

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



Μεταπτυχιακό Μάθημα  
ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

Άσκηση 1

Ομάδα

Τεριζή Χρυσούλα, 430  
Ντούσια Μαργαρίτα, 248  
Παπασιόμπας Μιχάλης, 447

Νοέμβριος 2020

# 1 Εισαγωγή

Στόχος της εργασίας είναι η κατασκευή ενός πολυωνυμικού μοντέλου το οποίο μπορεί να προσεγγίσει με τη μέγιστη δυνατή ακρίβεια τον αριθμό των συνολικών χρουσμάτων κατά τη διάρκεια μίας πανδημίας. Το πολυωνυμικό μοντέλο 4ου βαθμού το οποίο προσεγγίζει το πλήθος των χρουσμάτων είναι το εξής:

$$p(\alpha, x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \alpha_4 x^4$$

όπου  $\alpha = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$  είναι το άγνωστο διάνυσμα των συντελεστών του μοντέλου και το οποίο περιγράφει τα δεδομένα με τον καλύτερο δυνατό τρόπο. Αυτό σημαίνει ότι ελαχιστοποιεί τη συνάρτηση τετραγωνικού σφάλματος:

$$f(\alpha) = \sum_{i=1}^M [p(\alpha, x_i) - y_i]^2$$

όπου  $M = 30$  το πλήθος των διαθέσιμων μετρήσεων στα δεδομένα. Για την επίλυση και εύρεση της βέλτιστης λύσης της συνάρτησης  $f(\alpha)$  εφαρμόζουμε τέσσερις μεθόδους βελτιστοποίησης με παραγώγους, με στόχο την εύρεση λύσης που ικανοποιεί το παρακάτω κριτήριο:

$$\|\nabla f(\alpha)\| \leq 10^{-6}$$

Η αναφορά οργανώνεται ως εξής, Ενότητα 2 Μέθοδοι, μέσα στην οποία γίνεται μια σύντομη περιγραφή της λειτουργίας των τεσσάρων μεθόδων βελτιστοποίησης και Ενότητα 3 Αποτελέσματα, μέσα στην οποία παρουσιάζονται και αναλύονται τα αποτελέσματα των πειραμάτων. Στο τέλος της αναφοράς παραθέτουμε το Παράρτημα Α Εκτελέσιμα, στο οποίο περιγράφεται σύντομα η λειτουργία του κώδικα και τα εκτελέσιμα αρχεία.

## 2 Μέθοδοι

Οι μέθοδοι βελτιστοποίησης με παραγώγους οι οποίες εφαρμόστηκαν για την εύρεση του ελαχιστοποιητή  $\alpha$  της συνάρτησης σφάλματος  $f(\alpha)$  είναι οι ακόλουθες:

1. Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe
2. Μέθοδος Newton ασφαλούς περιοχής
3. Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe
4. Μέθοδος Steepest Descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe

Οι τρέχουσες μέθοδοι βελτιστοποίησης με παραγώγους χωρίζονται σε μεθόδους ευθύγραμμης αναζήτησης και ασφαλούς περιοχής. Το βασικό επαναληπτικό σχήμα των μεθόδων Ευθύγραμμης Αναζήτησης είναι ότι πρώτα γίνεται η επιλογή διεύθυνσης αναζήτησης και έπειτα ο υπολογισμός του βέλτιστου βήματος, ενώ στις μεθόδους Ασφαλούς Περιοχής πρώτα γίνεται η επιλογή της ασφαλούς περιοχής και έπειτα ο υπολογισμός της διεύθυνσης αναζήτησης εντός της ασφαλούς περιοχής. Ακολουθεί μια σύντομη περιγραφή αυτών των δυο στρατηγικών βελτιστοποίησης.

### Μέθοδοι Ευθύγραμμης Αναζήτησης με συνθήκες Wolfe

Οι μέθοδοι ευθύγραμμης αναζήτησης εξασφαλίζουν την επιλογή κατάλληλων βημάτων μέσα από την προσέγγιση της αντικειμενικής συνάρτησης  $f(\alpha)$  με ένα τετραγωνικό μοντέλο  $m_k(p)$  έτσι ώστε α) να μειώνεται σημαντικά η τιμή της αντικειμενικής συνάρτησης αλλά και β) να επιφέρει ταχεία σύγκλιση. Η πρώτη απαίτηση εκφράζεται μέσω της συνθήκης [Armijo](#) ενώ η δεύτερη μέσω της συνθήκης [καμπυλότητας](#).

1. [Συνθήκη Armijo](#) για την οποία το βήμα που επιλέχθηκε  $a_k > 0$  θα πρέπει να ικανοποιεί την σχέση:

$$f(x_k + a_k p_k) \leq f(x_k) + c_1 a_k \nabla f_k^T p_k \quad \text{όπου } c_1 \in (0, 1)$$

Η μείωση της αντικειμενικής συνάρτησης  $f(\alpha)$  είναι ανάλογη τόσο με το βήμα  $a_k$  όσο και με την παράγωγο κατά κατεύθυνση  $\nabla f_k^T p_k$ . Αν και η [Armijo](#) αποτελεί μια συνθήκη ικανοποιητικής μείωσης, δεν είναι από μόνη της αρκετή για να μας εξασφαλίσει ότι ο αλγόριθμός μας θα σημειώνει σημαντική πρόοδο κατά την αναζήτηση του ελαχίστου βήματος, καθότι ικανοποιείται για πολλές μικρές τιμές του  $a_k$ .

