης. Επιλέγουμε, έτσι, μια μεγάλη ακτίνα, ίση με 50, για την τυχαία επιλογή του αρχικού σημείου και ως μέγιστο αριθμό επαναλήψεων ορίζουμε τις 1000. Τα εξαγόμενα του Matlab έχουν ως ακολούθως.

```
DIMENSION N=2
Initial distance: 20.557343
```

```
ans =
```

	Accuracy	Iterations	Delta	Function_Value	Calls
Compass	6.4048e-06	317	6.1035e-06	4.1022e-11	1261
Enhanced Compass	6.4048e-06	177	6.1035e-06	4.1022e-11	702
Box	6.4048e-06	181	6.1035e-06	4.1022e-11	725
Hooke-Jeeves	6.4048e-06	105	6.1035e-06	4.1022e-11	732

DIMENSION N=5  
Initial distance: 29.083470

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
Compass	6.195e-06	648	6.1035e-06	3.8378e-11	6461
Enhanced Compass	6.195e-06	205	6.1035e-06	3.8378e-11	2035
Box	1.4863e-05	209	6.1035e-06	2.209e-10	6689
Hooke-Jeeves	6.195e-06	124	6.1035e-06	3.8378e-11	2264

DIMENSION N=20  
Initial distance: 30.748884

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
Compass	5.6895	1000	0.1	32.37	40001
Enhanced Compass	1.4589e-05	134	6.1035e-06	2.1284e-10	5303
Box	2.9351e-05	142	6.1035e-06	8.6147e-10	1.489e+08
Hooke-Jeeves	1.4589e-05	75	6.1035e-06	2.1284e-10	5339

Έχουμε να κάνουμε τις εξής παρατηρήσεις.

1. Με τη σειρά που αναφέρονται, οι μέθοδοι Hooke – Jeeves και Enhanced Compass είναι οι καλύτερες εκ των τεσσάρων. Έχουν πολύ καλή σύγκλιση σε όλες τις διαστάσεις και ο είναι εντυπωσιακό ότι ο αριθμός των επαναλήψεων μειώνεται όσο αυξάνεται η διάσταση. Η Hooke – Jeeves απαιτεί σχεδόν μισές επαναλήψεις απ' ότι η Enhanced Compass, ενώ ο (αυξανόμενος) αριθμός των κλήσεων είναι συγκρίσιμος.

2. Η μέθοδος Compass δίνει πολύ ικανοποιητικά αποτελέσματα στις 2 διαστάσεις, με λογικό αριθμό επαναλήψεων και κλήσεων συνάρτησης. Προχωρώντας στη διάσταση 5, έχουμε πάλι πολύ καλή σύγκλιση, αλλά με αρκετά περισσότερες επαναλήψεις και κλήσεις. Στη διάσταση 20, η Compass αποτυγχάνει τελείως, εξαντλώντας, μάλιστα, όλες τις διαθέσιμες επαναλήψεις. Επιπλέον, όλες αυτές οι επαναλήψεις είναι επιτυχημένες.

3. Η μέθοδος Box έχει πολύ καλά αποτελέσματα σε όλες τις διαστάσεις. Όμως, όσο μεγαλώνει η διάσταση, παρόλο που επίσης απαιτούνται όλο και λιγότερες επαναλήψεις, χρειάζεται πολλούς υπολογισμούς της συνάρτησης και γίνεται εξαιρετικά αργή.

#### 4.4. Βελτιστοποίηση της συνάρτησης Rosenbrock

Η συνάρτηση Rosenbrock θεωρείται ως μία από τις πιο «δύσκολες» περιπτώσεις ελαχιστοποίησης. Η ύπαρξη πολλών τοπικών ελαχίστων και η γεωμετρία του γραφήματος απαιτούν μικρή ακτίνα και πολλές επαναλήψεις (και συνακόλουθα πολλές κλήσεις της συνάρτησης) για να επιτευχθεί κάποιο λογικό αποτέλεσμα. Ως παράδειγμα, επιχειρούμε την ελαχιστοποίηση για ακτίνα ίση με 5, με μέγιστο αριθμό επαναλήψεων τις 5000, στη διάσταση 2.



