Τεχνικές Βελτιστοποίησης
Εργασία 1η

Παπουτσή Νικολέτα   ΑΕΜ : 10858

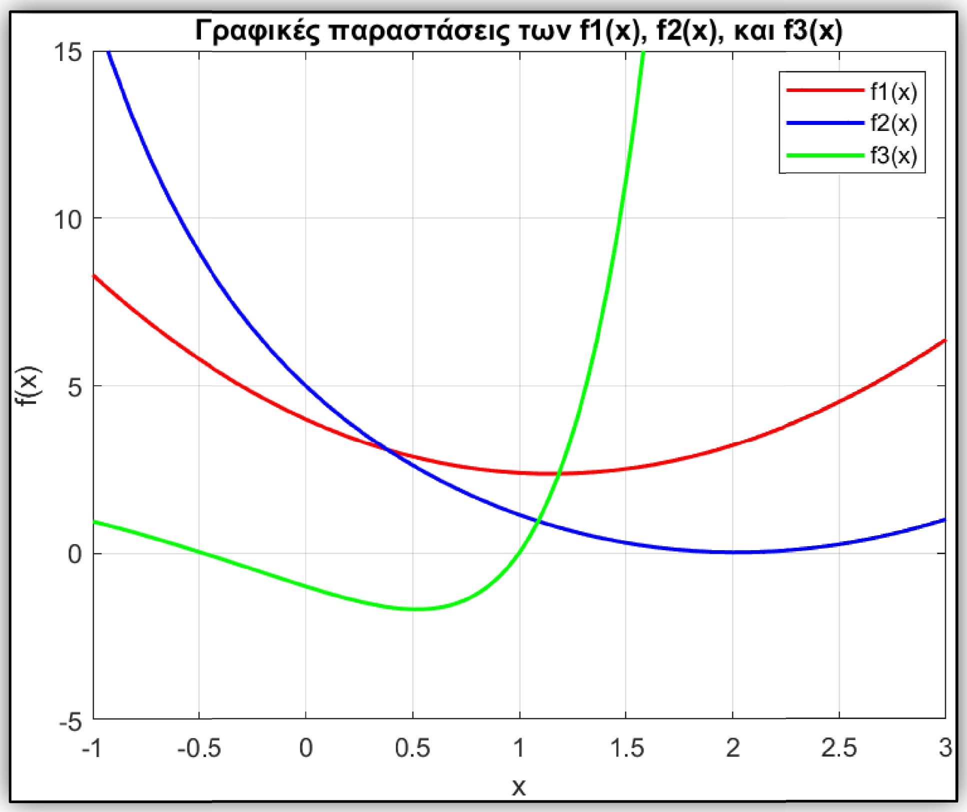
<u>Ζητούμενο</u> : Δίνονται τρεις κυρτές συναρτήσεις και αρχικό διάστημα **[a ,b]** , καλούμαστε να τις ελαχιστοποιήσουμε με τη χρήση των παρακάτω μεθόδων αναζήτησης:

1. **Μέθοδος της διχοτόμου**
2. **Μέθοδος του χρυσού τομέα**
3. **Μέθοδος Fibonacci**
4. **Μέθοδος της διχοτόμου με χρήση παραγώγων**

Το διάστημα αναζήτησης είναι το [-1, 3] και οι δοσμένες συναρτήσεις οι εξής :

- $\boldsymbol{f_1(x) = (x-2)^2 + \mathbf{x} \cdot \ln(\mathbf{x}+3)}$
- $\boldsymbol{f_2(x) = e^{-2x} + (x-2)^2}$
- $\boldsymbol{f_3(x) = e^{x} \cdot (x^3 - 1) + (x-1) \cdot \sin x}$

Αρχικά, παρουσιάζονται οι γραφικές παραστάσεις των συναρτήσεων στο διάστημα που μας ενδιαφέρει.



Παρακάτω παρουσιάζονται οι κώδικες του κάθε αλγορίθμου. Τυπώνει το τελικό διάστημα αναζήτησης και τον αριθμό επαναλήψεων.

- **<u>Μέθοδος της διχοτόμου</u>**

```
function [a,b,k] = bisector_algorithm(f,a1,b1,epsilon,l)

k=1;
a=a1;
b=b1;
while (b1-a1>l)
        x1= ((a1+b1)/2) - epsilon;
        x2=((a1+b1)/2) + epsilon;
        if (f(x1) < f(x2))
            b1=x2;
        else
            a1=x1;

        end
        k=k+1;
end
disp(k);
disp([a1 b1]);
return;
end
```

Αποτελέσματα:

```
>> bisector_algorithm(f1,-1,3,0.001,0.01)
    10

    1.1474    1.1572
```

```
>> bisector_algorithm(f2,-1,3,0.001,0.01)
    10

    2.0141    2.0239
```

```
>> bisector_algorithm(f3,-1,3,0.001,0.01)
    10

    0.5149    0.5247
```

- **<u>Μέθοδος του χρυσού τομέα</u>**

```
function [a,b,k] = golden_algorithm(a1,b1,l,f)
gamma=0.618;
k=1;
a=a1;
b=b1;
x1 = a1+(1-gamma)*(b1-a1);
x2= a1+gamma*(b1-a1);
f1=f(x1);
f2=f(x2);

while (b-a>=l)
    if f1>f2
        a=x1;
        x1=x2;
        x2=a +gamma*(b-a);
        f1=f2;
        f2=f(x2);
    elseif (f1<f2)
        b=x2;
        x2=x1;
        x1=a +(1-gamma)*(b-a);
        f2=f1;
        f1=f(x1);
    end
    k=k+1;
end
disp(k);
disp([a,b]);
return
end
```

Αποτελέσματα:

```
>> golden_algorithm(-1,3,0.01,f1)
    14

    1.1441    1.1516
```

```
>> golden_algorithm(-1,3,0.01,f2)
    14

    2.0155    2.0231
```

```
>> golden_algorithm(-1,3,0.01,f3)
    14

    0.5154    0.5232
```

o **Μέθοδος Fibonacci**

Τυπώνει τον αριθμό των επαναλήψεων και επιστρέφει το τελικό διάστημα αναζήτησης.

```
function [a,b,k] = fibonacci_algorithm(a1,b1,epsilon,l,f)
a=a1;
b=b1;
n=0;
while fibonacci(n)<=((b-a)/l)
    n = n+1;
end
disp(n-1);
fib=zeros(1,n+1); %αποθήκευση αριθμών fibonacci
for i= 1 : n+1
    fib(i) = fib(i) + fibonacci(i-1);
end
    x1 = a + (fib(n-1)/fib(n+1))*(b-a);
    x2 = a + (fib(n)/fib(n+1))*(b-a);
    f1 = f(x1);
    f2 = f(x2);
    k=1;
for j= 1 : (n-2)
    if (f(x1)>f(x2))
        a=x1;
        x1=x2;
        x2= a + (fib(n-k)/fib(n-k+1))*(b-a);
        if (k == (n-2))
            x1n = x1;
            x2n = x1 + epsilon;
            if (f(x1n) > f(x2n))
                a=x1n;
            else
                b=x2n;
            end
            disp([a b]);
            return
        else
            f1 = f2;
            f2 = f(x2);
            k = k+1;
        end
    elseif (f(x1)<f(x2))
        b = x2;
        x2 = x1;
        x1= a + ((fib(n-k-1)/fib(n-k+1))*(b-a));
        if k == (n-2)
            x1n = x1;
            x2n = x1 + epsilon;
            if (fx1n) > f(x2n)
                a=x1n;
            else
                b=x2n;
            end
            disp([a b]);
            return
        else
            f2 = f1;
            f1 = f(x1);
            k = k+1;
        end
    end
end
disp([a b]);
end
```

Αποτελέσματα:

```
>> fibonacci_algorithm(-1,3,0.001,0.01,f1)
    14

    1.1443    1.1574
```

```
>> fibonacci_algorithm(-1,3,0.001,0.01,f2)
    14

    2.0098    2.0230
```

```
>> fibonacci_algorithm(-1,3,0.001,0.01,f3)
    14

    0.5148    0.5279
```

- ## **Μέθοδος διχοτόμου με χρήση παραγώγων**

Ο αλγόριθμος επιστρέφει το τελικό διάστημα αναζήτησης. Στην περίπτωση που καταλήξει στο ίδιο το σημείο ελαχίστου τυπώνει το σημείο αυτό και επιστρέφει ένα άκυρο διάστημα [-1 ,-1] δεδομένου ότι ξεκινάμε την αναζήτηση στο [-1,3].

```
function [a,b,k]= bisector_algorithm_df(a1,b1,l,f)
syms x;
a=a1; b=b1;
lamda=l/(b1-a1);
ni=log(lamda)/log(1/2);
n=ceil(ni);
k=1;
disp(n);
df=diff(f,x);
Df=matlabFunction(df);
while k<=n
    xk=(a+b)/2;
    Dfk=Df(xk);
    if (Dfk == 0)
        disp(xk);
        disp([-1,-1]);
        return
    elseif (Dfk>0)
        b=xk;
    else
        a=xk;
    end
    if k==n
        disp([a b]);
        return
    else
        k=k+1;
    end
end
disp(k);
end
```

```
>> bisector_algorithm_df(-1,3,0.01,f1)
     9

    1.1484    1.1562
```

```
>> bisector_algorithm_df(-1,3,0.01,f2)
     9

    2.0156    2.0234
```

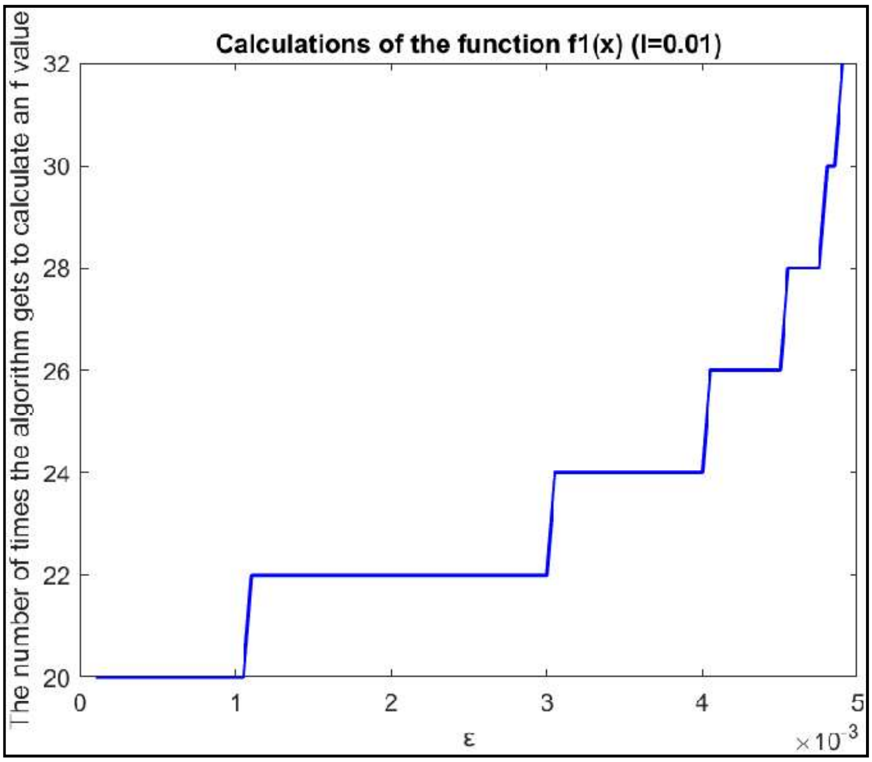
```
>> bisector_algorithm_df(-1,3,0.01,f3)
     9

    0.5156    0.5234
```