2. [Συνθήκη καμπυλότητας](#) για την οποία το βήμα που επιλέχθηκε  $a_k > 0$  θα πρέπει να ικανοποιεί την παρακάτω σχέση:

$$\nabla f(x_k + a_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k \quad \text{όπου } c_2 \in (c_1, 1)$$

Με την εισαγωγή και της συνθήκης καμπυλότητας εξασφαλίζουμε ότι θα αποκλειστούν από τον αλγόριθμό μας οι μη αποδεκτές τιμές του βήματος, επιτυγχάνοντας με αυτόν τον τρόπο ταχύτερη σύγκλιση.

### 2.1 Μέθοδος Newton με ευθύγραμμη αναζήτηση με συνθήκες Wolfe

Η μέθοδος Newton στοχεύει στην ελαχιστοποίηση της αντικειμενικής συνάρτησης  $f(\alpha)$  το οποίο επιτυγχάνει με τη δημιουργία μιας ακολουθίας από σημεία  $x_k$ , ξεκινώντας από ένα αρχικό σημείο  $x_0$  το οποίο συγκλίνει προς τον ελαχιστοποιητή της  $f(\alpha)$ . Όσο το μέτρο της gradient στο σημείο  $x_k$  είναι μεγαλύτερο από τη ζητούμενη ακρίβεια  $\epsilon=10^{-6}$ , υπολογίζουμε μια νέα διεύθυνση μείωσης  $p_k = -(\nabla^2 f_k)^{-1} \nabla f_k$ , και ενημερώνουμε τη τιμή του  $x_k$  ως εξής:  $x_k = x_k + a_k * p_k$ , όπου  $a_k$  το βέλτιστο βήμα της τρέχουσας επανάληψης, που προκύπτει από την κλήση της συνάρτησης [lineSearch](#) (Παράρτημα Α).

2.2 Μέθοδος BFGS με ευθύγραμμη αναζήτηση με συνθήκες Wolfe

Για τη μέθοδο αυτή χρησιμοποιούμε ένα αρχικό σημείο  $x_0$ , και ένα συμμετρικό και θετικά ορισμένο πίνακα  $H_0$  ο οποίος αποτελεί μια αρχική προσέγγιση του Εσσιανού πίνακα. Στην περίπτωση όπου είναι διαθέσιμος ο Εσσιανός πίνακας τότε μόνο για την πρώτη επανάληψη θέτουμε  $H_0$  ίσο με τον Εσσιανό, διαφορετικά θέτουμε το  $H_0$  να είναι ένα πολλαπλάσιο του μοναδιαίου πίνακα  $I$ . Όσο το μέτρο της gradient στο σημείο  $x_k$  είναι μεγαλύτερο από τη ζητούμενη ακρίβεια  $\epsilon=10^{-6}$ , η νέα διεύθυνση αναζήτησης  $p_k$  υπολογίζεται από το γινόμενο του  $-H_k \nabla f_k$ . Στην συνέχεια, η lineSearch υπολογίζει το βέλτιστο βήμα  $a_k$ , το οποίο οδηγεί στο καινούργιο σημείο  $x_{k+1} = x_k + a_k * p_k$ . Τέλος, υπολογίζονται οι μεταβλητές  $s_k = x_{k+1} - x_k$  και  $y_k = \nabla f_{k+1} - \nabla f_k$ , οι οποίες σε συνδυασμό και με τον θετικά ορισμένο πίνακα  $H_k$  δημιουργούν το νέο  $H_{k+1}$ . Περισσότερες λεπτομέρειες για τον τρόπο εκτέλεσης της μεθόδου παρατίθεται στο Παράρτημα Α.

2.3 Μέθοδος Steepest Descent με ευθύγραμμη αναζήτηση με συνθήκες Wolfe

Η μέθοδος steepest descent υπολογίζει την κατεύθυνση της πιο απότομης καθόδου της  $f(\alpha)$  στο σημείο  $x_k$  και χρησιμοποιεί για διεύθυνση μείωσης το διάνυσμα  $p_k = -\nabla f_k$ . Ο αλγόριθμος ξεκινάει από ένα αρχικό σημείο  $x_0$ , και όσο το μέτρο της gradient στο σημείο  $x_k$  είναι μεγαλύτερο από τη ζητούμενη ακρίβεια  $\epsilon = 10^{-6}$ , υπολογίζουμε μια νέα διεύθυνση μείωσης  $p_k$ , και ενημερώνουμε τη τιμή του  $x_k$  ως εξής:  $x_k = x_k + a_k * p_k$ , όπου  $a_k$  το βέλτιστο βήμα της τρέχουσας επανάληψης, που προκύπτει από την κλήση της συνάρτησης lineSearch (Παράρτημα Α). Η νέα παραγόμενη διεύθυνση αναζήτησης της Steepest Descent ελαχιστοποιεί τη συνάρτηση  $f(\alpha)$  μονοτονικά ή όχι.

