ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Πληροφορικής

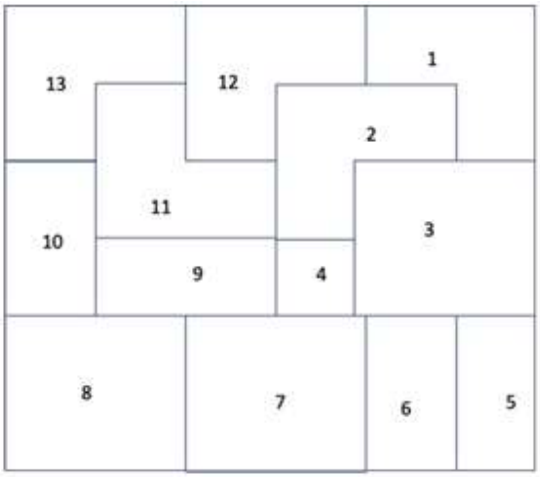
Εργασία Μαθήματος «ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΚΑΙ ΕΜΠΕΙΡΑ ΣΥΣΤΗΜΑΤΑ»

Προαιρετικη εργασια εαρινο εξαμηνο 2024, θεμα Α

Δεσποινιδη Δεσποινα, Π21026

Εκφώνηση της άσκησης

Για φοιτητές με επώνυμο από Α-Κ. Αναπτύξτε πρόγραμμα χρωματισμού του παρακάτω γράφου με χρήση γενετικών αλγορίθμων και γλώσσα προγραμματισμού της επιλογής σας. Τα διαθέσιμα χρώματα είναι 4: μπλε, κόκκινο, πράσινο, κίτρινο.



Χρησιμοποιείστε τυχαίο αρχικό πληθυσμό με πλήθος της δικής σας επιλογής. Χρησιμοποιείστε συνάρτηση καταλληλότητας και διαδικασία επιλογής γονέων σας της δικής σας επιλογής, επίσης. Χρησιμοποιείστε αναπαραγωγή με διασταύρωση ενός σημείου. Επιλέξτε αν θέλετε να κάνετε και μερική ανανέωση πληθυσμού σε κάποιο ποσοστό π.χ. 30% και μετάλλαξη ενός ψηφίου π.χ. στο 10% του πληθυσμού

Περιεχομενα

1. Γενική Περιγραφή της λύσης

Το γραφημα χωριζεται σε 13 περιοχες και εχουμε 4 διαφορετικα χρωματα.καθε διανυσμα, το οποιο εχει μηκος 13 ψηφια από το 1 εως το 4, θα αναπαριστα και έναν διαφορετικο τροπο χρωματισμου του γραφου.το 1 αντιστοιχει στο μπλε, το 2 στο κόκκινο, το 3 στο πράσινο και το 4 στο κίτρινο.Ο γραφος θα πρεπει να χρωματιστει με τετοιο τροπο ώστε οι περιοχες που ακουμπουν η μια την άλλη, οι γειτονικες, να μην χρωματιζονται με το ιδιο χρωμα.

Αρχικα θα πρεπει να οριστει ένας αρχικος πληθυσμος, ο οποιος θα παραγεται τυχαια.

Επειτα,για κάθε ενα διανυσμα μηκους 13 ψηφιων, έναν προγονο, θα πρεπει να υπολογιστει η ποσοτητα καταλληλοτητας του, δηλαδη κατά ποσο ακουλουθει τους κανονες το ότι τα γειτονικα μερη του γραφου δεν εχουν το ιδιο χρωμα.

Αφου γινει η αξιολογηση, θα πρεπει να βρεθει το μεγαλυτερο ποσοστο καταλληλοτητας από τα διανυσματα, και να συγκριθει με ένα οριο καταλληλοτητας που εχει οριστει.Εαν είναι μεγαλυτερο από το οριο, τοτε ο αλγοριθμος τερματιζει και εμφανιζεται το διανυσμα που αντιστοιχει στο ποσοστο καταλληλοτητας που εγινε η συγκριση.Εαν όχι τοτε θα πρεπει να παραχθει μια νεα γενια.

Η παραγωγη της νεας γενιας θα προκυπτει από την προηγουμενη, δηλαδη την τωρινη.Η νεα γενια θα εχει πληθος N, με N να είναι το πληθος της τωρινης.

Στην αρχη επιλεγεται ένα ποσοστο από την τωρινη γενια, το οποιο θα περασει στην επομενη όπως είναι.