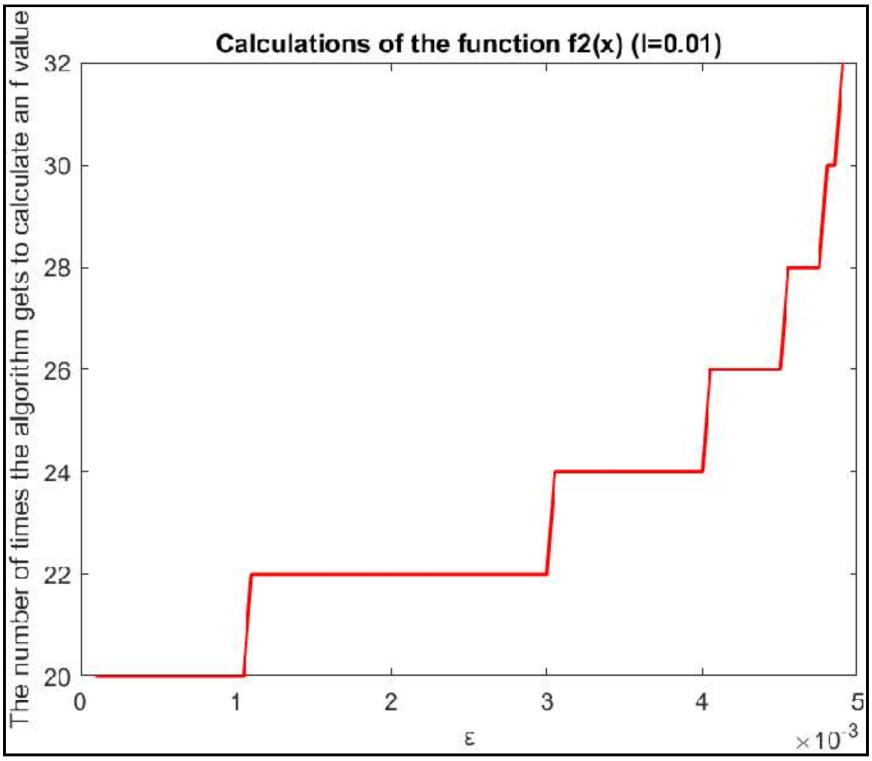
<u>Θέμα 1ο : Αλγόριθμος της Διχοτόμου</u>

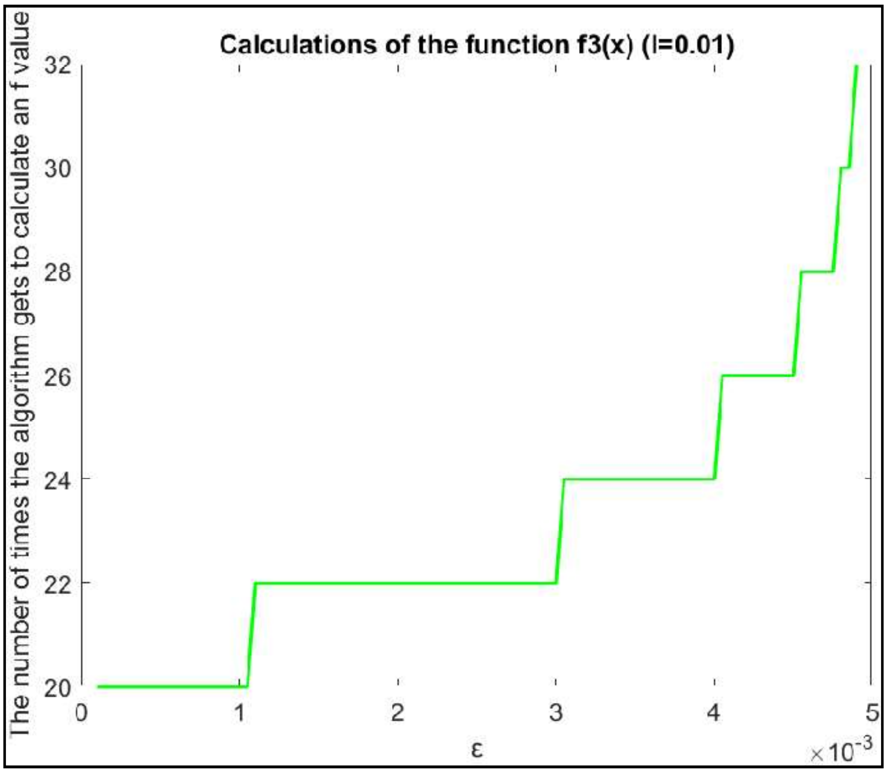
Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις του πλήθους υπολογισμών της f συναρτήσει του **ε** για l = 0.01 σταθερό. Για το **ε** τέθηκε ο περιορισμός **ε <** $\boldsymbol{l/2}$**.** Οι κλήσεις της συνάρτησης για τον συγκεκριμένο αλγόριθμο είναι **n=2k** , με **k** τον αριθμό των βημάτων για **bk – ak < l**.

Σχήμα 1. Αριθμός υπολογισμών της συνάρτησης $f1(x)$ για σταθερό l.



Σχήμα 2. Αριθμός υπολογισμών της συνάρτησης $f2(x)$ για σταθερό l.

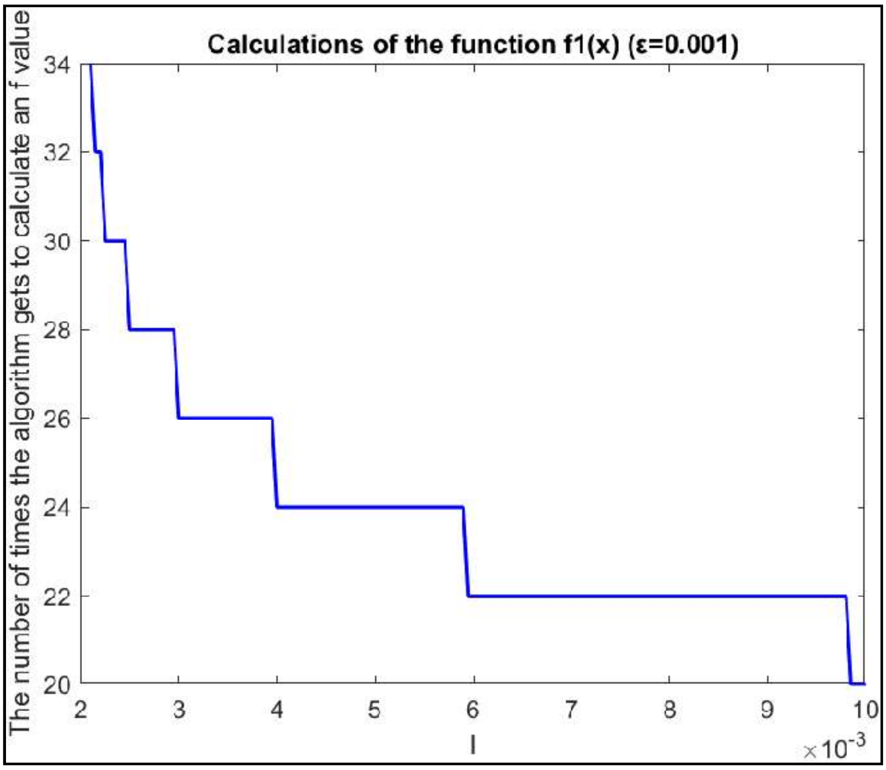




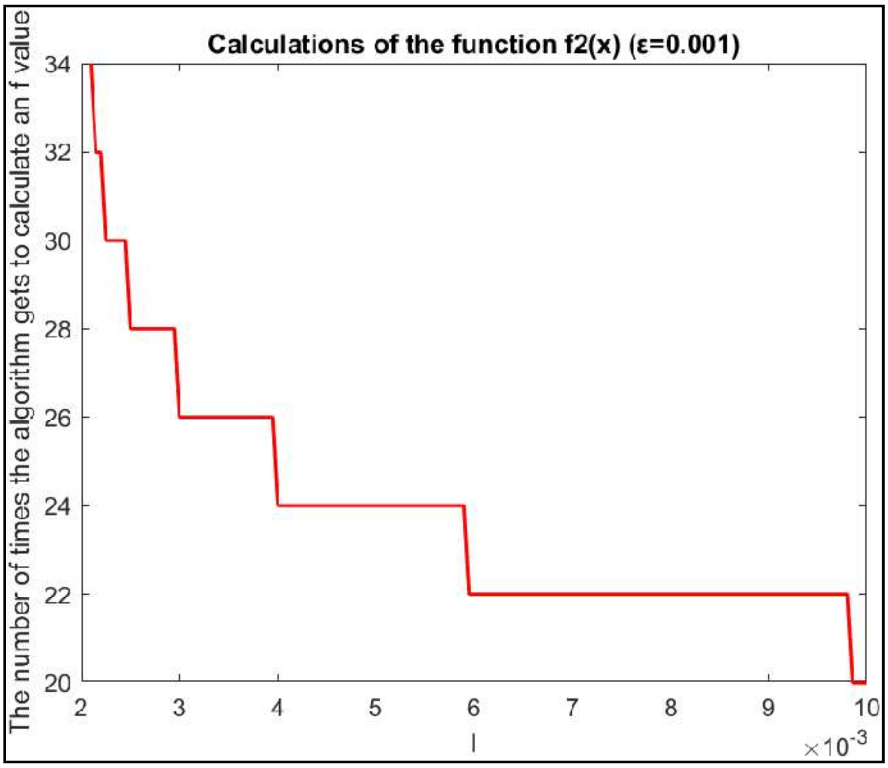
Σχήμα 3. Αριθμός υπολογισμών της συνάρτησης $f3(x)$ για σταθερό l.

Καθώς, το **ε** αυξάνεται, αυξάνεται και ο αριθμός των βημάτων του αλγορίθμου και αυξάνεται, κατά συνέπεια και ο αριθμός υπολογισμών της συνάρτησης . Επιπλέον, αν δεν ικανοποιείται η συνθήκη $\boldsymbol{\varepsilon < l/2}$ τότε, το εύρος του διαστήματος (**bk – ak**) δεν θα γίνει ποτέ μικρότερο του l .

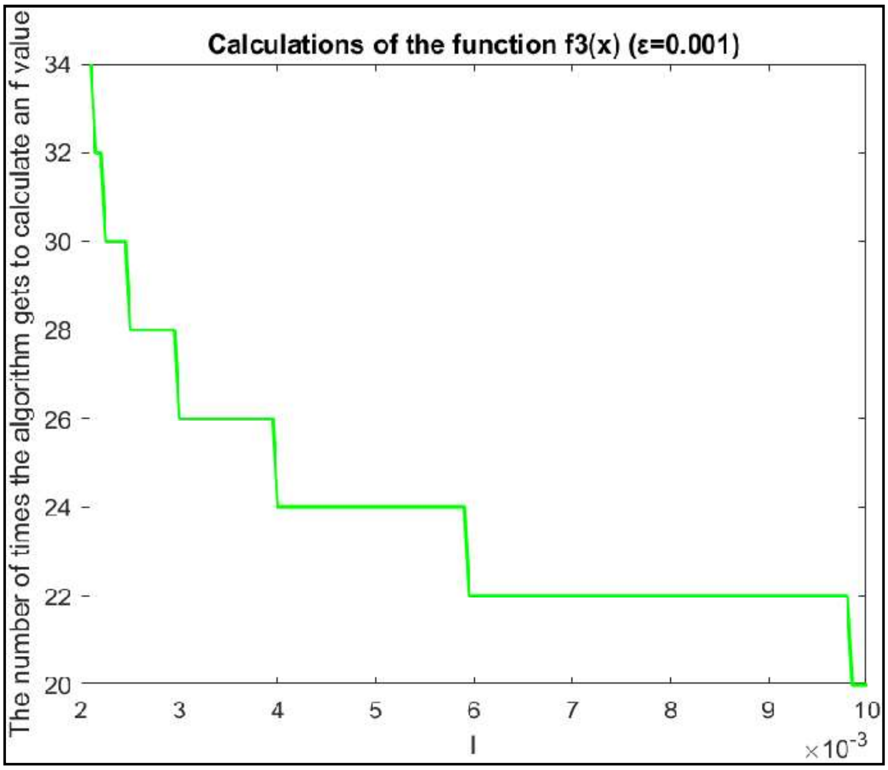
Έπειτα, ακολουθούν οι γραφικές παραστάσεις του πλήθους υπολογισμών της f συναρτήσει του **l** για **ε = 0.001** σταθερό. Η μεταβολή του **l**, όπως παρατηρείται παρακάτω, επιδρά διαφορετικά από τη μεταβολή του **ε**. Συγκεκριμένα, με την αύξηση του **l** επακολουθεί μείωση του αριθμού υπολογισμών της κάθε συνάρτησης f .



Σχήμα 4. Αριθμός υπολογισμών της συνάρτησης $f1(x)$ για σταθερό ε.