2.4 Μέθοδος Newton ασφαλούς περιοχής

Οι μέθοδοι ασφαλούς περιοχής προσεγγίζουν την αντικειμενική συνάρτηση  $f(\alpha)$  με τη βοήθεια ενός τετραγωνικού μοντέλου  $m_k$  μέσω μιας ακολουθίας βημάτων  $x_k$ . Επίσης, ορίζουν μια περιοχή γύρω από την τρέχουσα βέλτιστη λύση μέσα στην οποία το τετραγωνικό μοντέλο προσεγγίζει την αντικειμενική συνάρτηση  $f(\alpha)$ . Αυτές οι μέθοδοι ορίζουν ένα μέγεθος του βήματος πριν τη βελτίωση της κατεύθυνσης και αν ύστερα απο την ανανέωση του βήματος επέλθει μείωση της τιμής της αντικειμενικής συνάρτησης, τότε θεωρούμε ότι το τετραγωνικό μοντέλο που ορίσαμε αποτελεί μια καλή αναπαράσταση της αρχικής αντικειμενικής συνάρτησης. Σε αντίθετη περίπτωση, το τετραγωνικό μοντέλο δεν αποτελεί καλή αναπαράσταση εντός της συγκεκριμένης περιοχής.

Επομένως, για τη μέθοδο αυτή χρησιμοποιούμε ένα τετραγωνικό μοντέλο  $m_k$  εντός μιας ασφαλούς περιοχής με ακτίνα  $\Delta_k > 0$  γύρω από το σημείο  $x_k$  και ως διεύθυνση αναζήτησης  $p_k$  λαμβάνεται το ολικό ελάχιστο του  $m_k$  εντός της ασφαλούς περιοχής, δηλαδή  $\|p_k\| < \Delta_k$ . Περισσότερες λεπτομέρειες για τον τρόπο εκτέλεσης της μεθόδου παρατίθεται στο Παράρτημα Α.

3 Αποτελέσματα

Στην τρέχουσα ενότητα παραθέτουμε τα αποτελέσματα των πειραμάτων για την κάθε μια μέθοδο βελτιστοποίησης ξεχωριστά για κάθε ένα από τα πέντε διαφορετικά σημεία εκκίνησης  $x_1, x_2, x_3, x_4, x_5$  όπου είναι τα ακόλουθα:

$$x_1 = [ 5.025, -4.898, 0.119 \ 3.981 \ 7.818 ]$$
$$x_2 = [ 9.185, 0.944, -7.227, -7.014, -4.849 ]$$
$$x_3 = [ 6.814, -4.914, 6.285, -5.129, 8.585 ]$$
$$x_4 = [ -2.966, 6.616, 1.705, 0.994, 8.343 ]$$
$$x_5 = [ -7.401, 1.376, -0.612, -9.762, -3.257 ]$$

Αρχικά, στον Πίνακα 1 παραθέτουμε τις αρικοποιήσεις κάποιων βασικών παραμέτρων, όπως είναι το μέγιστο βήμα (a\_max), οι τιμές  $c_1$  και  $c_2$  για τις συνθήκες Wolfe για τις μεθόδους με ευθύγραμμη αναζήτηση, τη μέγιστη ακτίνα (radius\_max) και το κατώφλι αποδοχής της λύσης (eta) για την μέθοδο ασφαλούς περιοχής.

Μέθοδος	a_max	c1	c2	radius_max	eta
Newton Ευθύγραμμης Αναζήτησης	2	$10^{-4}$	0.9	-	-
BFGS Ευθύγραμμης Αναζήτησης	2	$10^{-4}$	0.9	-	-
Newton Ασφαλούς Περιοχής	-	-	-	1	0.2
Steepest Descent Ευθύγραμμης Αναζήτησης	1	$10^{-4}$	0.9	-	-

Table 1: Παραμετροποίηση αλγορίθμων

Παρατηρούμε ότι η καλύτερη μέθοδος βελτιστοποίησης η οποία εντοπίζει πιο γρήγορα τον ελαχιστοποιητή  $\alpha$  της αντικειμενικής συνάρτησης  $f(\alpha)$  είναι η μέθοδος Newton με ευθύγραμμη αναζήτηση. Η σύγκλιση του αλγορίθμου ολοκληρώνεται σε μόλις 2 επαναλήψεις για τα σημεία  $x_1, \dots, x_3$  και σε 1 επανάληψη για τα σημεία  $x_4$  και  $x_5$  (Πίνακας 2). Τέλος, ο ελαχιστοποιητής  $\alpha$  της αντικειμενικής συνάρτησης  $f(\alpha)$  που προέκυψε είναι ο ακόλουθος:

$$\alpha = [2.19714047, \ 0.88569895, \ -0.80419248, \ 0.60753577, \ -0.09500749]^\top$$

Η δεύτερη καλύτερη μέθοδος βελτιστοποίησης, η οποία βρίσκει και εκείνη με επιτυχία τον ελαχιστοποιητή  $\alpha$  της αντικειμενικής συνάρτησης  $f(\alpha)$  είναι η μέθοδος BFGS ευθύγραμμης αναζήτησης. Η σύγκλιση του