DIMENSION N=2  
Initial distance: 3.334013

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	0.012873	5000	1.2207e-05	3.2973e-05	20000
<b>Enhanced Compass</b>	0.1129	5000	9.7656e-05	0.0024496	19976
<b>Box</b>	1.8181	5000	0.00019531	0.41356	20001
<b>Hooke-Jeeves</b>	0.0070101	4406	6.1035e-06	9.7965e-06	32532

Βλέπουμε ότι όλες οι μέθοδοι, πλην της Hooke – Jeeves, εξαντλούν τις δυνατότητές τους, κινούμενες όμως προς τη σωστή κατεύθυνση. Για το λόγο αυτό, επιλέγουμε ακτίνα ίση με 2 και μέγιστο αριθμό επαναλήψεων στις 10000.

DIMENSION N=2  
Initial distance: 1.042965

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	0.0064943	6660	6.1035e-06	8.4108e-06	26640
<b>Enhanced Compass</b>	0.0064943	5206	6.1035e-06	8.4108e-06	20820
<b>Box</b>	0.007062	6508	6.1035e-06	9.9496e-06	26033
<b>Hooke-Jeeves</b>	0.0064943	2520	6.1035e-06	8.4108e-06	17604

DIMENSION N=5  
Initial distance: 0.833229

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	0.0091565	8281	6.1035e-06	2.0869e-05	82796
<b>Enhanced Compass</b>	0.0091565	5052	6.1035e-06	2.0869e-05	50512
<b>Box</b>	0.012924	6726	6.1035e-06	4.1391e-05	2.1523e+05
<b>Hooke-Jeeves</b>	0.0091565	1611	6.1035e-06	2.0869e-05	31092

DIMENSION N=20  
Initial distance: 1.103075

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	0.0095127	8031	6.1035e-06	2.2711e-05	3.192e+05
<b>Enhanced Compass</b>	0.0095127	4112	6.1035e-06	2.2711e-05	1.6442e+05
<b>Box</b>	0.02004	1886	6.1035e-06	0.0001012	1.9776e+09
<b>Hooke-Jeeves</b>	0.0078967	1468	6.1035e-06	1.5714e-05	1.1592e+05

Η ακρίβεια δεν είναι τόσο καλή όσο στις άλλες συναρτήσεις, για καμία διάσταση και καμία μέθοδο. Η Hooke – Jeeves παραμένει η καλύτερη επιλογή, με δεύτερη στη σειρά τη Box. Για την πρώτη μέθοδο, όπως και

για την ενισχυμένη Compass, ο αριθμός των επαναλήψεων μειώνεται όσο αυξάνει η διάσταση. Η απλή Compass ανταποκρίνεται σχετικά καλά.

#### 4.5. Βελτιστοποίηση της συνάρτησης Trid

Η σφαιρική συνάρτηση είναι μια ακόμη σχετικά «εύκολη» περίπτωση ελαχιστοποίησης. Η διαφορά με τη σφαιρική συνάρτηση είναι ότι η θέση και η τιμή του ολικού ελαχίστου μεταβάλλονται με τη διάσταση. Επιλέγουμε ως ακτίνα για την τυχαία επιλογή του αρχικού σημείου την 20 και ως μέγιστο αριθμό επαναλήψεων τον 1000. Τα εξαγόμενα του Matlab έχουν ως ακολούθως.

DIMENSION N=2  
Initial distance: 2.881161

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	5.5418e-06	68	6.1035e-06	-2	269
<b>Enhanced Compass</b>	1.1755e-05	47	6.1035e-06	-2	185
<b>Box</b>	8.7051e-06	46	6.1035e-06	-2	185
<b>Hooke-Jeeves</b>	1.1755e-05	35	6.1035e-06	-2	216

DIMENSION N=5  
Initial distance: 9.879002

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	9.0135e-06	286	6.1035e-06	-30	2854
<b>Enhanced Compass</b>	9.0135e-06	114	6.1035e-06	-30	1091
<b>Box</b>	1.5638e-05	105	6.1035e-06	-30	3361
<b>Hooke-Jeeves</b>	9.0135e-06	71	6.1035e-06	-30	1200