Στη συνεχεια, θα πρεπει να επιλεχτουν ζευγαρια,N/2, για να παραχθουν νεα μελη μεσω μιας διαδικασιας διασταυρωσης.

Στο προβλημα η επιλογη γινεται με τη χρηση τροχου, οπου ο τροχος διαθετει τοσες θεσεις, οσες και το αθροισμα των σωστων που εχει κάθε διανυσμα, και κάθε μελος από τη τωρινη γενια παιρνει οσες θεσεις του αντιστοιχουν αναλογα με το ποσοστο καταλληλοτητας του.Για κάθε μελος της τωρινης γενιας, αθροιζονται οσα σωστα εχει κάθε διανυσμα , και επειτα κάθε μελος θα παρει τοσες θεσεις οσες τα σωστα που εχει.Οσα περισσοτερα σωστα εχει , τοσο περισσοτερες θεσεις του τροχου θα πιανει.Ο τροχος θα ‘’γυρισει’’ N φορες, για να επιλεχτουν τα ζευγαρια, δηλαδη η πρωτη τυχαια επιλογη θα διασταυρωθει με τη δευτερη τυχαια επιλογη, η τριτη τυχαια επιλογη μετα τεταρτη τυχαια επιλογη κτλ.

Αφου γινει η επιλογη των ζευγαριων , αυτά θα διασταυρωθουν μεταξυ τους, στο προβλημα θα χρησιμοποιηθει η αναπαραγωγη με διασταυρωση ενός σημειου.Καθε ζευγαρι αναπαραγει 2 απογονους.

Μετα τη διασταυρωση ένα ποσοστο του νεου πληθυσμου μπορει να παθει καποια μεταλλαξη σε ένα τυχαιο ψηφιο, δηλαδη να αλλαξει.Η επιλογη του μελους που θα γινει η μεταλλαξη είναι τυχαια.

Τελος επαναλαμβανονται τα βηματα από την αξιολογηση μεχρι να βρεθει ένα μελος του οποιου το ποσοστο καταλληλοτητας θα είναι ισο ή μεγαλυτερο του οριου.

2. Κώδικας προγράμματος

2.1. Συναρτηση αρχικοποιηση πληθους

```
void arxikopoihsh_plh8ous(list<array<int, 13>> &lista,int k){
    for(int j=0; j<k;j++){
        array<int, 13> pin;
        for(int i = 0; i < 13; i++) {
            int n = 0;
            while(n == 0) {//oste akoma kai an petuxei 0, na 3anampei sthn epanalhpsh
                n = rand() % 5;
            }
            pin[i] = n;
        }
        lista.push_back(pin);
    }
}
```

Η συναρτηση αυτή χρησιμοποιειται μια φορα στην αρχη του προγραμματος, ετσι ώστε να αρχικοποιησει τον αρχικο πληθυσμο.Δεχεται δυο ορισματα, μια λιστα η οποια περιεχει πινακες μεγεθοθς 13 στοιχειων τυπου ακαιρεοι, και έναν ακεραιο αριθμο ο οποιος δινει το μεγεθος του πληθυσμου.

Η συναρτηση εχει 3 δομες επαναληψης. Η μια τρεχει τοσες οσες είναι και ο πληθυσμος. Η δευτερη τρεχει 13 φορες, ώστε για κάθε μελος του πληθυσμου να εχουν κάθε ένα από 13 τυχαια ψηφια από το 1 εως το 4, οπου οι αριθμοι αναπαριστουν τα χρωματα και οι θεσεις στον πινακα τις περιοχες του γραφου.Με την τριτη επαναληψη επιτυγχανεται η τυχαιοτητα.

2.2. Συναρτηση υπολογισμου καταλληλοτητας

```
//kanones
/*
0!=1,2,11
1!=2,3,10,11
2!=3,4,5,6
3!=6,8
4!=5
5!=6
6!=7,8
7!=8,9
8!=9,10
9!=10,12
10!=11,12
11!=12
12!=
*/
```

Για την συναρτηση υπολογισμου καταλληλοτητας, θα πρεπει να εφαρμοστουν οι κανονες που θα πρεπει να ισχυουν ετσι ώστε ένα μελος να θεωρειται εαν είναι καταλληλο και ποσο καταλληλο ειναι , με βαση το πως είναι χρωματισμενα τα σημεια του γραφου.Ετσι παρατηρωντας τον γραφο, διακρινονται τα μερη που ακουμπανε μεταξυ τους.Αυτα τα μερη δεν πρεπει να εχουν το ιδιο χρωμα.Αφου για κάθε περιοχη βρεθουν ολες οι περιοχες που το ακουμπανε, διαγραφονται τα διπλοτυπα ετσι ώστε να μην μετρηθει ένα λαθος δυο φορες.Δηλαδη μη μετρηθει δυο φορες το λαθος ότι η περιοχη 1 ακουμπαει την περιοχη 2 και μετα ότι η περιοχη 2 ακουμπαει την περιοχη 1.