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	Χρόνος εκτέλεσης (sec)
Newton Ευθύγραμμης Αναζήτησης	<b>2</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>1</b>	0.03
BFGS Ευθύγραμμης Αναζήτησης	18	16	20	15	12	0.26
Newton Ασφαλούς Περιοχής	17	20	22	17	21	0.21
Steepest Descent Ευθύγραμμης Αναζήτησης	1000	1000	1000	1000	1000	18.89

Table 2: Σύγκλιση (πλήθος επαναλήψεων) και χρόνος εκτέλεσης κάθε μεθόδου βελτιστοποίησης για τα αρχικά σημεία  $x_1, \dots, x_5$ . Στην τελευταία στήλη αναφέρεται ο μέσος απαιτούμενος χρόνος εκτέλεσης κάθε μεθόδου

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
Newton	1	1	1	1	1
Cauchy	9	9	9	8	7
Dogleg	7	10	12	8	13
Συρρίκνωση ακτίνας	0	0	0	0	0
Αύξηση ακτίνας	8	8	6	7	5

Table 3: Συνολικά επιλεχθέντα σημεία **Newton**, **Cauchy** και **Dogleg** και συνολικό πλήθος συρρίκνωσης και αύξησης ακτίνας ασφαλούς περιοχής για την μέθοδο Newton Ασφαλούς Περιοχής για κάθε ένα αρχικό σημείο  $x_1, \dots, x_5$

αλγορίθμου είναι επίσης αρκετά γρήγορη καθώς μέσα σε 12 με 20 επαναλήψεις ολοκληρώνεται για οποιοδήποτε από τα σημεία εκκίνησης (**Πίνακας 2**).

Στην συνέχεια, η **τρίτη** καλύτερη μέθοδος βελτιστοποίησης είναι η Newton Ασφαλούς Περιοχής. Η μέθοδος συγκλίνει με επιτυχία στο ελάχιστο  $\alpha$  της συνάρτησης σφάλματος  $f(\alpha)$  μέσα σε 17 με 22 επαναλήψεις ανάλογα το σημείο εκκίνησης (**Πίνακας 2**). Μία σύντομη αναφορά του συνολικού πλήθους των επιλεγμένων σημείων, **Newton**, **Cauchy** και **Dogleg** γίνεται στον πάνω μέρος του **Πίνακα 3**. Παρατηρούμε ότι, για κάθε αρχικό σημείο το πλήθος των **Newton** σημείων είναι 1, ανεξάρτητα από την αρχικοποίηση της μέγιστης ακτίνας ασφαλούς περιοχής, το πλήθος των **Cauchy** σημείων κυμαίνεται από 7 με 9 σημεία και τέλος, οι περισσότερες επιλογές σημείων **Dogleg** είναι μεταξύ 7 με 13. Επιπλέον, το συνολικό πλήθος συρρικνώσεων και αυξήσεων της ακτίνας ασφαλούς περιοχής αναφέρονται στο κάτω μέρος του **Πίνακα 3**. Παρατηρούμε ότι συρρίκνωση ακτίνας δεν έχει πραγματοποιηθεί για κανένα από τα 5 διαφορετικά αρχικά σημεία  $x_1, \dots, x_5$ , ενώ η αύξηση ακτίνας ασφαλούς περιοχής επιλέγεται για κάθε σημείο  $x_1, \dots, x_5$ .

Τέλος, η μέθοδος βελτιστοποίησης η οποία **δεν προσεγγίζει καλά** τον ελαχιστοποιητή  $\alpha$  της αντικειμενικής συνάρτησης  $f(\alpha)$  είναι η **Steepest Descent**, η οποία με το πέρας των 1000 επαναλήψεων δεν έχει βρει τον ελαχιστοποιητή για κανένα από τα διαφορετικά σημεία εκκίνησης. Συγκεκριμένα, η μέθοδος έπειτα από το τις 1000 επαναλήψεις επέστρεψε τα ακόλουθα σημεία  $\alpha_1, \dots, \alpha_5$  για κάθε αρχικό σημείο  $x_1, \dots, x_5$  αντίστοιχα,

$$\begin{aligned} \alpha_1 &= [5.100038, -4.498962, 0.066485, 1.961737, -0.490557]^\top \\ \alpha_2 &= [6.702096, 6.702096, -3.227991, -0.061775, 0.337126]^\top \\ \alpha_3 &= [5.469935, -4.708639, 7.385062, -4.713762, 0.965106]^\top \\ \alpha_4 &= [-2.064716, 6.318050, 0.473381, -2.060834, 0.514610]^\top \\ \alpha_5 &= [-3.137680, 5.080902, 4.031951, -3.945349, 0.804124]^\top \end{aligned}$$