DIMENSION N=20  
Initial distance: 11.763658

ans =

	<u>Accuracy</u>	<u>Iterations</u>	<u>Delta</u>	<u>Function_Value</u>	<u>Calls</u>
<b>Compass</b>	1.5111	1000	0.0125	-1520	39815
<b>Enhanced Compass</b>	0.00091191	411	6.1035e-06	-1520	16330
<b>Box</b>	9.2842e-05	108	6.1035e-06	-1520	1.1325e+08
<b>Hooke-Jeeves</b>	0.00091191	185	6.1035e-06	-1520	13960

Ισχύουν παρόμοιες παρατηρήσεις με τη σφαιρική συνάρτηση, με τις εξής διαφορές:

1. Οι επαναλήψεις της Hooke – Jeeves και της Enhanced Compass και της Box αυξάνονται με τη διάσταση. Επίσης, στη διάσταση 20, η ακρίβεια των δύο πρώτων μεθόδων δεν είναι ιδιαίτερα καλή, αλλά η τιμή της συνάρτησης πετυχαίνει το ελάχιστο.

2. Στη διάσταση 20 η μέθοδος Box δίνει τα καλύτερα αποτελέσματα, ενώ και η Compass συμπεριφέρεται καλά.

#### 4.6. Βελτιστοποίηση της συνάρτησης Ackley

Η συνάρτηση Ackley έχει πάρα πολλά τοπικά ελάχιστα, με αποτέλεσμα να «μπερδεύει» τις μεθόδους ελαχιστοποίησης. Για το λόγο αυτό, αν ξεκινήσουμε μακριά από το ολικό ελάχιστο, δεν αναμένεται σύγκλιση σε αυτό. Ως παράδειγμα, επιχειρούμε την ελαχιστοποίηση για ακτίνα ίση με 5 στη διάσταση 2.

DIMENSION N=2  
Initial distance: 3.981945

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
<b>Compass</b>	4.2133	33	6.1035e-06	9.0011	120
<b>Enhanced Compass</b>	4.2133	26	6.1035e-06	9.0011	94
<b>Box</b>	4.2133	32	6.1035e-06	9.0011	129
<b>Hooke-Jeeves</b>	4.2133	25	6.1035e-06	9.0011	135

Βλέπουμε ότι έχουμε ταχύτατη σύγκλιση, αλλά σε κάποιο άλλο σημείο που είναι προφανώς τοπικό ελάχιστο. Για το λόγο αυτό, ξεκινάμε με μια πολύ μικρή ακτίνα, ίση με 1, και ορίζουμε ξανά ως μέγιστο αριθμό επαναλήψεων το 1000.

DIMENSION N=2  
Initial distance: 0.337164

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
<b>Compass</b>	3.9425e-06	36	6.1035e-06	1.1151e-05	134
<b>Enhanced Compass</b>	3.9425e-06	28	6.1035e-06	1.1151e-05	104
<b>Box</b>	8.6398e-06	31	6.1035e-06	2.4439e-05	125
<b>Hooke-Jeeves</b>	3.9425e-06	28	6.1035e-06	1.1151e-05	159

DIMENSION N=5  
Initial distance: 0.592987

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
<b>Compass</b>	6.1767e-06	56	6.1035e-06	1.105e-05	538
<b>Enhanced Compass</b>	6.1767e-06	30	6.1035e-06	1.105e-05	285
<b>Box</b>	1.5262e-05	34	6.1035e-06	2.7303e-05	1089
<b>Hooke-Jeeves</b>	6.1767e-06	28	6.1035e-06	1.105e-05	401

DIMENSION N=20  
Initial distance: 0.508501

ans =

	Accuracy	Iterations	Delta	Function_Value	Calls
Compass	1.7879e-05	169	6.1035e-06	1.5992e-05	6593
Enhanced Compass	1.7879e-05	29	6.1035e-06	1.5992e-05	1095
Box	2.9428e-05	32	6.1035e-06	2.6324e-05	3.3554e+07
Hooke-Jeeves	1.7879e-05	29	6.1035e-06	1.5992e-05	1695

Η σύγκλιση για όλες τις μεθόδους και για όλες τις διαστάσεις είναι πολύ καλή και γρήγορη.