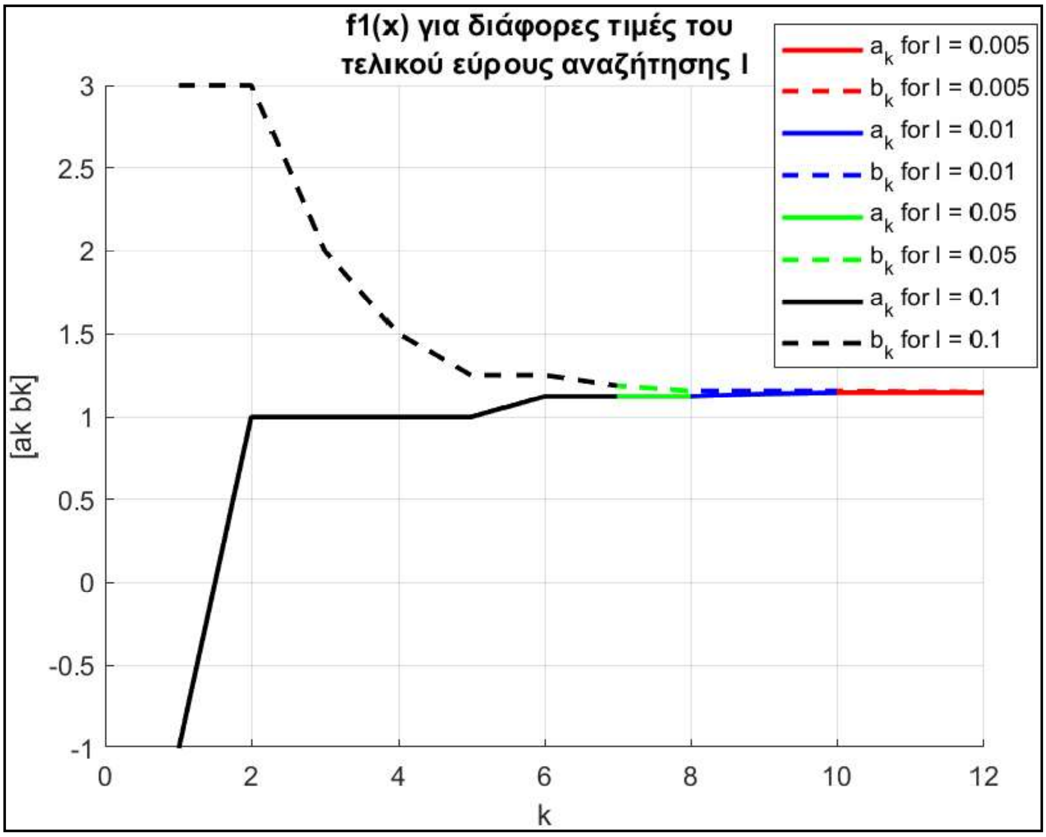
Σχήμα 5. Αριθμός υπολογισμών της συνάρτησης $f2(x)$ για σταθερό ε.



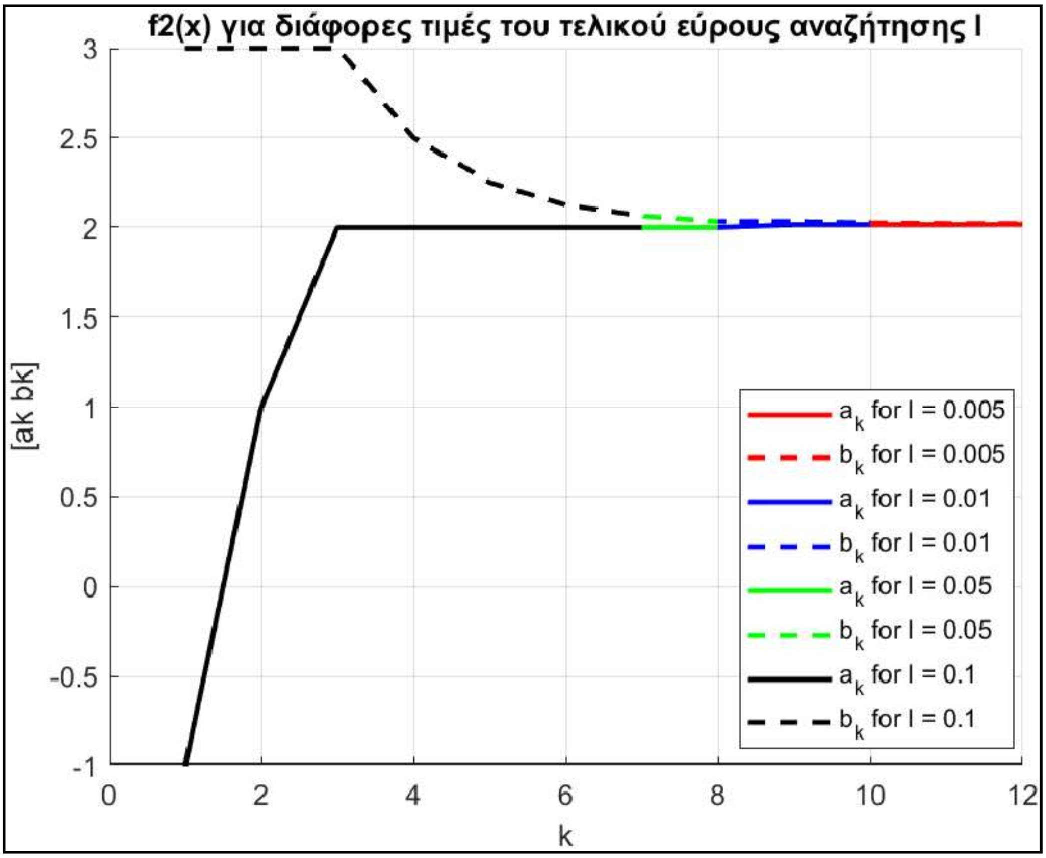
Σχήμα 6. Αριθμός υπολογισμών της συνάρτησης $f3(x)$ για σταθερό ε.

Παρατηρούμε τρία ίδια γραφήματα και στις δύο περιπτώσεις. Το αποτέλεσμα αυτό επιβεβαιώνει ότι οι υπολογισμοί της f εξαρτώνται πρωτίστως από το πλήθος επαναλήψεων k , το οποίο καθορίζεται από τις τιμές των l και ε , όταν και οι τρεις συναρτήσεις μελετώνται πάνω στο ίδιο αρχικό διάστημα [a , b] . Επιπλέον, σε κάθε επανάληψη εκτελούνται δύο υπολογισμοί σημείων που σημαίνει ότι ο αριθμός υπολογισμών της συνάρτησης αυξάνεται κατά δύο σε κάθε επανάληψη.

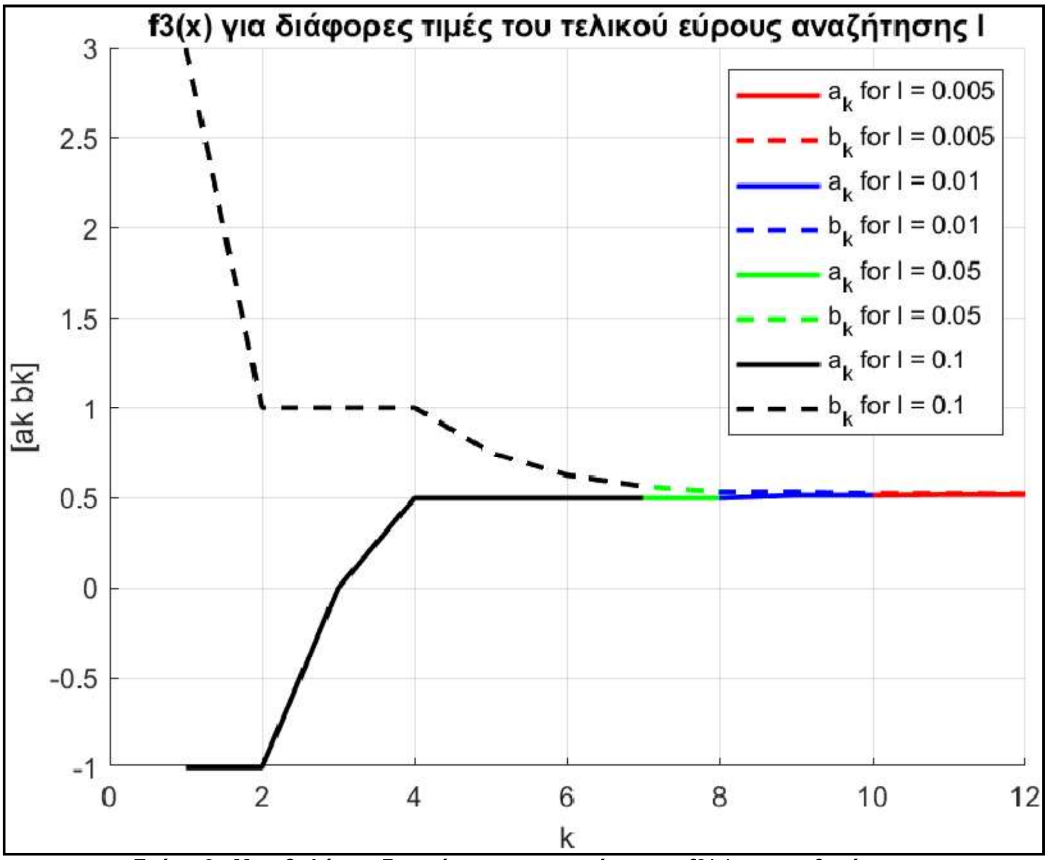
Στη συνέχεια, δημιουργήθηκαν για κάθε συνάρτηση οι γραφικές παραστάσεις των άκρων των διαστημάτων για τέσσερις διακριτές τιμές **l = [0.005 , 0.01 , 0.05 , 0.1]** . Σε κάθε διάγραμμα, η καμπύλη από «κάτω» δείχνει το άκρο **a** σε κάθε βήμα **k**, ενώ η «πάνω» καμπύλη δείχνει το άκρο **b**. Για κάθε τιμή του **l**, τα άκρα που αντιστοιχούν στο ίδιο διάστημα έχουν ίδιο χρώμα. Όπως φαίνεται παρακάτω, η μείωση της τιμής του **l** οδηγεί σε μικρότερο διάστημα αναζήτησης και σε περισσότερες επαναλήψεις , αφού ο αλγόριθμος τερματίζει όταν **bk - ak < l** . Πολύ μικρές μεταβολές του **l** οδηγούν στο ίδιο αριθμό επαναλήψεων και κατά συνέπεια στο ίδιο τελικό διάστημα αναζήτησης. Έτσι, σε κάθε επανάληψη ο αλγόριθμος επιλέγει ακριβώς το ίδιο διάστημα, ανεξαρτήτως **l** , για αυτό υπάρχει επικάλυψη μεταξύ των καμπυλών. Τέλος, όπου υπάρχει μόνο μαύρο χρώμα στο διάγραμμα, περιλαμβάνονται και οι τέσσερις τιμές του **l**, ενώ στις περιοχές με πράσινο περιλαμβάνονται και το μπλε και το κόκκινο, όπου μπλε περιλαμβάνεται και κόκκινο , και το κόκκινο αντιστοιχεί μόνο στο **l** = 0.005 .



Σχήμα 7. Μεταβολή του διαστήματος της συνάρτησης f1(x) για σταθερό ε.