### Βέλτιστο μοντέλο

Όπως έχουμε ήδη αναφέρει παραπάνω, τρεις από τις τέσσερις μεθόδους βελτιστοποίησης με παραγώγους (Newton Ευθύγραμμης Αναζήτησης, Newton Ασφαλούς Περιοχής και BFGS Ευθύγραμμης Αναζήτησης), βρίσκουν με επιτυχία τον ελαχιστοποιητή  $\alpha$  της συνάρτησης σφάλματος  $f(\alpha)$ . Στην **Εικόνα 1** απεικονίζονται οι προσεγγίσεις του βέλτιστου μοντέλου σε σύγκριση με τα πραγματικά δεδομένα. Παρατηρούμε ότι το πολυώνυμο  $p(\alpha, x)$  προσεγγίζει με **μεγάλη** ακρίβεια τις πραγματικές τιμές των κρουσμάτων από 14 Οκτωβρίου 2020 (ημέρα 1) έως και 12 Νοεμβρίου 2020 (ημέρα 30). Όμως, από 13 έως 17 Νοεμβρίου 2020 (ημέρες 31 έως 35) το πολυώνυμο προβλέπει λιγότερα συνολικά κρούσματα σε σχέση με τα πραγματικά.

Όσον αφορά την μέθοδο βελτιστοποίησης Steepest Descent, η οποία με το πέρας των 1000 επαναλήψεων αδυνατεί να εντοπίσει τον ολικό ελαχιστοποιητή της συνάρτησης σφάλματος  $f(\alpha)$ , παρατηρούμε ότι οι προσεγγίσεις αποκλίνουν αρκετά από τις πραγματικές τιμές (**Εικόνα 2**). Αναμέναμε αυτήν την συμπεριφορά της μεθόδου να εμφανιστεί, διότι η συγκεκριμένη μέθοδος έχει την τάση να χρειάζεται αρκετές επαναλήψεις έτσι ώστε να βρει την βέλτιστη λύση.

### Σφάλμα αντικειμενικής συνάρτησης

Ένα σημαντικό σημείο είναι η αξιολόγηση της ποιότητας των λύσεων που παίρνουμε από την κάθε μια μέθοδο και οι οποίες καθορίζονται από τις παρεχόμενες τιμές σφάλματος. Στόχος των μεθόδων είναι η εύρεση του επόμενου βέλτιστου βήματος, που οδηγεί στην ελαχιστοποίηση της αντικειμενικής συνάρτησης  $f(\alpha)$  έως ότου επιτευχθεί η απαιτούμενη ακρίβεια  $\epsilon$  για το μέτρο της gradient. Η μείωση του σφάλματος μπορεί να είναι είτε μονοτονική είτε όχι.

Η **Εικόνα 3** απεικονίζει την εξέλιξη του σφάλματος σε κάθε επανάληψη για κάθε μέθοδο και για κάθε αρχικό σημείο  $x_1, \dots, x_5$ . Αρχικά, για την μέθοδο Newton με ευθύγραμμη αναζήτηση (Εικόνα 3 (a)) παρατηρούμε

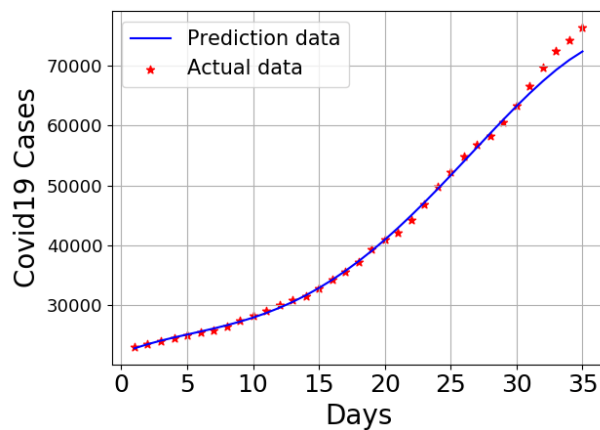


Figure 1: Προσέγγιση βέλτιστου μοντέλου στα πραγματικά δεδομένα (ημέρες:1-30) και πρόβλεψη για το επόμενο διάστημα 5 ημερών