Η συναρτηση λοιπον μετραει για κάθε μελος ποσα λαθη εχει, και επειτα τα διαιρει δια 26(τα συνολικα λαθη που μπορει να γινουν), και ετσι βγαινει σαν αποτελεσμα το ποσοστο καταλληλοτητας.Τα ποσοστα αυτά τα βαζει σε μια λιστα που δεχεται δεκαδικους αριθμους.Οι θεσεις των λιστων που εχουν τα ποσοστα και τα μελη είναι αναλογες.

Επιπλεον η συναρτηση βρισκει το μεγιστο ποσοστο καταλληλοτητας και κραταει τη θεση του, και αποθηκευσει σε άλλη λιστα τα ποσα λαθη εχει κάθε μελος.

2.3. Συναρτηση αρχικοποιηση τροχου

```
void arxikopoihsh_troxou(list<float> pososta,list<int> &troxos,double sum1,list<int> la8h){
    int j=0;
    for (auto& p : pososta) {
    p =p / sum1;//upologizei gia to ka8ena poso pososto tou troxou 8a pianei.auto ginetai an
    cout <<"gia to "<<j<<" "<< p <<"% tou troxou"<< endl;
    j++;
    }
    int i = 0;
    int metrhths = 0;
    for (const auto& l : la8h) {
        int count = 26-l;
        for (int k = 0; k < count; ++k) {
                troxos.push_back(metrhths);
                i++;
        }
        metrhths++;
    }
}
```

Η συναρτηση αρχικοποιηση τροχου δεχεται ως ορισματα την λιστα που κραταει τα ποσοστα καταλληλοτητας κάθε διανυσματος,μια λιστα τυπου int οπου εκει θα λειτουργησει όπως ένας τροχος ,μια double μεταβλητη που κραταει το συνολικο αθροισμα ολων των ποσοστων καταλληλοτητας κάθε διανυσματος και μια λιστα που κραταει ποσα λαθη εχει κάθε διανυσμα.Η συναρτηση λειτουργει όπως έναν τροχο, οπου κάθε διανυσμα θα πιανει ένα ποσοστο του τροχου, αναλογα με το ποσοστο καταλληλοτητας του.Ετσι οσο μεγαλυτερο είναι το ποσοστο καταλληλοτητας ενός διανυσματος, τοσο περισσοτερες θεσεις θα εχει μεσα στον τροχο.

2.4. Συναρτηση μεταλλαξης

```
void metala3h(list<array<int,13>> &lista,float k){
    cout<<endl;
    int pl=lista.size();
    int n;
    int j;
    int per=((((k*pl)/100)+0.5)/1);//ypologizei poses metala3eis 8a ginoun analoga me to pososto kai to plh8os
    cout<<"metala3eis: "<<per<<endl;
    while(per!=0){
        n=rand()%lista.size();
        j=rand()%13;
        cout<<"to "<<n<<endl;
        auto it = next(lista.begin(),n);
        for (int elem : *it) {//emfanizei to prin ths metalla3hs
            cout << elem << " ";
        }
        while(true){//gia na e3asfalisoume oti h metala3h 8a ferei enan diaforetiko ari8mo apo auto pou eixe kai oti autos o ari8mos de 8a einai 0
            int m=rand() % 5;
            if((*it)[j]!=m & m!=0){
                (*it)[j] =m;
                break;
            }
        }
        cout << "metala3h: "<<endl;
        for (int elem : *it) {//kai to meta
            cout << elem << " ";
        }
        cout << endl;
        per-=1;
    }
}
```

Η μεταλλαξη σε έναν γενετικο αλγοριθμο εφαρμοζεται σε ένα ποσοστο του πληθυσμου και αλλαζει το γενετικο τους υλικο,αλλαζοντας ένα ψηφιου αυτου τυχαια.Ετσι και η συναρτηση μεταλλαξης εκτελει αυτή τη λειτουργια, εχοντας ως ορισματα μια λιστα που περιεχει τον νέο πληθυσμο, και μια float μεταβλητη η οποια περιεχει το ποσοστο του πληθυσμου που θα γιενει η μεταλλαξη.