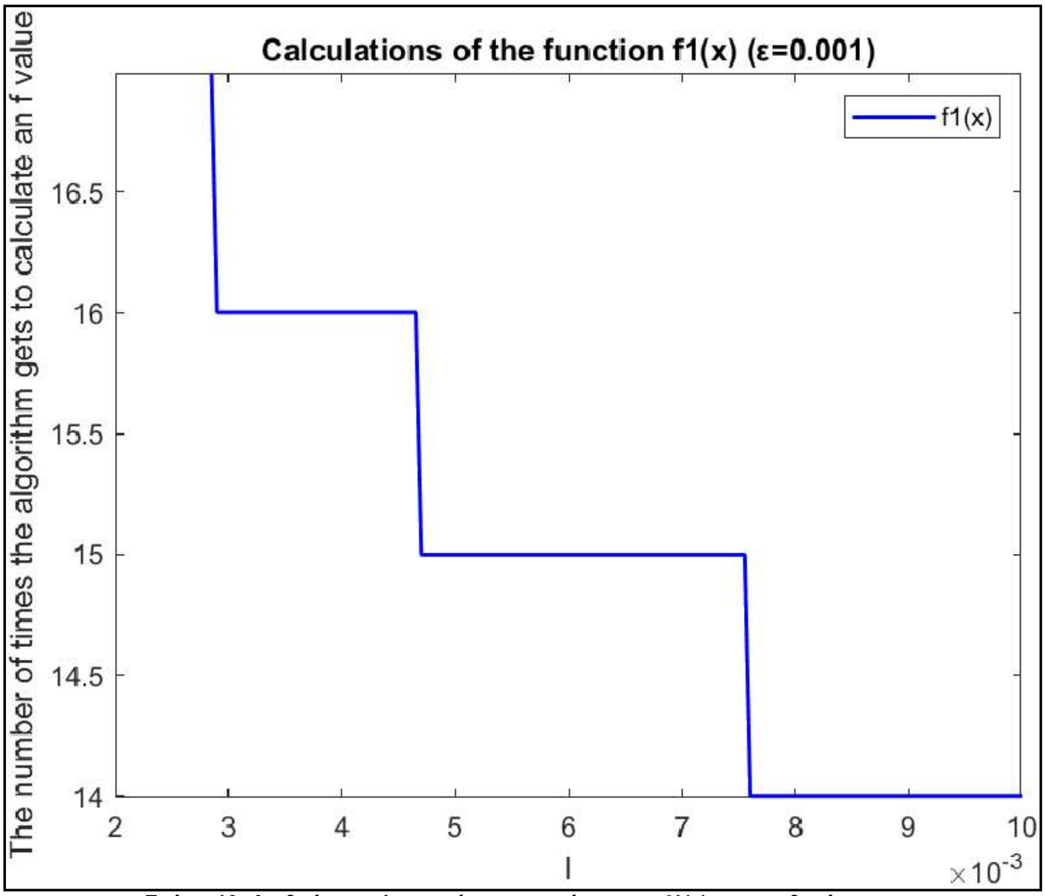
Σχήμα 8. Μεταβολή του διαστήματος της συνάρτησης f2(x) για σταθερό ε.



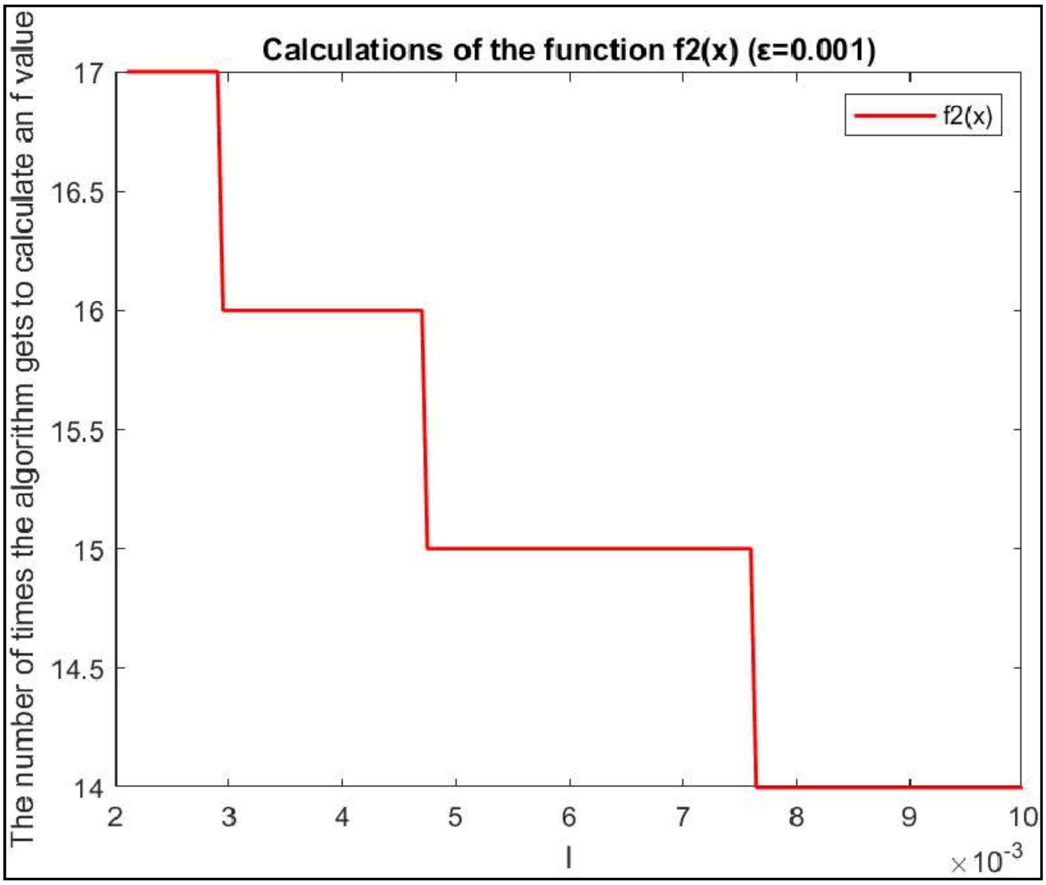
Σχήμα 9. Μεταβολή του διαστήματος της συνάρτησης f3(x) για σταθερό ε.

Θέμα 2ο – Αλγόριθμος του Χρυσού Τομέα

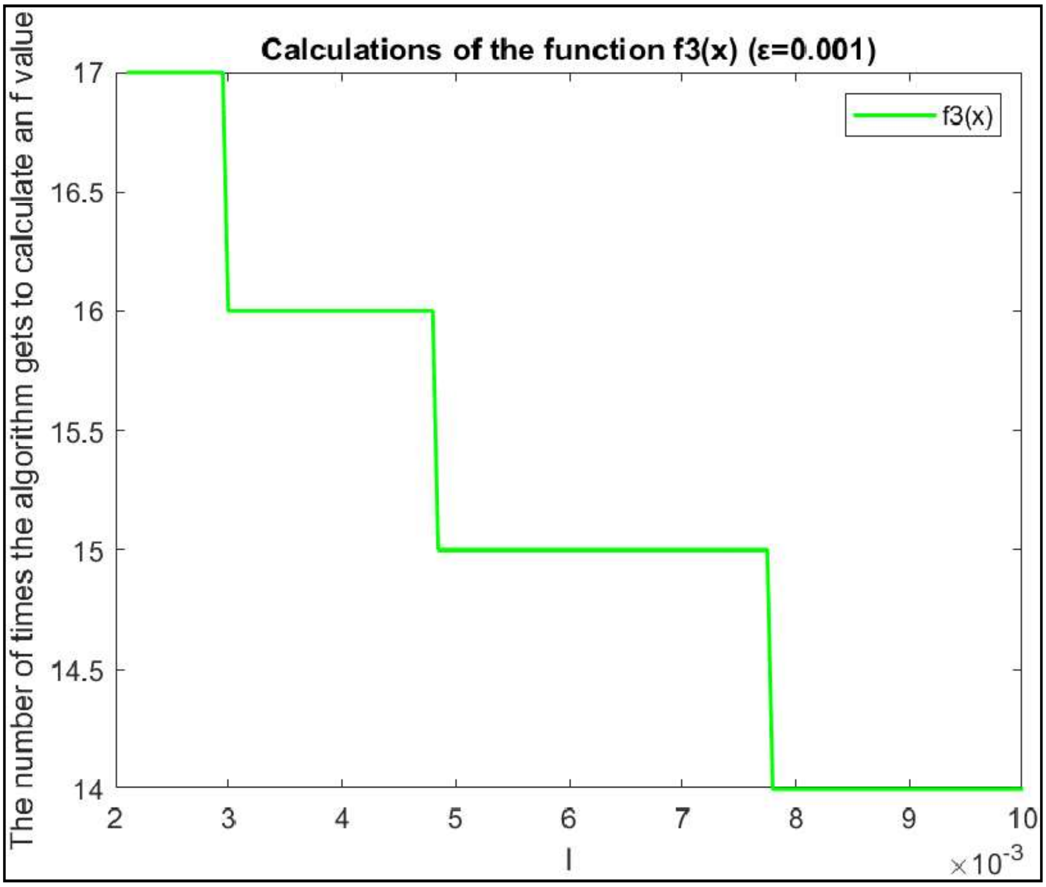
Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις του πλήθους υπολογισμών της f συναρτήσει του **l** για **ε = 0.001** σταθερό. Για τον αλγόριθμο του χρυσού τομέα, ο συνολικός αριθμός κλήσεων της συνάρτησης είναι $n = k + 1$, όπου k ο αριθμός των βημάτων μέχρι να ισχύσει **bk – ak < l** . Πράγματι, απαιτεί μικρότερο αριθμό υπολογισμών , αφού οι τιμές της συνάρτησης για **X1k** και **X2k** υπολογίζονται μόνο στην αρχή και έπειτα υπολογίζεται μόνο μια τιμή της.



Σχήμα 10. Αριθμός υπολογισμών της συνάρτησης $f1(x)$ για σταθερό ε.



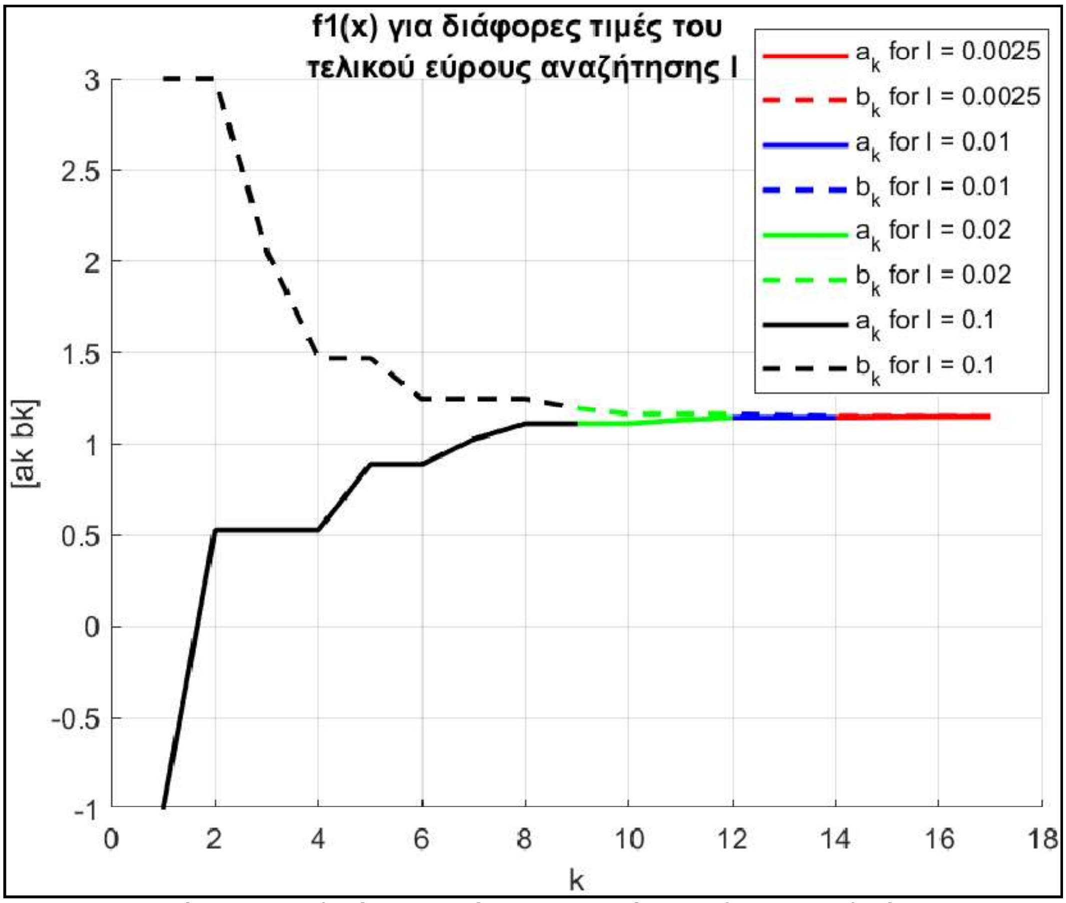
Σχήμα 11. Αριθμός υπολογισμών της συνάρτησης $f2(x)$ για σταθερό ε.