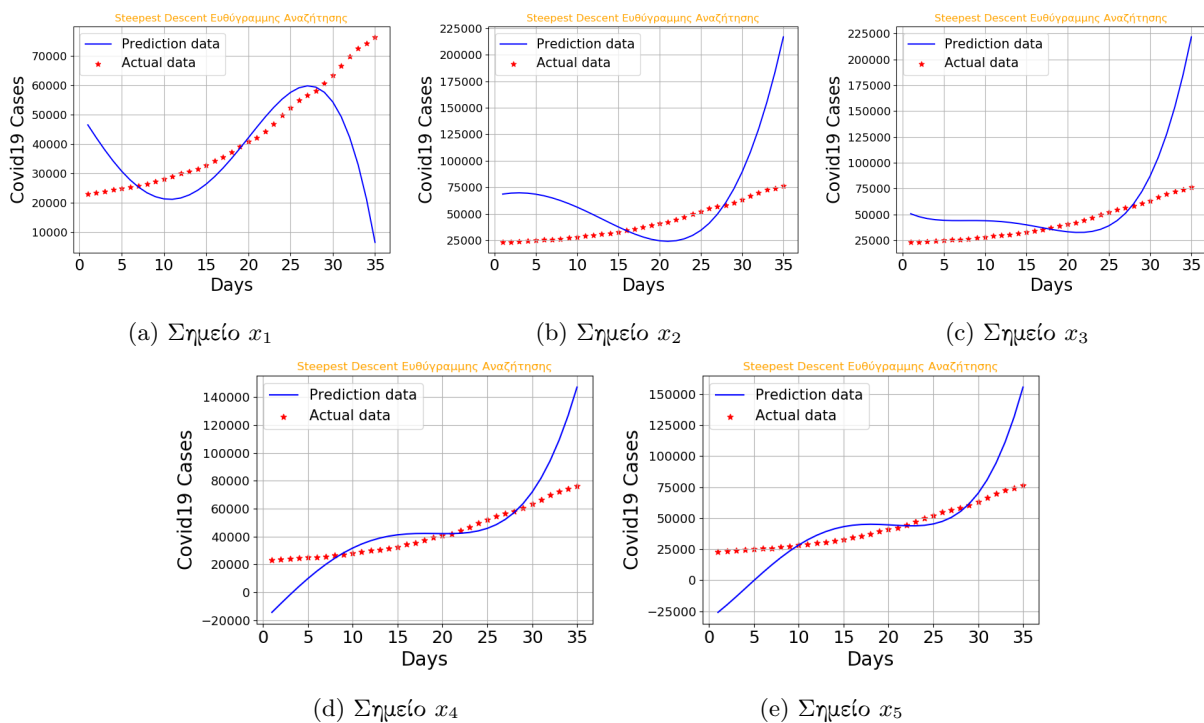


Figure 2: Προσέγγιση μη βέλτιστου μοντέλου για την μέθοδο Steepest Descent στα πραγματικά δεδομένα (ημέρες:1-30) και πρόβλεψη για το επόμενο διάστημα 5 ημερών για κάθε σημείο εκκίνησης

ότι επέρχεται σύγκλιση με μεγάλη ταχύτητα, και αυτό ωφείλεται στο γεγονός ότι το αρχικό σημείο είναι κοντά στη ρίζα.

Για την μέθοδο BFGS με ευθύγραμμη αναζήτηση (Εικόνα 3 (b)) παρατηρούμε ότι η αντικειμενική συνάρτηση μειώνεται απότομα στις πρώτες δύο επαναλήψεις για κάθε αρχικό σημείο  $x_1, \dots, x_5$ . Έπειτα για τις επόμενες 2 επαναλήψεις υπάρχει μια μικρή μείωση του σφάλματος, και απο εκείνο το σημείο και ύστερα υπάρχει μονοτονική μείωση του σφάλματος.

Για την μέθοδο Newton Ασφαλούς Περιοχής (Εικόνα 3 (c)) παρατηρούμε ότι η τιμή της αντικειμενικής συνάρτησης για την μέθοδο και για κάθε αρχικό σημείο  $x_1, \dots, x_5$  μειώνεται μονοτονικά, το οποίο μας εγγυάται τη σύγκλιση της μεθόδου.

Για την μέθοδο Steepest Descent με ευθύγραμμη αναζήτηση (Εικόνα 3 (d)) παρατηρούμε ότι για κάθε αρχικό σημείο  $x_1, \dots, x_5$  η τιμή της αντικειμενικής συνάρτησης μειώνεται μονοτονικά στις πρώτες επαναλήψεις και έπειτα η μείωση γίνεται με πολύ μικρές τιμές καθώς πλησιάζει τον ελαχιστοποιητή.

