Szavazókörzetek újraosztása

András Réka

June 8, 2017

Chapter 1

Bevezetés

A redistricting kifejezés a választói körzetek határainak újra rajzolását jelenti. Erre a folyamatra időnként szükség van, mert egy terület lakosságának a száma nagy mértékben változhat. Választói szemszögből, ez azt jelenti, hogy a sürűbben lakott körzetek polgárainak szavazata kevesebbet ér, mint a ritkábban lakott kerületekben élő polgároké. Ez ellentmond a demokrácia 'egy ember egy szavat' elvével. A szavazókörzetek újraosztása mégsem az igazságos választásokkal fonódott össze, hanem a gerrymandering fogalmával. A gerrymandering Elbridge Gerry szenátorról kapta a nevét, aki jóváhagyott egy olyan körzet felosztási tervet,amely jelentős előnyhöz jutatta a demokratákat. A különös alakú választókerület egy szalamandrához hasonlított. A Gerry névhez illesztve az angol 'salamander' szót megkapjuk a gerrymandering kifejezést. A gerrymandering vagy választókerület-manipuláció a választói körzeteknek egy olyan felosztását jelenti, amely kedvező/kedvezőtlen helyzetbe hozhat valamely pártot,etnikai,nyelvi,vallási csoportot,stb.. A választási eredmények ilyen manipulációjá kiváltképp a többségi elven müködő demokratikus rendszerekben fordulhat elő. Ezek a rendszerek nem arányosan választják meg képviselőiket, az a képviselő nyer aki a szavazatok többségét kapta, így mindig lesznek kárba veszett szavazatok. Tegyük fel, hogy egy 100 szavazóból álló körzetben, A párt 90 szavazatot kap, B párt 10-et. Ebben az esetben B párt vesztesége 10, A párté 39 (90 minusz a győzelemhez szükséges 51 szavazat). A választókerület-manipulációt végző párt arra törekszik, hogy az ellenfél kárba veszett szavazatait maximalizálja, míg az ő esetében minimalizálja a veszteséget. Annak érdekében, hogy az újraosztás folyamata igazságos legyen, az érintett országok különböző törvényeket hoztak egy felosztás elfogadására vonatkozólag. Legtöbb államban független megbízottak végzik az újraosztást, de ha automatizálni lehetne az egész folyamatot, biztosabb lenne, hogy a választópolgárok jogai nem sérülnek.

Chapter 2

GIS, QGIS, Python

A GIS(Geographic Information System) egy számítógépes rendszer, amelyet földrajzi helyhez kapcsolódó adatok megjelenítésére, elemzésére és tárolására dolgoztak ki. Az elképzelés a következő:

1. földrajzi koordinátákkal rendelkező adatok gyűjtése
2. adatok modellezése
3. adatok elemzése
4. eredmények megjelenítése térképen

2.1 Adatgyűjtés

A GPS-el felszerelt telefonok, autók folyamatosan képesek térbeli adatokat szolgáltatni,és a szatelitek már mindenki számára elérhető felvételeket készítenek a Föld felszínéről.

2.2 Adatmodellezés

A GIS kétféle modellt használ a térbeli adatok tárolására. Az egyik a raszteres(grid) adatmodell. Ez egy pixelekből felépülő adatmodell, ahol minden pixelhez azonos méretü területi egységet rendelünk. A pixelek sorokban és oszlopokban helyezkednek el, ez adja az adatok geometria jellemzőit, ugyanakkor minden mezőhöz értékek sorát rendelhetjük, mindegyiket megfeleltetve egy-egy tulajdonságnak. Gyakori raszter adat fájlformátumok: JPEG2000, IMG, GeoTIFF. A második adatmodell a vektoros adatmodell, ami pontokat (x,y koordináták) tartalmaz, és arra vonatkozó szabályokat, hogy a pontok miként alkotnak vonalakat és poligonokat (zárt vonalak). Ezen adatok tárolása gyakran shapefájlokban történik. Ez igazából legkevesebb három másik különböző formátumú fájlt foglal magába. A DBF (dBase file) egy adatbázis tábla, ez tartalmazza az adataink attribútumait. Az shp kiterjesztésű fájl tartalmazza a földrajzi adatokat és ehhez még tartozik egy index fájl.

2.3 Vetítés

Mivel a Föld pontjainak térképre való vetítésének több módja is van, a vetítésre vonatkozó információk tárolása is szükséges, amennyiben a térképeket egymásra akarjuk illeszteni. A használt vetítés tárolása a projection (.prj) fájlokban történik. A térképvetítés matematikai transzformációk sorozata, amely során egy szferikus felület felszínén található pontok síkba vetitődnek. Először a Föld szabálytalan felszínét egy

szabályos geometria alakzatként kell kezelnünk, mint a gömb vagy szferoid, aztán az így kapott felületet vetíthetjük síkba. Ez a folyamat mindig valamilyen torzúlást eredményez, ezért van olyan sok féle vetítés. Van amelyik a pontok közti távolságot őrzi meg, mások a levetített rész területét, formáját tartják meg. Nem létezik egy egységes vetítési eljárás. Vannak elterjedtebbek,mint például a Google Maps és a Mango Map által használt Web Mercator vetítés.