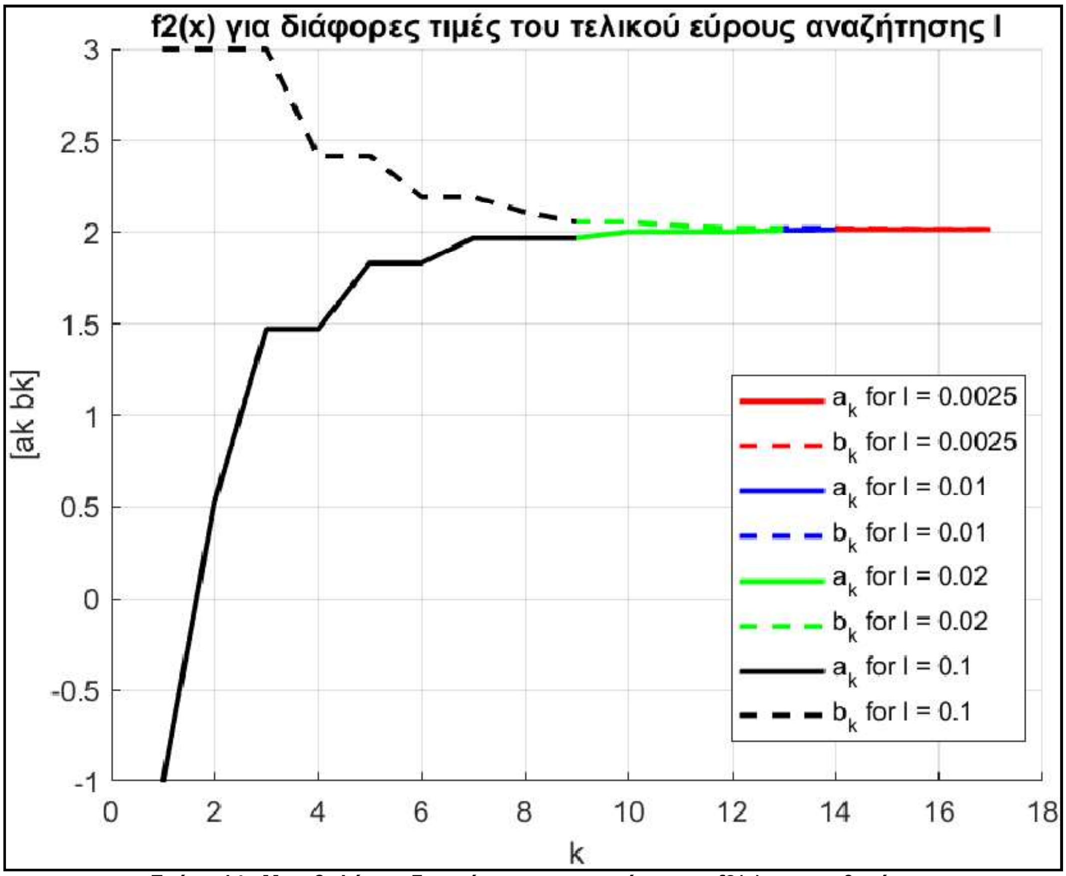
Σχήμα 12. Αριθμός υπολογισμών της συνάρτησης $f3(x)$ για σταθερό ε.

Και πάλι με την ίδια λογική με πριν τα γραφήματα ταυτίζονται. Ενώ η διαφορά εντοπίζεται στο βήμα αύξησης το οποίο είναι πλέον ίσο με την μονάδα.

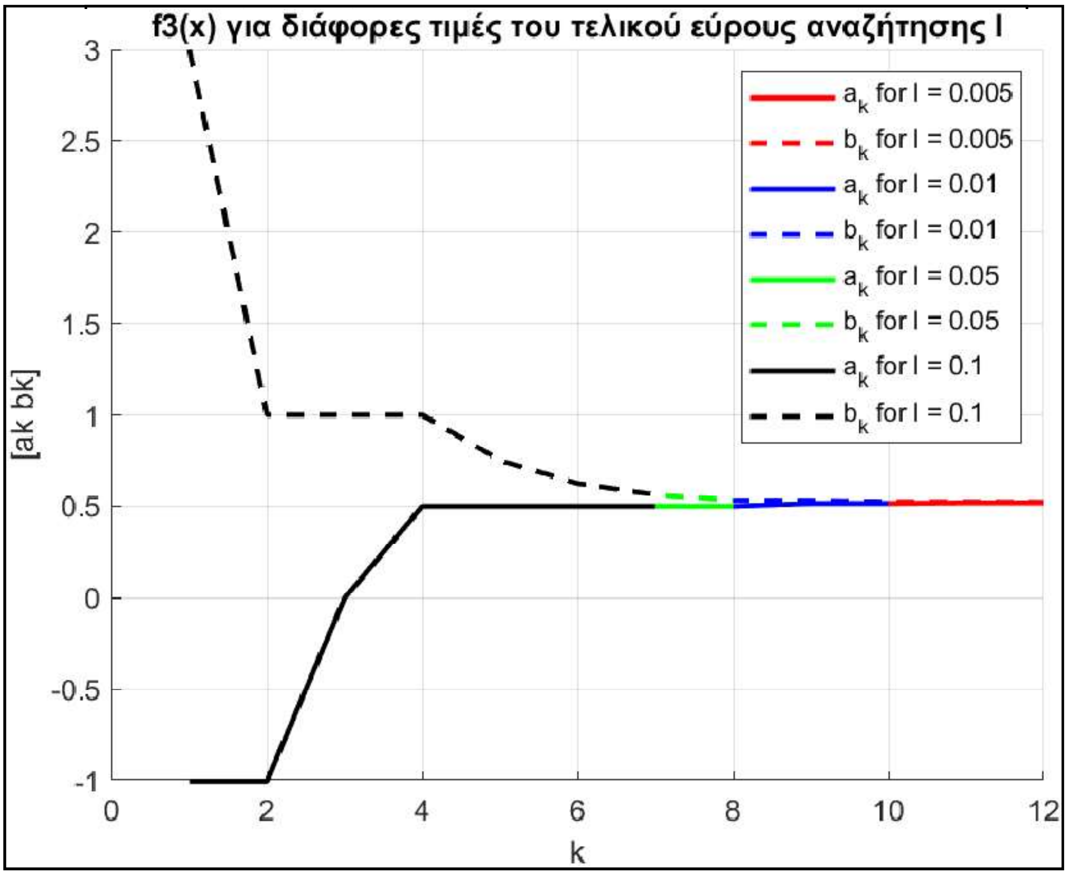
Ακολουθούν οι γραφικές παραστάσεις των όρων (k , ak) και (**k** ,**bk**) για 4 διακριτές τιμές **l = [0.0025 , 0.01 , 0.02 , 0.1]** .



Σχήμα 13. Μεταβολή του διαστήματος της συνάρτησης f1(x) για σταθερό ε.



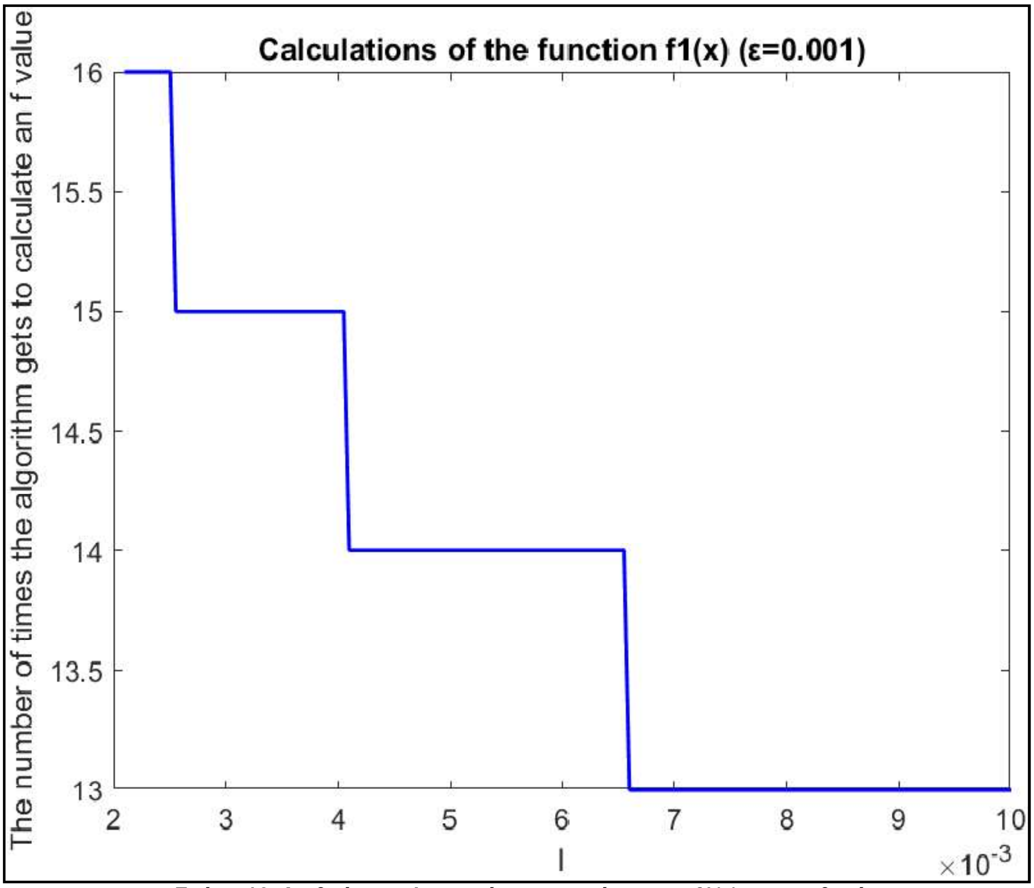
Σχήμα 14. Μεταβολή του διαστήματος της συνάρτησης f2(x) για σταθερό ε.



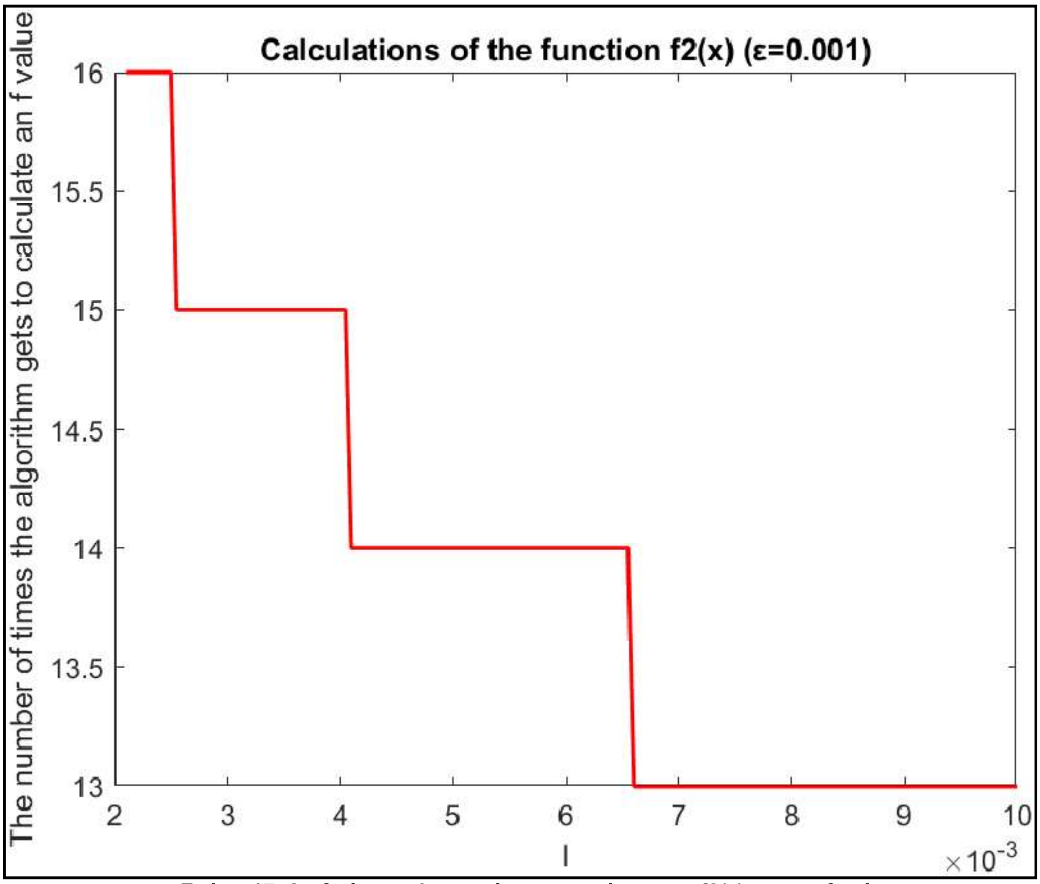
Σχήμα 15. Μεταβολή του διαστήματος της συνάρτησης f3(x) για σταθερό ε.