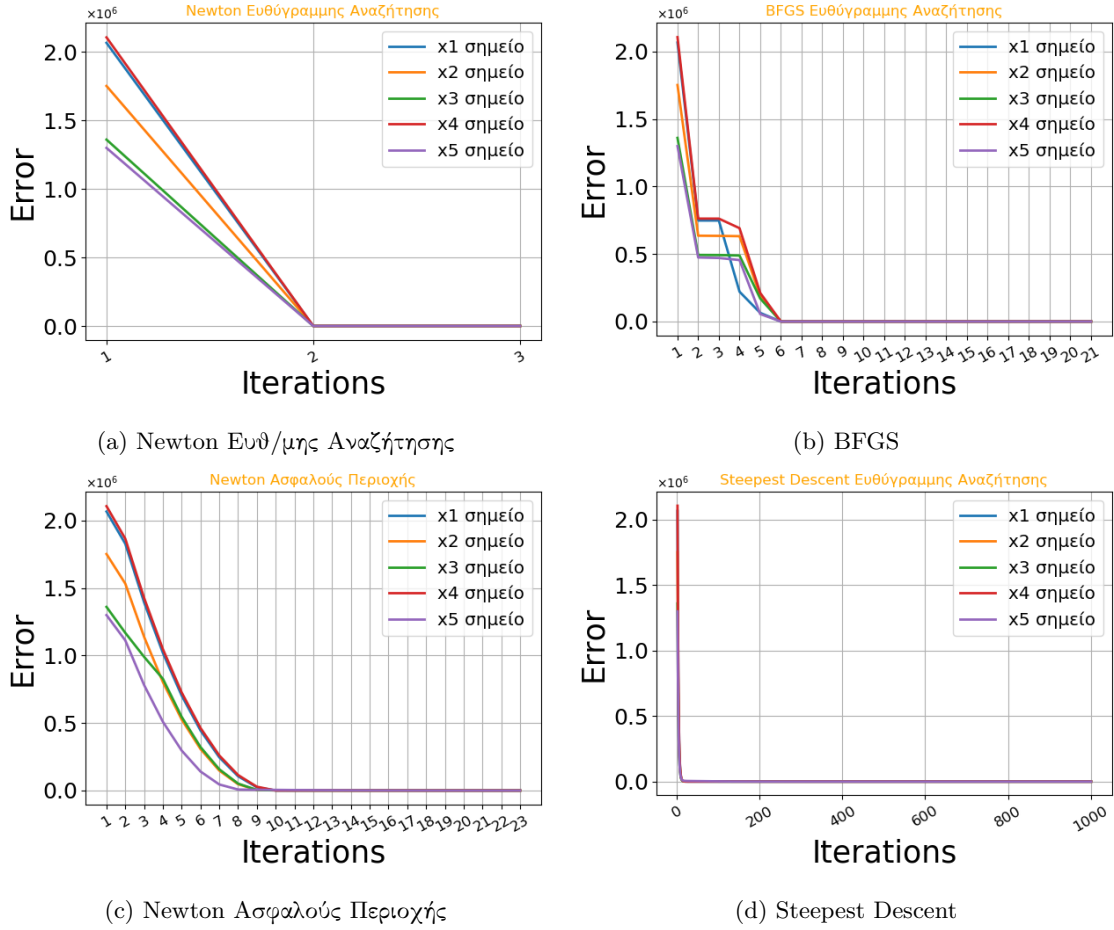


Figure 3: Απεικόνιση σφάλματος για τις μεθόδους (a) Newton Ευθύγραμμης Αναζήτησης, (b) BFGS Ευθύγραμμης Αναζήτησης, (c) Newton Ασφαλούς Περιοχής (d) Steepest Descent Ευθύγραμμης Αναζήτησης για κάθε σημείο εκκίνησης  $x_1, \dots, x_5$

## A Εκτελέσιμα

Στο τρέχων παράρτημα περιγράφεται η λειτουργία του κώδικα που αναπτύχθηκε για την τρέχουσα εργασία και ο τρόπος εκτέλεσης του εκτελέσιμου αρχείου. Ακολουθεί η περιγραφή των 10 επισυναπτόμενων αρχείων:

### 1. main.py

Είναι το κύριο εκτελέσιμο αρχείο. Γίνεται η φόρτωση των δεδομένων και των αρχικών σημείων. Δέχεται ως είσοδο από τον χρήστη το σημείο εκκίνησης και τη μέθοδο βελτιστοποίησης που επιθυμεί να εκτελέσει και κάνει τις αντίστοιχες κλήσεις στις αντίστοιχες μεθόδους. Με την ολοκλήρωση της μεθόδου φτιάχνει την γραφική παράσταση (*cases\_αρχικό-σημείο.png*) με τις πραγματικές τιμές κρουσμάτων και αυτές που προβλέπει το μοντέλο θέτοντας ως είσοδο στην συνάρτηση τον ελαχιστοποιητή που επιστρέφει η κάθε μέθοδος και τη γραφική παράσταση σφάλματος για κάθε επανάληψη (*error\_αρχικό-σημείο.png*).

### 2. load.py

Η συνάρτηση διαβάζει τα δεδομένα (ημέρα - συνολικά κρούσματα) από το αρχείο *covid\_data\_30-GR.dat* και τα επιστρέφει. Οι ημέρες είναι διαιρεμένες με το 10 και οι πραγματικές τιμές συνολικών κρουσμάτων με το 10000. Επίσης, διαβάζει τα αρχικά σημεία  $x_1, \dots, x_5$  της εκφώνησης από το αρχείο *initial\_points.dat* και τα επιστρέφει.

### 3. objectiveF.py

Η συνάρτηση υπολογίζει για ένα δωθέν σημείο  $\alpha$  την τιμή της συνάρτησης τετραγωνικού σφάλματος  $f(\alpha)$ .

### 4. gradient.py

Δέχεται ως είσοδο τις συντεταγμένες ενός σημείου  $a$  και υπολογίζει το διάνυσμα (gradient) με μερικές παραγώγους 1<sup>ης</sup> τάξης της συνάρτησης τετραγωνικού σφάλματος  $f(\alpha)$ , όπου:

$$\nabla f(\alpha) = 2 \begin{bmatrix} \sum_{i=1}^{M=30} (p(a, x_i) - y_i) \\ \sum_{i=1}^{M=30} (p(a, x_i) - y_i) x_i \\ \sum_{i=1}^{M=30} (p(a, x_i) - y_i) x_i^2 \\ \sum_{i=1}^{M=30} (p(a, x_i) - y_i) x_i^3 \\ \sum_{i=1}^{M=30} (p(a, x_i) - y_i) x_i^4 \end{bmatrix}$$