2.5. Συναρτηση μερικης ανανεωσης πληθυσμου

```
void merikh_ananeosh(list<array<int,13>> lista, list<array<int,13>> &lista2,float k){
    int pl=lista.size();
    int n;
    int per=((((k*pl)/100)+0.5)/1);//ypologizei posoi progonoi 8a perasoun apeu8eias sthn epomenh genia analoga me to pososto kai to plh8os
    cout<<"merikh ananeosh: "<<per<<endl;
    list<int> tuxh;//periexei tous ari8mous ton akolou8ion pou exoun klhro8ei
    while(per!=0){
        bool flag=true;
        n=rand()%lista.size();
        for (auto& t : tuxh) {//elegxoume oste o tuxaios ari8mos pou parax8hke na mhn uparxei hdh
            if(t==n){
                flag=false;
            }
        }
        if(flag){
            cout<<"to "<<n<<endl;
            auto it = next(lista.begin(),n);
            for (int elem : *it) {
                cout << elem << " ";
            }
            lista2.push_back(*it);
            cout << endl;
            per-=1;
            tuxh.push_back(n);
        }
    }
}
```

Σε έναν γεννητικο αλγοριθμο,ένα ποσοστο του αρχικου πληθυσμου περναει όπως είναι στην επομενη γενια.Ετσι και η συναρτηση μερικης ανανεωσης , δεχεται ως ορισματα τη λιστα που είναι ο αρχικος πληθυσμος, τη λιστα οπου θα μπει ο νεος πληθυσμος και έναν float k οπου περιεχει το ποσοστο πληθυσμου που θα γιενει μερικη ανανεωση.

2.6. Συναρτηση main

Στη συναρτηση main τρεχει επαναλληπτικη διαδικασια καλωντας τις συναρτησεις (2.1 εως 2.5) εως οτου βρεθει ένα διανυσμα οπου το ποσοστο καταλληλοτητας του θα είναι πανω από το οριο που εχει οριστει, το 0.88, ή εως οτου παραξει 3 γενιες,για να αποφεχθει ο αλγοριθμος να κανει πολλες επαναληψεις ή ακομα και απειρες.

3. Screenshots

```
c:\MinGW\bin>genetikoi3
oi apogonoi
1 4 3 1 2 1 4 4 4 4 2 3 2
2 4 2 1 1 3 2 4 4 2 2 4 3
4 4 4 4 1 3 4 2 3 1 3 4 1
3 4 4 3 3 4 2 2 4 2 1 3 2
2 2 2 1 1 1 4 2 4 3 3 4 3
2 4 3 1 3 2 2 3 1 2 3 1 2
2 4 4 4 2 4 3 3 3 2 3 4 4
3 4 4 1 2 2 3 3 2 3 4 3 4
3 3 2 3 2 3 1 2 3 3 2 2 1
3 3 1 4 1 1 3 2 1 1 1 1 2
2 3 4 2 3 2 4 3 2 2 4 1 1
4 1 1 2 3 3 1 4 2 1 3 1 3
1 2 4 1 1 4 3 4 3 2 1 1 1
3 3 3 4 2 1 2 1 2 2 1 4 4
2 2 1 1 2 2 1 1 1 1 4 3 2
gia to 0. 0.77 % epituxia
6 sfalmata
gia to 1. 0.81 % epituxia
5 sfalmata
gia to 2. 0.58 % epituxia
11 sfalmata
gia to 3. 0.77 % epituxia
6 sfalmata
gia to 4. 0.69 % epituxia
8 sfalmata
gia to 5. 0.85 % epituxia
4 sfalmata
gia to 6. 0.62 % epituxia
10 sfalmata
gia to 7. 0.73 % epituxia
7 sfalmata
gia to 8. 0.77 % epituxia
6 sfalmata
gia to 9. 0.69 % epituxia
8 sfalmata
gia to 10. 0.85 % epituxia
4 sfalmata
gia to 11. 0.77 % epituxia
```