#### 4.7. Εξάρτηση των μεθόδων από τις επαναλήψεις

Σε αυτή και την επόμενη παράγραφο, θα συνεχίσουμε τα αριθμητικά πειράματα από μια διαφορετική σκοπιά. Πρώτα, θα σταθεροποιήσουμε την ακτίνα σε μια μεγάλη τιμή, τη 10 ακριβώς και, ξεκινώντας από το 2000, θα αυξάνουμε κατά 2000 τις επαναλήψεις μέχρι να πετύχουμε σύγκλιση όλων των μεθόδων για τη συνάρτηση Rosenbrock. Επειδή η μέθοδος Box είναι πολύ αργή στις μεγάλες διαστάσεις, δουλεύουμε για  $N = 2$ .

Η ανάλυση γίνεται με το script `all_methods_iter`. Ο παρακάτω πίνακας μας δίνει τον αριθμό των επαναλήψεων που χρειάστηκε η κάθε μέθοδος για τον αντίστοιχο μέγιστο επιτρεπτό.

```
Iterations needed for convergence
```

ans =

Max_Allowed	Compass	Enh_Compass	Box	Hooke_Jeeves
2000	2000	2000	2000	2000
4000	4000	4000	4000	4000
6000	6000	6000	6000	4544
8000	5527	8000	8000	4840
10000	10000	9233	10000	4102
12000	9753	9548	12000	4669
14000	12077	8753	14000	3352
16000	10611	9749	16000	4528
18000	11347	9291	18000	4126
20000	10871	9463	20000	4414
22000	10465	9642	22000	4474
24000	11086	9611	24000	4470
26000	10031	9781	23770	4770
28000	8085	8920	21934	4890
30000	8108	8499	21542	4938
32000	10035	9232	23299	4770
34000	8843	9257	23127	4783
36000	12392	8462	21677	3432
38000	6999	8594	19751	4992
40000	11518	9406	23912	4167

Ακολουθεί ο πίνακας με τις αντίστοιχες τιμές της αντικειμενικής συνάρτησης.

Corresponding function value

ans =

Max_Allowed	Compass	Enh_Compass	Box	Hooke_Jeeves
2000	4.1433	3.9992	4.9801	2.0359
4000	0.10216	0.1876	1.2044	0.00015218
6000	0.014235	0.016522	1.0039	1.1238e-05
8000	7.3086e-06	1.7704e-05	0.39131	7.3086e-06
10000	0.000102	1.2471e-05	0.57117	1.2471e-05
12000	1.1024e-05	1.1024e-05	0.48951	1.1024e-05
14000	9.7543e-06	9.7543e-06	0.13515	9.7543e-06
16000	9.0486e-06	9.0486e-06	0.21372	9.0486e-06
18000	7.4998e-06	7.4998e-06	0.11502	7.4998e-06
20000	1.1772e-05	1.1772e-05	0.03954	1.1772e-05
22000	1.1561e-05	1.1561e-05	0.0012296	1.1561e-05
24000	9.7741e-06	9.7741e-06	1.8301e-05	9.7741e-06
26000	7.46e-06	7.46e-06	9.2776e-06	7.46e-06
28000	9.288e-06	9.288e-06	1.0505e-05	9.288e-06
30000	1.3082e-05	1.3082e-05	1.3258e-05	1.3082e-05
32000	1.1882e-05	1.1882e-05	1.2388e-05	1.1882e-05
34000	1.2837e-05	1.2837e-05	1.3186e-05	1.2837e-05
36000	1.0537e-05	1.0537e-05	1.1501e-05	1.0537e-05
38000	9.179e-06	9.179e-06	1.0389e-05	9.179e-06
40000	8.4272e-06	8.4272e-06	9.8908e-06	8.4272e-06

Από τα παραπάνω, γίνεται σαφές ότι η κατάσταση αρχίζει να ξεκαθαρίζεται:

- για την Compass, μετά τις 12000 επαναλήψεις,
- για την ενισχυμένη Compass, μετά τις 10000 επαναλήψεις,
- για τη Box, μετά τις 24000 επαναλήψεις, και
- για τη Hooke – Jeeves, μετά τις 4000 επαναλήψεις.