Θέμα 3ο – Αλγόριθμος Fibonacci

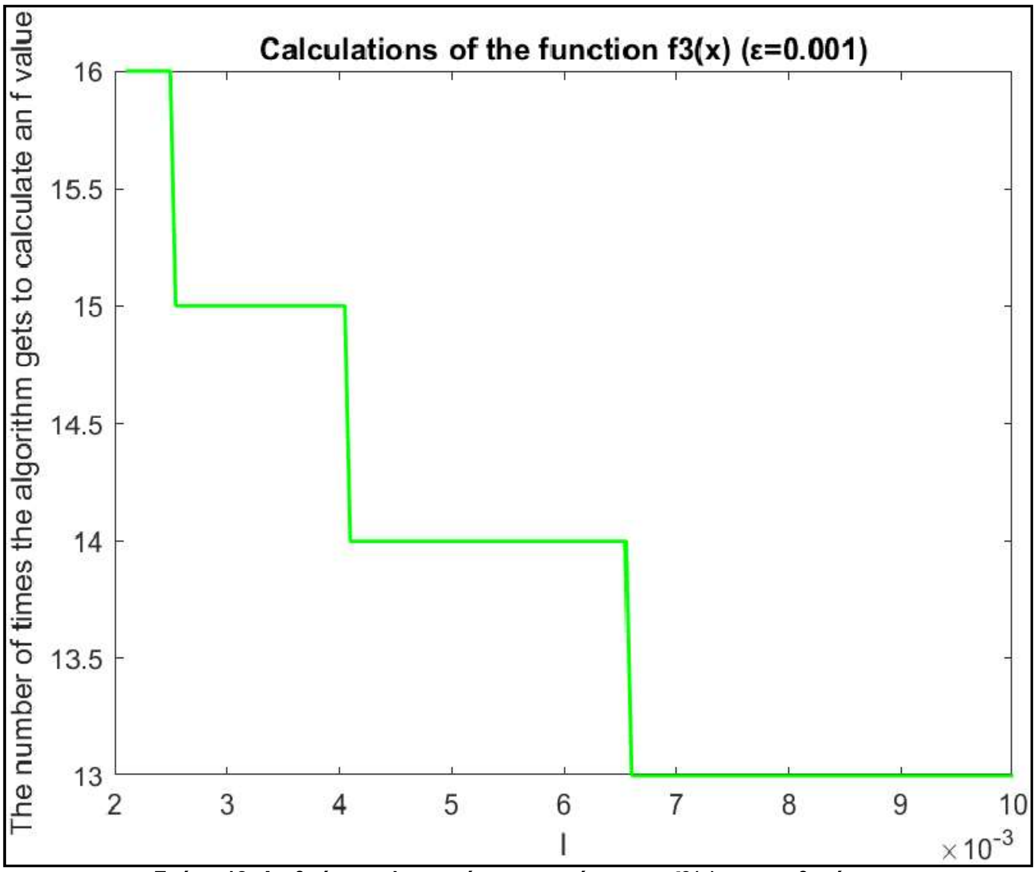
Παρακάτω παρουσιάζονται οι γραφικές παραστάσεις του πλήθους υπολογισμών της f συναρτήσει του **l** για **ε = 0.001** σταθερό με χρήση της μεθόδου Fibonacci. Για τον αλγόριθμο, όπως και στον αλγόριθμο του χρυσού τομέα , ο συνολικός αριθμός κλήσεων της συνάρτησης είναι $n = k + 1$. Τα βήματα που εκτελεί ο αλγόριθμος εξαρτώνται από τη σχέση $\mathbf{F_n} > \frac{\mathbf{b-a}}{\mathbf{l}}$. Επίσης, πέρα από τις τιμές της συνάρτησης για x_k , αποθηκεύονται και οι αριθμοί της σειράς Fibonacci μέχρι το n, ώστε να επιταχυνθεί η διαδικασία του αλγορίθμου.



Σχήμα 16. Αριθμός υπολογισμών της συνάρτησης $f1(x)$ για σταθερό ε.



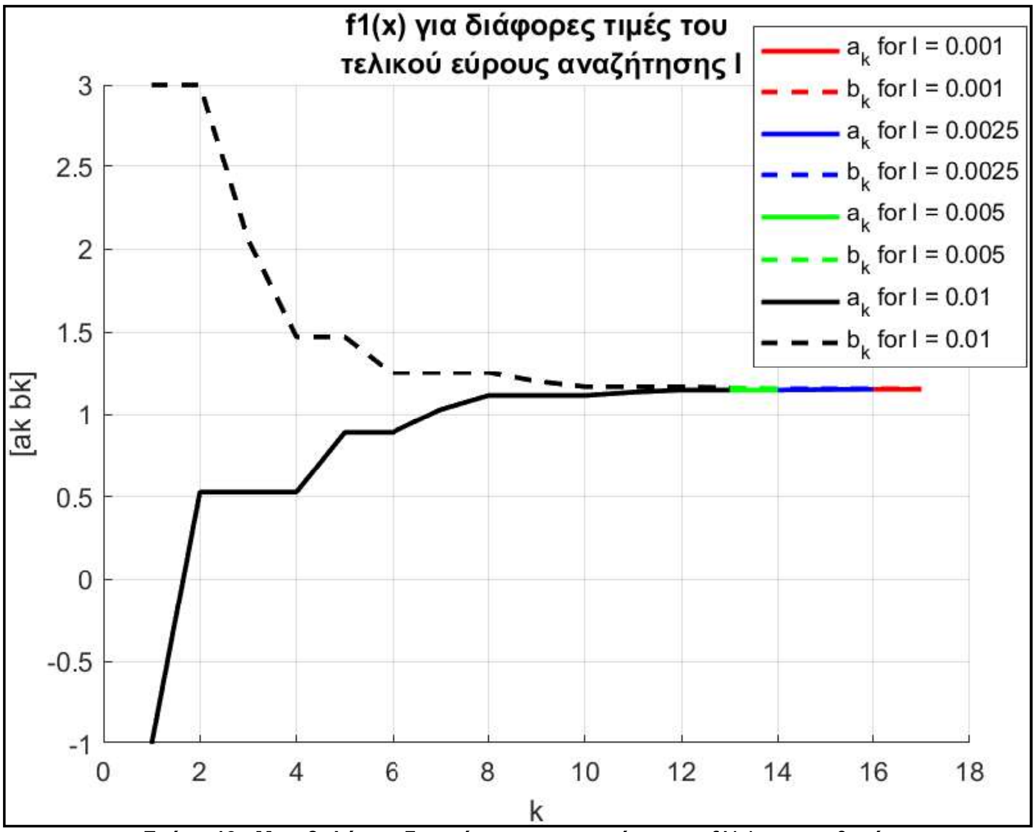
Σχήμα 17. Αριθμός υπολογισμών της συνάρτησης $f2(x)$ για σταθερό ε.



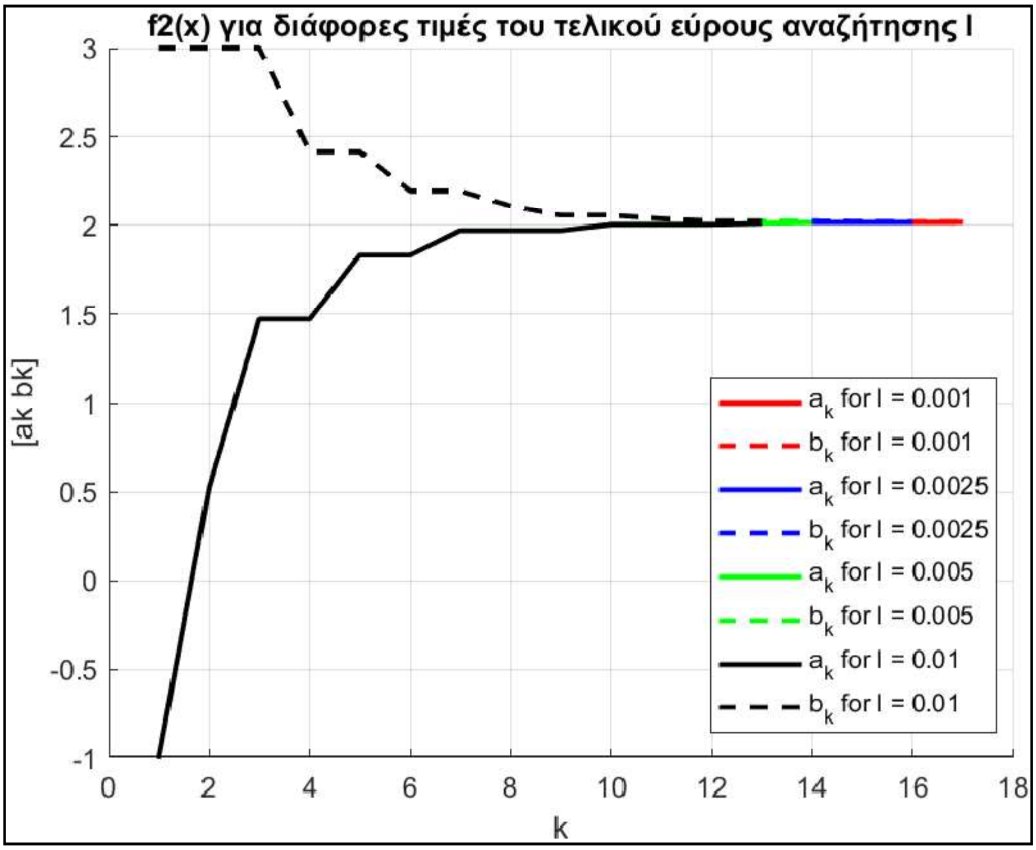
Σχήμα 18. Αριθμός υπολογισμών της συνάρτησης $f3(x)$ για σταθερό ε.

Και σε αυτόν τον αλγόριθμο το βήμα αύξησης είναι μοναδιαίο λόγω ενός νέου υπολογισμού που χρειάζεται σε κάθε επανάληψη. Εξαιρείται η τελική επανάληψη κατά την οποία ο αλγόριθμος υπολογίζει την f σε δύο σημεία. Επίσης, λαμβάνουμε τα ίδια γραφήματα για όλες τις συναρτήσεις αφού, ο αλγόριθμος θα εκτελέσει τον ίδιο αριθμό επαναλήψεων.

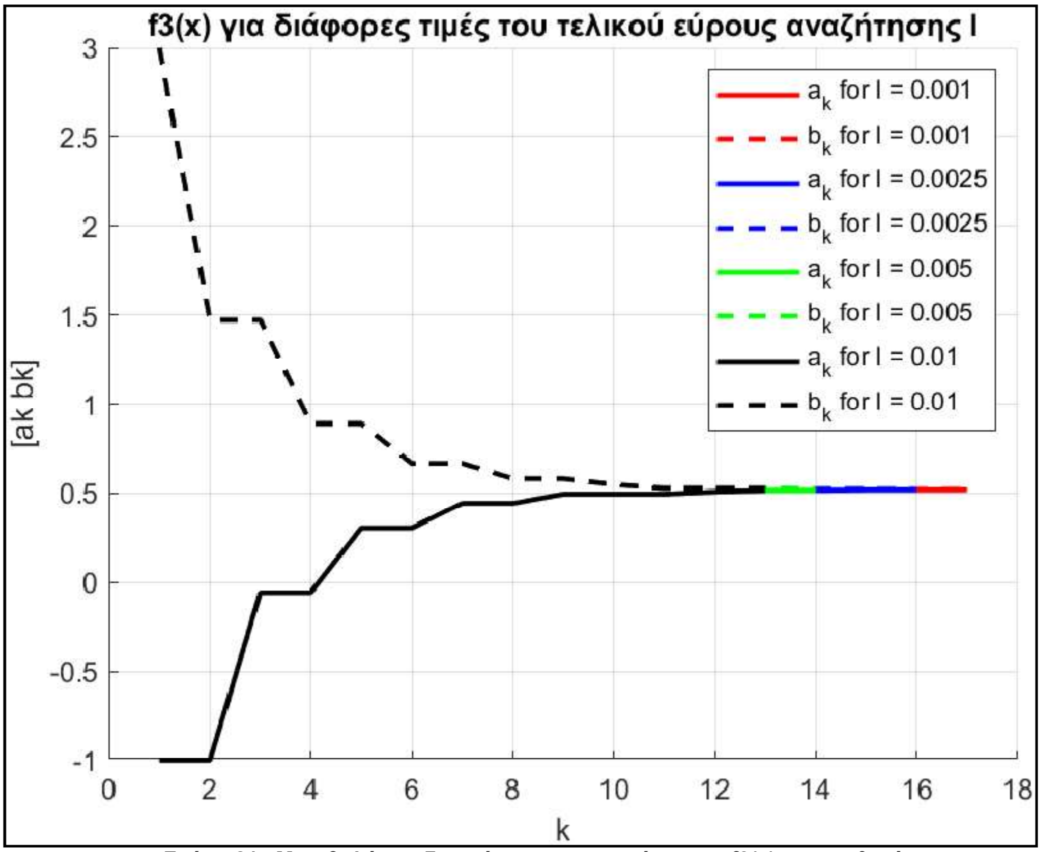
Ακολουθούν οι γραφικές παραστάσεις των όρων (k , ak) και (**k** ,**bk**) για 4 διακριτές τιμές **l = [0.001 , 0.0025 , 0.005 , 0.01]** .