Στην αρχη φαινονται οι απογονοι, τα ποσοστα καταλληλοτητας για το κάθε διανυσμα και το πληθος των σφαλματων.Ελεγχει αν καποιο από αυτά είναι ισο ή μεγαλυτερο του οριου, το 0.88,ώστε να τερματισει ο αλγοριθμος.

```
gia to 0 0.07% tou troxou
gia to 1 0.07% tou troxou
gia to 2 0.05% tou troxou
gia to 3 0.07% tou troxou
gia to 4 0.06% tou troxou
gia to 5 0.08% tou troxou
gia to 6 0.06% tou troxou
gia to 7 0.07% tou troxou
gia to 8 0.07% tou troxou
gia to 9 0.06% tou troxou
gia to 10 0.08% tou troxou
gia to 11 0.07% tou troxou
gia to 12 0.07% tou troxou
gia to 13 0.07% tou troxou
gia to 14 0.05% tou troxou
merikh ananeosh: 5
to 1
2 4 2 1 1 3 2 4 4 2 2 4 3
to 3
3 4 4 3 3 4 2 2 4 2 1 3 2
to 9
3 3 1 4 1 1 3 2 1 1 1 1 2
to 12
1 2 4 1 1 4 3 4 3 2 1 1 1
to 5
2 4 3 1 3 2 2 3 1 2 3 1 2
zeugaria:
to 12 me to 3
to 10 me to 10
to 6 me to 5
to 7 me to 1
to 12 me to 9

metala3eis: 1
to 3
1 2 4 1 1 4 3 4 3 2 1 1 1 metala3h:
1 2 4 1 1 4 1 4 3 2 1 1 1
```

Εφοσον δεν βρεθει καποιο διανυσμα με ποσοστο καταλληλοτητας ισο ή μεγαλυτερο του 0.88, κάθε ένα από τα διανυσματα καταλαμβανουν ένα ποσοστο του τροχου, ώστε να επιλεχθουν με τυχαιο τροπο τα ζευγαρια, και αυτά με μεγαλυτερο ποσοστο καταλληλοτητας καταλαμβανουν και περισσοτερες θεσεις στο τροχο.Εφοσον αρχικοποιηθει ο τροχος, ένα 30% του αρχικου πληθυσμου θα περασει όπως είναι στην επομενη γενια, ο τροπος επιλογης είναι τυχαιως.Επειτα δημιουργουνται τα ζευγαρια και γινεται διασταυρωση αυτων με διασταυρωση ενός σημειου.Στη νεα γενια που εχει δημιουργηθει,γινεται μεταλλαξη σε ένα ποσοστο της 10%, οπου επιλεγεται τυχαια και αλλαζει ένα τυχαια επιλεγμενο ψηφιο.

```
2 4 2 1 1 3 2 4 4 2 2 4 3
3 4 4 3 3 4 2 2 4 2 1 3 2
3 3 1 4 1 1 3 2 1 1 1 1 2
1 2 4 1 1 4 1 4 3 2 1 1 1
2 4 3 1 3 2 2 3 1 2 3 1 2
1 2 4 1 1 4 3 2 4 2 1 3 2
2 3 4 2 3 2 4 3 2 2 4 1 1
2 4 4 4 2 4 3 3 1 2 3 1 2
3 4 4 1 2 2 3 4 4 2 2 4 3
1 2 4 1 1 4 3 2 1 1 1 1 2
gia to 0. 0.81 % epituxia
5 sfalmata
gia to 1. 0.77 % epituxia
6 sfalmata
gia to 2. 0.69 % epituxia
8 sfalmata
gia to 3. 0.77 % epituxia
6 sfalmata
gia to 4. 0.85 % epituxia
4 sfalmata
gia to 5. 0.88 % epituxia
3 sfalmata
gia to 6. 0.85 % epituxia
4 sfalmata
gia to 7. 0.77 % epituxia
6 sfalmata
gia to 8. 0.81 % epituxia
5 sfalmata
gia to 9. 0.73 % epituxia
7 sfalmata
to 5
1 2 4 1 1 4 3 2 4 2 1 3 2
0.88% epituxia
```

Επαναλλαμβανεται η διαδικασια εως οτου βρεθει καποιο διανυσμα που ικανοποιει την συνθηκη.Στη περιπτωση που εχουν δημιουργηθει 3 γενιες, και το ποσοστο καταλληλοτητας του μεγιστου ηταν μικροτερο του 0.88, τοτε θα εμφανιζε αυτό και ο αλγοριθμος θα τερματιζει.