#### 4.8. Εξάρτηση των μεθόδων από την ακτίνα

Τώρα θα σταθεροποιήσουμε τις επαναλήψεις σε μια μικρή τιμή, τη 1000 και ξεκινώντας από ακτίνα ακριβώς 10, θα την υποδιπλασιάζουμε μέχρι να πετύχουμε σύγκλιση όλων των μεθόδων για τη συνάρτηση Rosenbrock. Θα δουλέψουμε και πάλι στη διάσταση,  $N = 2$ .

Η ανάλυση γίνεται με το script `all_methods_radius`. Ο παρακάτω πίνακας μας δίνει τον αριθμό των επαναλήψεων που χρειάστηκε η κάθε μέθοδος για τον αντίστοιχο μέγιστο επιτρεπτό.

Iterations needed for convergence

ans =

Radius	Compass	Enh_Compass	Box	Hooke_Jeeves
10	1000	1000	1000	1000
5	1000	1000	1000	1000
2.5	1000	1000	1000	1000
1.25	1000	1000	1000	1000
0.625	1000	1000	1000	1000
0.3125	1000	1000	1000	1000
0.15625	1000	1000	1000	1000
0.078125	1000	1000	1000	1000
0.039062	1000	1000	1000	1000
0.019531	1000	943	401	466
0.0097656	1000	681	22	348
0.0048828	20	19	19	19
0.0024414	20	20	21	20
0.0012207	20	20	18	20
0.00061035	17	17	17	17
0.00030518	15	15	15	15

Ακολουθεί ο πίνακας με τις αντίστοιχες τιμές της αντικειμενικής συνάρτησης.

Corresponding function value

ans =

Radius	Compass	Enh_Compass	Box	Hooke_Jeeves
10	4.7216	4.2123	4.6312	2.8021
5	1.5825	1.7788	1.9726	1.2164
2.5	0.33754	0.40865	0.60551	0.11032
1.25	0.11194	0.087393	0.10883	0.0092883
0.625	0.015574	0.009741	0.02111	0.00075885
0.3125	0.0021197	0.00097908	7.3727e-05	0.00054191
0.15625	0.00042668	0.00018027	0.0020127	1.7013e-05
0.078125	0.00040087	0.00016657	4.5216e-05	1.2377e-05
0.039062	0.00027191	0.00012574	1.9398e-05	8.7699e-06
0.019531	2.0516e-05	8.3988e-06	9.9364e-06	8.3988e-06
0.0097656	8.3397e-06	8.1107e-06	1.2232e-06	8.1107e-06
0.0048828	8.469e-06	8.469e-06	9.2657e-07	8.3276e-06
0.0024414	2.9456e-06	2.9456e-06	4.622e-10	2.5804e-06
0.0012207	1.4357e-06	1.4357e-06	9.2303e-07	1.4066e-06
0.00061035	1.7233e-07	1.7233e-07	4.0812e-09	1.7233e-07
0.00030518	3.2486e-08	3.2486e-08	8.4281e-09	3.2486e-08

Παρατηρούμε ότι για να πετύχουμε καλή σύγκλιση σε 1000 ή λιγότερες επαναλήψεις, θα πρέπει η ακτίνα να γίνει μικρότερη

- περίπου από 0.01 για την Compass,
- περίπου από 0.2 για την ενισχυμένη Compass,
- περίπου από 0.04 για τη Box, και
- περίπου από 0.15 για τη Hooke – Jeeves.

## ΠΑΡΑΡΤΗΜΑ Α΄

### Κώδικες Matlab

#### Α΄.1. Το κύριο script all\_methods

```
f=input('Give objective function: ');
x_min=input('Give point of global minimum: ');
r=input('Give the radius for the initial approximation: ');
Delta_0=input('Give initial step: ');
Delta_tol=input('Give step tolerance: ');
nmax=input('Give maximum number of iterations: ');

N=length(x_min);
x_0=x_min+(r/sqrt(N))*rand(1,N);
I=eye(N);

Accuracy=zeros(4,1);
Iterations=zeros(4,1);
Delta=zeros(4,1);
Function_Value=zeros(4,1);
Calls=zeros(4,1);

% Compass Best Search
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;

    for i=1:N
        x_trial=x_current+Delta_current*I(i,:);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current-Delta_current*I(i,:);
        fvminus=feval(f,x_trial); k=k+1;
```

```

        if fvminus<fmin
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end

sol_compass=x_current;
Accuracy(1,1)=norm(sol_compass-x_min);
Iterations(1,1)=n-1;
Delta(1,1)=Delta_current;
Function_Value(1,1)=fmin;
Calls(1,1)=k;

% Enhanced Compass Search
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));