Σχήμα 19. Μεταβολή του διαστήματος της συνάρτησης f1(x) για σταθερό ε.



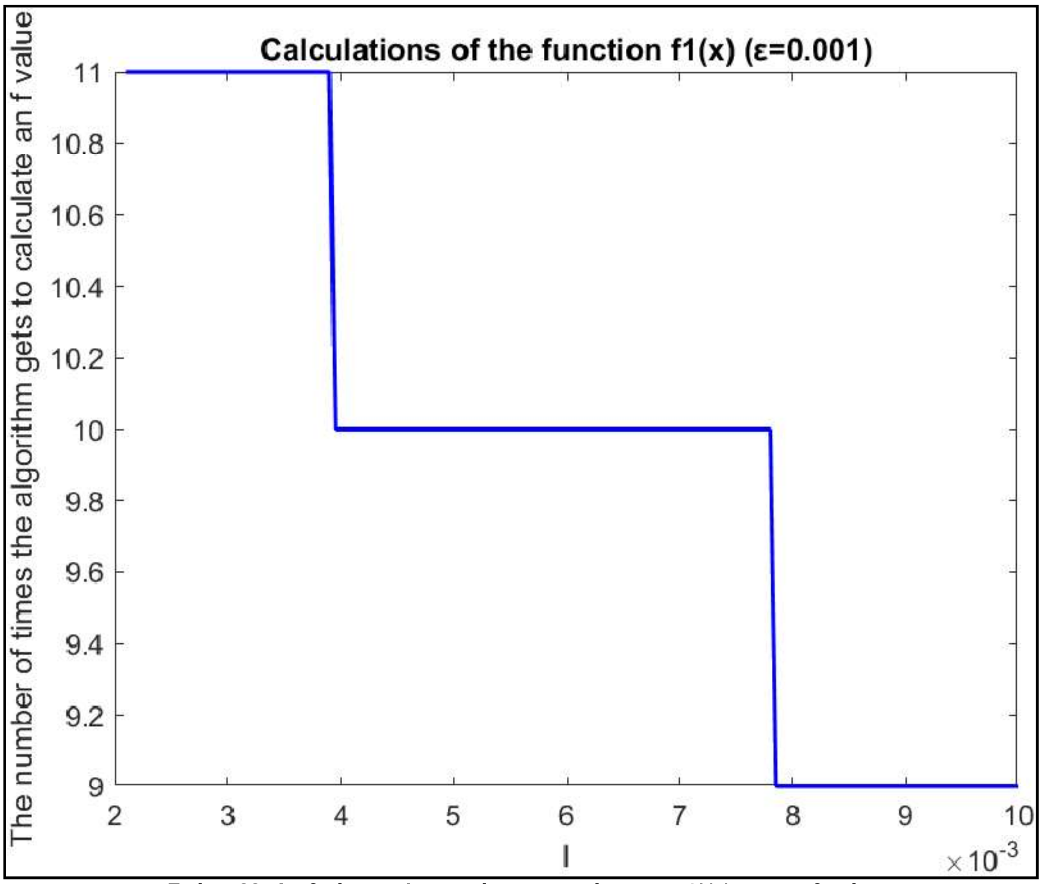
Σχήμα 20. Μεταβολή του διαστήματος της συνάρτησης f2(x) για σταθερό ε.



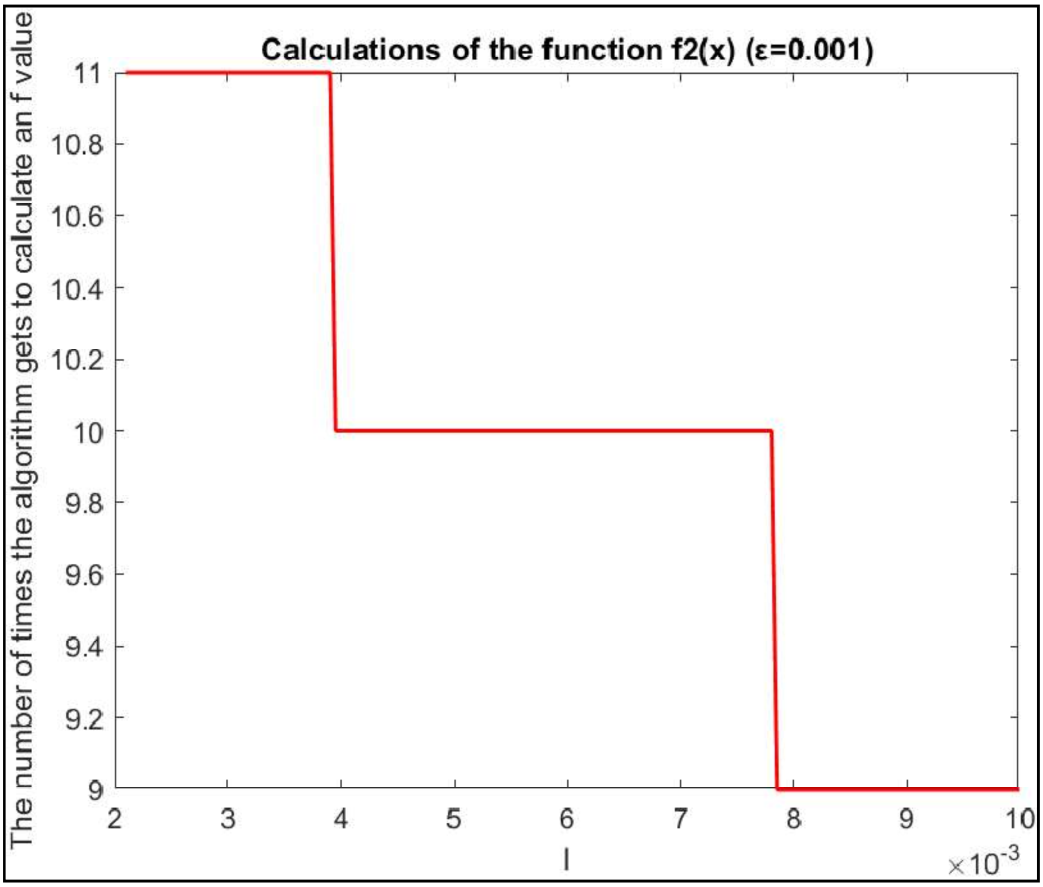
Σχήμα 21. Μεταβολή του διαστήματος της συνάρτησης f3(x) για σταθερό ε.

<u>Θέμα 4ο –Αλγόριθμος της Διχοτόμου με παραγώγους</u>

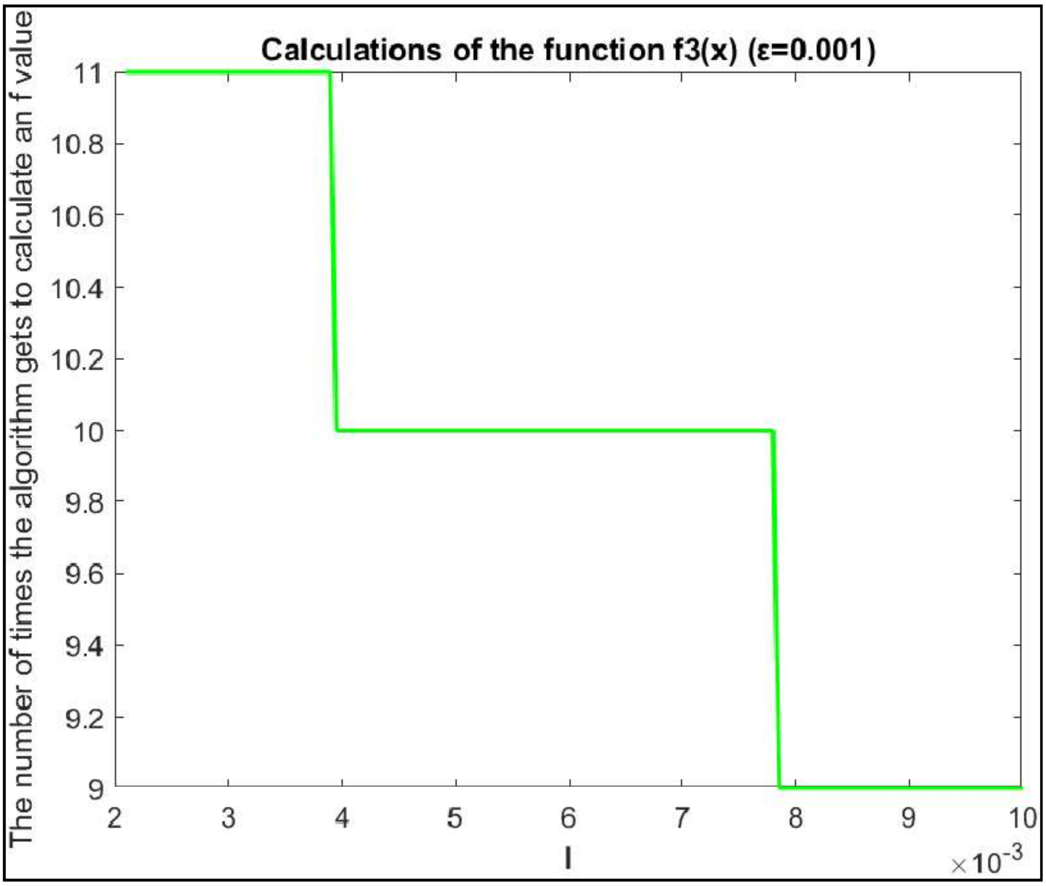
Στη περίπτωση αυτή η γραφική παράσταση θα απεικονίζει το πλήθος υπολογισμών τιμών της παραγώγου της συνάρτησης, καθώς το l μεταβάλλεται. Τα βήματα που θα εκτελέσει ο αλγόριθμος εξαρτώνται από τη σχέση $\left(\frac{1}{2}\right)^{n} \leq \frac{l}{b-a}$, δηλαδή n το μεγαλύτερο, ώστε να ισχύει η σχέση, όπου n ο αριθμός κλήσεων της παραγώγου της f.



Σχήμα 22. Αριθμός υπολογισμών της συνάρτησης $f1(x)$ για σταθερό ε.