### 5. hessian.py

Υπολογίζεται ο Εσσιανός πίνακας της συνάρτησης τετραγωνικού σφάλματος  $f(\alpha)$ . Ο Εσσιανός για το τρέχων πρόβλημα δεν εξαρτάται από τις συντεταγμένες του αρχικού σημείου, είναι δηλαδή σταθερός. Αν ο Εσσιανός πίνακας είναι θετικά ορισμένος και συμμετρικός τότε εξασφαλίζουμε ότι οι διευθύνσεις που

επιλέγονται είναι συναρτήσεις μείωσης. Επομένως, ο παραγόμενος Εσσιανός πίνακας είναι ο ακόλουθος, όπου  $i \in 1, \dots, M$ :

$$\nabla^2 f(\alpha) = 2 \begin{bmatrix} M & \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \sum x_i^6 \\ \sum x_i^3 & \sum x_i^4 & \sum x_i^5 & \sum x_i^6 & \sum x_i^7 \\ \sum x_i^4 & \sum x_i^5 & \sum x_i^6 & \sum x_i^7 & \sum x_i^8 \end{bmatrix}$$

#### 6. lineSearch.py

Η συνάρτηση υλοποιεί τον αλγόριθμο ευθύγραμμης αναζήτησης, με στόχο την εύρεση του επόμενου βέλτιστου βήματος που ικανοποιεί τις συνθήκες [Armijo](#) και [καμπυλότητας](#), αντίστοιχα. Στην περίπτωση που δεν ικανοποιούνται οι συνθήκες Wolfe γίνεται κλήση της συνάρτησης *zoom.py* και επιστρέφεται το νέο βήμα.

#### 7. zoom.py

Η συνάρτηση υπολογίζει το βέλτιστο βήμα μέσα από ένα φραγμένο διάστημα. Η επιλογή του βήματος γίνεται με παρεμβολή μέσα στο φραγμένο διάστημα και συγκεκριμένα με την μέθοδο της διχοτόμησης.

#### 8. newton.py

Αρχικά γίνεται έλεγχος αν ο Εσσιανός πίνακας είναι θετικά ορισμένος (έλεγχος ότι όλες οι ιδιοτιμές του είναι θετικές) και συμμετρικός. Στην περίπτωση που δεν είναι Θ.Ο γίνεται η τροποποίηση του σε Θ.Ο. Έπειτα, αρχίζει η εκτέλεση του επαναληπτικού σχήματος. Η συνάρτηση επιστρέφει τον ελαχιστοποιητή που έχει βρει και τα σφάλματα της αντικειμενικής συνάρτησης για κάθε επανάληψη.

#### 9. bfgs.py

Αρχικά γίνεται έλεγχος αν ο Εσσιανός πίνακας είναι θετικά ορισμένος (έλεγχος ότι όλες οι ιδιοτιμές του είναι θετικές) και συμμετρικός. Στην περίπτωση που δεν είναι θετικά ορισμένος γίνεται η τροποποίησή του σε θετικά ορισμένο και έπειτα, αρχίζει η εκτέλεση του επαναληπτικού σχήματος. Για κάθε νέα προσέγγιση του Εσσιανού πίνακα γίνεται έλεγχος αν είναι θετικά ορισμένος και σε αντίθετη περίπτωση γίνεται τροποποίησή του. Η συνάρτηση επιστρέφει τον ελαχιστοποιητή που έχει βρει και τα σφάλματα της αντικειμενικής συνάρτησης για κάθε επανάληψη.

#### 10. trustRegion.py

Γίνεται υλοποίηση του Dogleg αλγορίθμου με σκοπό την εύρεση της βέλτιστης διεύθυνσης αναζήτησης. Αρχικά, αν το σημείο Newton βρίσκεται εντός της ασφαλούς περιοχής τότε επιλέγεται αυτό, στην συνέχεια αν το Cauchy σημείο είναι εκτός ασφαλούς περιοχής τότε επιλέγεται το σημείο πάνω στο σύνορο της ασφαλούς περιοχής διαφορετικά επιλέγεται το Dogleg σημείο. Έπειτα, γίνεται ενημέρωση της ακτίνας της ασφαλούς περιοχής ανάλογα με το αν η ποιότητα της λύσης είναι καλή. Η συνάρτηση επιστρέφει τον ελαχιστοποιητή που έχει βρει και τα σφάλματα της αντικειμενικής συνάρτησης για κάθε επανάληψη.

### Οδηγίες ορθής εκτέλεσης κώδικα

Η υλοποίηση των μεθόδων έγινε σε γλώσσα **Python 3**. Οι βιβλιοθήκες που έχουν χρησιμοποιηθεί είναι οι Scipy, Matplotlib και NumPy. Επίσης, τα modules που χρησιμοποιήθηκαν είναι τα ακόλουθα, random, time και sys. Η γενική μορφή της εντολής εκτέλεσης του προγράμματος είναι η εξής:

**python3 main.py point\_id method\_name**

όπου point\_id  $\in \{1, 2, 3, 4, 5\}$  και method\_name  $\in \{newton, steepestDescent, trustRegion, bfgs\}$ . Παρακάτω παρατίθεται ένα ενδεικτικό παράδειγμα εκτέλεσης για το αρχικό σημείο  $x_3$  και την μέθοδο Newton Ασφαλούς Περιοχής:

**python3 main.py 3 trustRegion**