```



```

        fvminus=feval(f,x_trial); k=k+1;
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
end

if check==fmin
    Delta_current=0.5*Delta_current;
    nu=nu+1;
else
    x_current=x_next;
end

n=n+1;

end

sol_encomp=x_current;
Accuracy(2,1)=norm(sol_encomp-x_min);
Iterations(2,1)=n-1;
Delta(2,1)=Delta_current;
Function_Value(2,1)=fmin;
Calls(2,1)=k;

%Box Search
A=permn([1 -1],N);
[r,c]=size(A);
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;

    for i=1:r
        x_trial=x_current+Delta_current*A(i,:);
        fv=feval(f,x_trial); k=k+1;
        if fv<fmin
            x_next=x_trial;
            fmin=fv;
            continue
        end
    end
end

```

```

        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end

sol_box=x_current;
Accuracy(3,1)=norm(sol_box-x_min);
Iterations(3,1)=n-1;
Delta(3,1)=Delta_current;
Function_Value(3,1)=fmin;
Calls(3,1)=k;

% Hooke-Jeeves Search
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial); k=k+1;
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
        end
    end
end

```

```

        continue
    end
end

if check==fmin
    Delta_current=0.5*Delta_current;
    nu=nu+1;
else
    x_p=2*x_next-x_current;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_p+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_p+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial); k=k+1;
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end

    x_current=x_next;
end

n=n+1;

end

sol_hoje=x_current;
Accuracy(4,1)=norm(sol_hoje-x_min);
Iterations(4,1)=n-1;
Delta(4,1)=Delta_current;
Function_Value(4,1)=fmin;
Calls(4,1)=k;

% Presentation of results

display(' ')
fprintf('DIMENSION N=%u\n',N)

```

```
fprintf('Initial distance: %f\n',norm(x_0-x_min))
R={'Compass';'Enhanced Compass';'Box';'Hooke-Jeeves'};
table(Accuracy,Iterations,Delta,Function_Value,Calls,'RowNames',R)
```

### A.2. To script all\_methods\_iter

```
f=@rosenbrock;
x_min=[1,1];
r=10;
Delta_0=0.1;
Delta_tol=0.00001;
N=length(x_min);
I=eye(N);
step=2000;

Iterations=zeros(20,4);
Function_Value=zeros(20,4);

for m=1:20
    nmax=m*step;

    v=rand(1,N);
    x_0=x_min+(r/norm(v))*v;

    % Compass Best Search
    x_current=x_0;
    Delta_current=Delta_0;
    fmin=feval(f,x_current); k=1;
    n=1;
    nu=0;

    while n<=nmax && Delta_current>Delta_tol

        check=fmin;

        for i=1:N
            x_trial=x_current+Delta_current*I(i,:);
            fvplus=feval(f,x_trial); k=k+1;
            if fvplus<fmin
                x_next=x_trial;
                fmin=fvplus;
                continue
            end
            x_trial=x_current-Delta_current*I(i,:);
            fvminus=feval(f,x_trial); k=k+1;
            if fvminus<fmin
                x_next=x_trial;
                fmin=fvminus;
                continue
            end
        end
    end
end
```

```

        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end

sol_compass=x_current;
Iterations(m,1)=n-1;
Function_Value(m,1)=fmin;

% Enhanced Compass Search
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial); k=k+1;
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
end
end