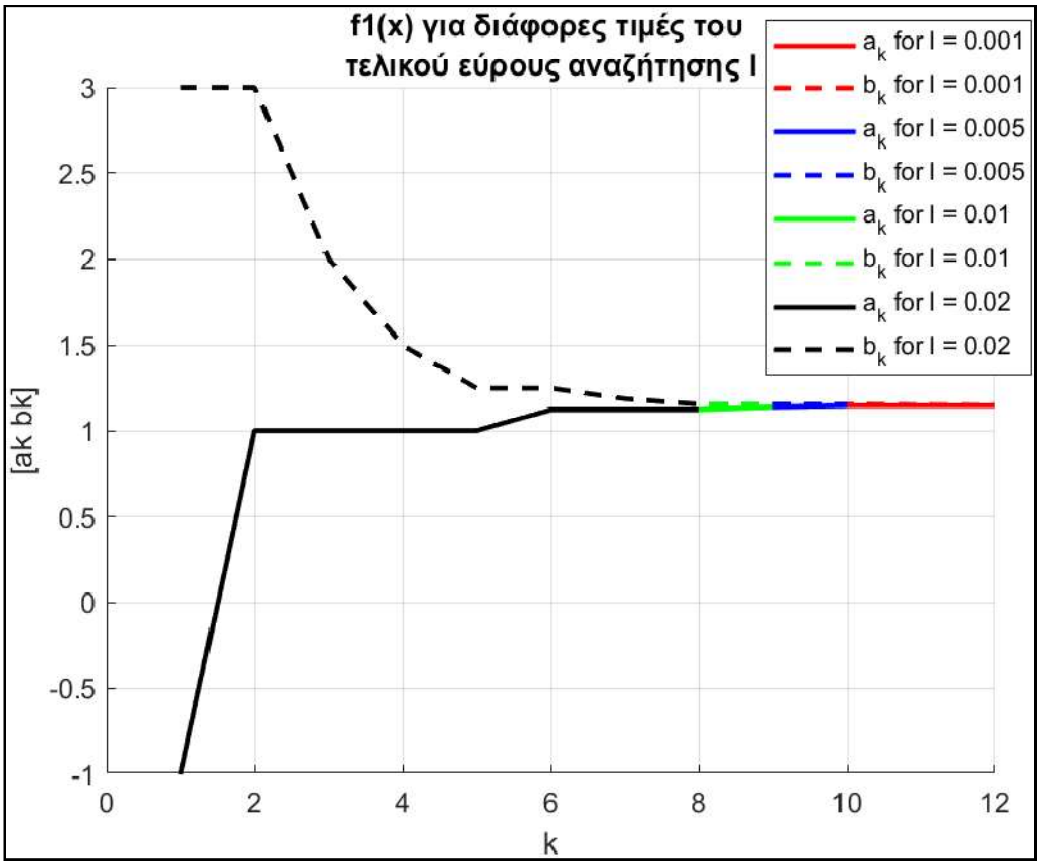
Σχήμα 23. Αριθμός υπολογισμών της συνάρτησης $f2(x)$ για σταθερό ε.



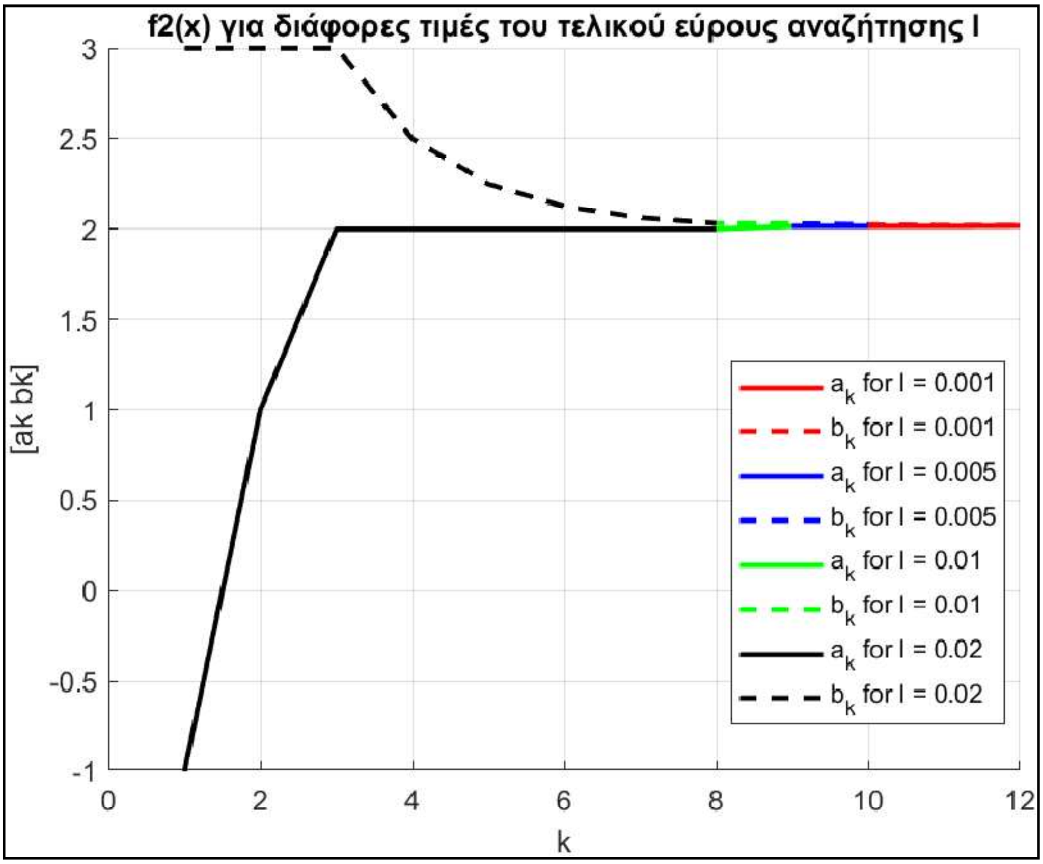
Σχήμα 24. Αριθμός υπολογισμών της συνάρτησης $f3(x)$ για σταθερό ε.

Ο αλγόριθμος θα τρέξει 9,10 ή 11 φορές, καθώς απαιτείται ένας υπολογισμός ανά επανάληψη και η ακρίβεια αυξάνει.

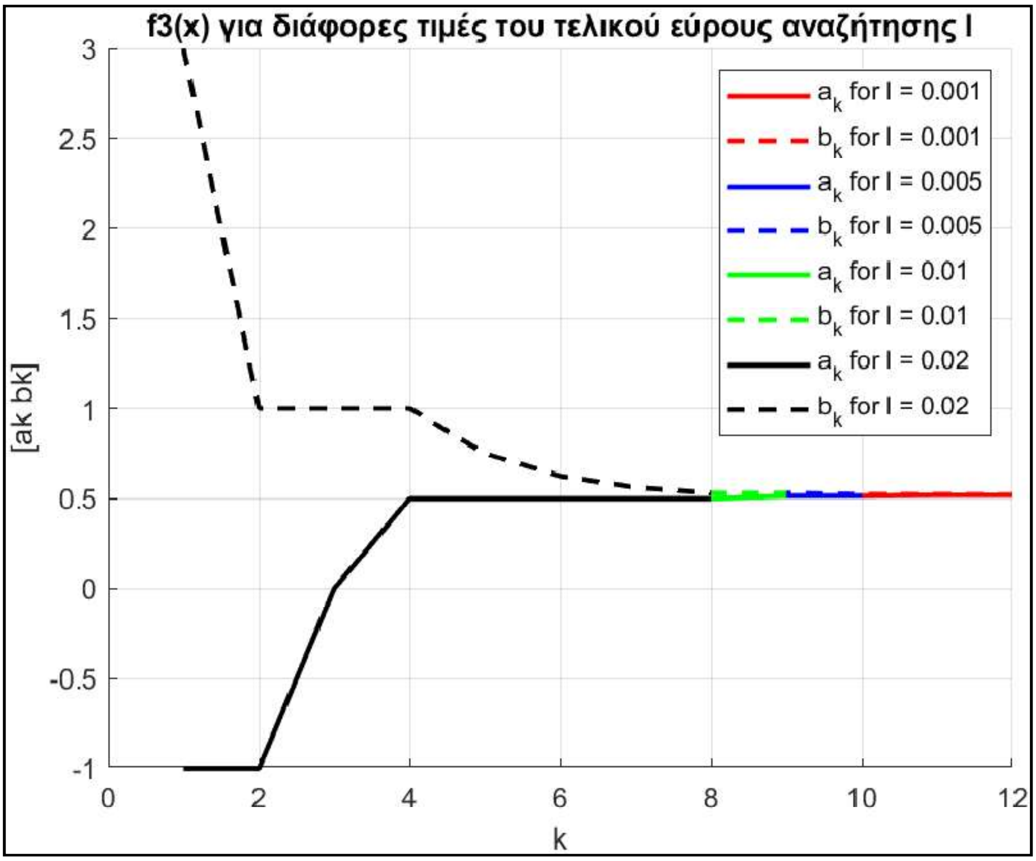
Ακολουθούν οι γραφικές παραστάσεις των όρων (k , ak) και (**k ,bk**) για 4 διακριτές τιμές **l = [0.001 , 0.005 , 0.01 , 0.02]**



Σχήμα 25. Μεταβολή του διαστήματος της συνάρτησης f1(x) για σταθερό ε.



Σχήμα 26. Μεταβολή του διαστήματος της συνάρτησης f2(x) για σταθερό ε.



Σχήμα 27. Μεταβολή του διαστήματος της συνάρτησης f3(x) για σταθερό ε.

Όπως, και στα προηγούμενα διαγράμματα , παρατηρείται επικάλυψη των σημείων που αντιστοιχούν στις επαναλήψεις του αλγορίθμου για μικρές τιμές του l με εκείνα των μεγαλύτερων τιμών.

<u>Τελικοί σχολιασμοί και σύγκριση των υπό μελέτη μεθόδων:</u>

Στον παρακάτω πίνακα συνοψίζονται οι κλήσεις των συναρτήσεων f για κάθε μέθοδο συναρτήσει του διαστήματος l , με **l= 0.1** (ε=0.001).

l=0.1	Αλγόριθμος Διχοτόμου	Αλγόριθμος Χρυσού Τομέα	Αλγόριθμος Fibonacci	Αλγόριθμος Διχοτόμου με χρήση παραγώγων
f1(x)	10	14	14	9
f2(x)	10	14	14	9
f3(x)	10	14	14	9

- <u>Μέθοδος διχοτόμου χωρίς τη χρήση παραγώγου</u>.

Επιστρέφει διαστήματα που περιλαμβάνουν το σημείο ελαχίστου και για τις τρείς συναρτήσεις, ικανοποιώντας συγχρόνως την απαίτηση το εύρος του τελικού διαστήματος αναζήτησης **l** να ισούται με 0.01. Ο συνολικός αριθμός υπολογισμών τιμών της συνάρτησης fi ήταν 20.

- <u>Μέθοδος του χρυσού τομέα</u>

Εκτελεί 14 υπολογισμούς τιμών της συνάρτησης fi επιστρέφοντας κατάλληλα διαστήματα, που περιλαμβάνουν το σημείο ελαχίστου και ικανοποιούν την απαίτηση **l**=0.01.

- <u>Μέθοδος Fibonacci</u>

Εκτελεί συνολικά 13 υπολογισμούς τιμών της fi και επιστρέφει διαστήματα που ικανοποιούν τη συνθήκη για το εύρος του τελικού διαστήματος αναζήτησης.

- Μέθοδος διχοτόμου με χρήση παραγώγων

Υπολογίζει συνολικά 9 τιμές της παραγώγου. Στη συνέχεια, επιστρέφει είτε σημεία όπου η παράγωγος μηδενίζεται είτε διαστήματα αναζήτησης που πληρούν την επιθυμητή ακρίβεια.

Τελικά, από τον πίνακα παρατηρούμε ότι και οι τρείς συναρτήσεις απαιτούν τον ίδιο αριθμό βημάτων, όταν εφαρμόζεται ο ίδιος αλγόριθμος. Ο πιο αποδοτικός αλγόριθμος αποδεικνύεται πως είναι η μέθοδος της διχοτόμου με τη χρήση παραγώγων, καθώς επιτυγχάνει την διαδικασία βελτιστοποίησης με τον μικρότερο αριθμό υπολογισμών. Έπειτα, ακολουθεί η μέθοδος Fibonacci, μετά η μέθοδος του χρυσού τομέα και τελευταία η μέθοδο της διχοτόμου χωρίς τη χρήση παραγώγων. Έτσι, επιβεβαιώνεται και η θεωρητική ανάλυση σχετικά με την αποδοτικότητα των αλγορίθμων του βιβλίου.

Είναι σημαντικό να αναφερθεί ότι στη μέθοδο της διχοτόμου με χρήση παραγώγων, η ανάγκη υπολογισμού της παραγώγου της αντικειμενικής συνάρτησης, κάτι που δεν είναι πάντα εφικτό. Επιπλέον, οι μέθοδοι Fibonacci και Χρυσού Τομέα για μεγάλα n απαιτούν περίπου τον ίδιο αριθμό υπολογισμών της αντικειμενικής συνάρτησης.

Βιβλιογραφία

Η υλοποίηση των αλγορίθμων, καθώς και οι παρατηρήσεις βασίστηκαν στο βιβλίο *«Τεχνικές Βελτιστοποίησης»* του κ. Ροβιθάκη, Εκδόσεις ΤΖΙΟΛΑ.