```

```
    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;

end

sol_encomp=x_current;
Iterations(m,2)=n-1;
Function_Value(m,2)=fmin;

%Box Search
A=permn([1 -1],N);
[r,c]=size(A);
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;

    for i=1:r
        x_trial=x_current+Delta_current*A(i,:);
        fv=feval(f,x_trial); k=k+1;
        if fv<fmin
            x_next=x_trial;
            fmin=fv;
            continue
        end
    end

    if check==fmin
        Delta_current=0.5*Delta_current;
        nu=nu+1;
    else
        x_current=x_next;
    end

    n=n+1;
```

```

end

sol_box=x_current;
Iterations(m,3)=n-1;
Function_Value(m,3)=fmin;

% Hooke-Jeeves Search
x_current=x_0;
Delta_current=Delta_0;
fmin=feval(f,x_current); k=1;
n=1;
nu=0;

while n<=nmax && Delta_current>Delta_tol

    check=fmin;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_current+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin
            h=h+I(i,:);
            x_next=x_trial;
            fmin=fvplus;
            continue
        end
        x_trial=x_current+Delta_current*(h-I(i,:));
        fvminus=feval(f,x_trial); k=k+1;
        if fvminus<fmin
            h=h-I(i,:);
            x_next=x_trial;
            fmin=fvminus;
            continue
        end
    end
end

if check==fmin
    Delta_current=0.5*Delta_current;
    nu=nu+1;
else
    x_p=2*x_next-x_current;
    h=zeros(1,N);

    for i=1:N
        x_trial=x_p+Delta_current*(I(i,:)+h);
        fvplus=feval(f,x_trial); k=k+1;
        if fvplus<fmin

```

```

        h=h+I(i,:);
        x_next=x_trial;
        fmin=fvplus;
        continue
    end
    x_trial=x_p+Delta_current*(h-I(i,:));
    fvminus=feval(f,x_trial); k=k+1;
    if fvminus<fmin
        h=h-I(i,:);
        x_next=x_trial;
        fmin=fvminus;
        continue
    end
end
end

    x_current=x_next;
end

n=n+1;

end

sol_hoje=x_current;
Iterations(m,4)=n-1;
Function_Value(m,4)=fmin;

end

% Presentation of results

Max_Allowed=step*(1:20)';
Compass=Iterations(:,1);
Enh_Compass=Iterations(:,2);
Box=Iterations(:,3);
Hooke_Jeeves=Iterations(:,4);
display(' ')
display('Iterations needed for convergence')
table(Max_Allowed,Compass,Enh_Compass,Box,Hooke_Jeeves)

Compass=Function_Value(:,1);
Enh_Compass=Function_Value(:,2);
Box=Function_Value(:,3);
Hooke_Jeeves=Function_Value(:,4);
display(' ')
display('Corresponding function value')
table(Max_Allowed,Compass,Enh_Compass,Box,Hooke_Jeeves)

```



**A.3. To script all\_methods\_radius**

```
f=@rosenbrock;
x_min=ones(1,20);
Delta_0=0.1;
Delta_tol=0.00001;
N=length(x_min);
I=eye(N);
nmax=1000;

Iterations=zeros(16,4);
Function_Value=zeros(16,4);

for m=1:16
    r=10/2^(m-1);

--- after that point is the same as all_methods_iter ---
```



## Βιβλιογραφία

- [1] L. ARMJO, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal of Mathematics **16**, 1 – 3, 1966.
- [2] G. E. P. BOX, *Evolutionary operation: a method for increasing industrial productivity*, Applied Statistics **6**, 81 – 101, 1957.
- [3] C. DAVIS, *Theory of positive linear dependence*, American Journal of Mathematics **76**, 733 – 746, 1954.
- [4] A. A. GOLDSTEIN, *Constructive Real Analysis*, Harper & Row, New York, 1967.
- [5] I. GRIVA, S. NASH AND A. SOFER, *Linear and Nonlinear Programming*, 2nd Edition, SIAM, Philadelphia, 2009.
- [6] R. HOOKE AND T. A. JEEVES, *Direct search solution of numerical and statistical problems*, Journal of the ACM **8**, 212 – 229, 1961.
- [7] R. M. LEWIS, V. TORZCON AND M. W. TROSSET, *Direct search methods: Then and now*, Journal of Computational and Applied Mathematics **124**, 191 – 207, 2000.
- [8] J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM, *Testing unconstrained optimization software*, ACM Transactions for Mathematical Software **7**, 17 – 41, 1981.
- [9] T. G. KOLDA, R. M. LEWIS AND V. TORZCON, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Review **45**, 385 – 482, 2003.
- [10] V. TOZCON, *On the convergence of pattern search algorithms*, SIAM Journal of Optimization **7**, 1 – 25, 1997.
- [11] P. WOLFE, *Convergence conditions for ascent methods*, SIAM Review **11**, 226–235, 1969.