2.4 Adatelemzés

A QGIS számos előre beépített statisztikai és geometria eszközt nyújt az adatok elemzésére. Például: interpolálás, heatmappek generálása, GPS adatok betöltése, SQL lekérdezések végrehajtása (számolhatunk átlagokat, előfordulásokat, összegzéseket) , convex hurkok meghatározása, osztályozás, kontúrok meghúzása. Mindezek mellett lehetőséget nyújt az adatok elemzésére általunk írt python vagy c++ kóddal. Amikor betöltünk egy layert a QGIS felépíti az annak megfelelő objektumot, amely a QgsInterface osztály iface nevű instanciáján keresztűl érhetünk el. Az így kapott layer objektumtól kérthetünk egy iterátort a featurekhez. A featurek tartalmazzák a geometria információkat (koordináták,terület,stb..) és az adatbázishoz tartozó attribútumok értékeit. Ha a scriptet külső forrásból futtatjuk, importolnunk kell a qgis.core nevű könyvtárat, ez tartalmazza az alap GIS funkcionalitásokat. Pythonban írhatunk pluginokat a QGIS-hez, vagy készíthetünk egy különálló applikációt

2.5 Felhasználás

Az adatok reprezentálásának és elemzésének a GIS által biztosított módját sok területen használják: az epidemiológusok a vírusok globális terjedését vizsgálják, a kriminológusok bűnözési mintákat keresnek, a geológusok a Föld fizikai elváltozásait megfigyelve előre jelzik a földmozgásokat,vagy lehetséges fosszilis erőforrásokat keresnek, a globális felmelegedéssel kapcsolatos kutatásokban is használnak földrajzi információs rendszereket.

Chapter 3

Algoritmus

3.1 A szimulált hűtés módszere

A szimulálát hűtés módszere egy metaheurisztikus algoritmus, amelyet kombinatorikus optimalizációs problémák megoldására használnak. A kombinatorikus optimalizálás, olyan optimalizációs problémákat old meg, ahol a megoldástér véges. A szimulált hűtésnek több változata is van: Párhuzamos Szimulált Hűtés, Gyors Szimulált Hűtés, Adaptív Szimulált Hűtés. A továbbiakban a Klasszikus Szimulált Hűtés módszerét tárgyaljuk, amelyet Kirkpatrick, Gelatt, és Vecchi vezetett be 1983-ban [2] , optimalizációs problémák megoldására. A cikkben leírják a termodinamikai rendszerek viselkedését, majd megmutatják hogyan használható az, mint optimalizációs technika. Egy termodinamikai rendszert különböző változók összessége ír le (hőmérséklet,energia,nyomás,entrópia,stb..), míg egy metaheurisztikus algoritmusnak csupán két eszközre van szüksége (költségfüggvény, szomszédos megoldást generáló függvény). A megfeleltetés a következő: az optimalizációhoz használt költségfüggvény a rendszer energiaszintjének felel meg. Amikor a rendszer eléri az egyensúlyi állapotát az energia minimális. A cél a rendszer egyensúlyi állapotának az elérése, vagyis a költségfüggvény minimumának a megtalálása. Egy szomszédos állapotot a jelenlegi állapoton eszközölt random perturbációk által kapunk meg, ez a rendszeren belüli atommozgásoknak feleltethető meg. Bevezetünk egy hőmérséklet változót a termodinamikai rendszerek mintájára, amely szabályozza a keresésünk viselkedését. Ahhoz, hogy megtaláljuk egy anyag egyensúlyi állapotát nem elégendő ha lehűtjük. Az analógiát követve, a keresés alacsony hőmersékleten azt jelenti, hogy mindig csak a jobb megoldásokat fogadjuk el, gyakorlatilag lokális minimumot keresünk. Ha stabil kristályokat akarunk kapni, először forráspont fölé kell melegíteni az anyagot, majd lassan hűteni, elegendő időt hagyva a molekuláknak, hogy kristályokat formáljanka. Nagy hőmérsékleten a molekulák kaotikus mozgást végeznek, ennek megfelelően az algoritmus is nagy ugrásokat tesz a megoldás térben, így elkerülhető az, hogy lokális minimumba ragadunk. Azt hogy a keresés mekkora valószínűséggel lép a jelenleginél rosszabb állapotba a rendszer aktuális hőmérséklete befolyásolja. Az iterációk során akkor fogadunk el egy új megoldást, ha az jobbnak bizonyúl, mint a jelenlegi megoldás. Amennyiben nem alacsonyabb az új megoldásunk költségfüggvényének az értéke, akkor a jelenlegi megoldást a Metropolis kritérium alapján,$\exp((E - E_0)/T)$ valószínűséggel cseréljük le. Ahol E az aktuális megoldás objektív értéké, E_0 a szomszédos megoldás objektív értéke, T a rendszer hőmérséklete.

3.2 Több-célfüggvényű algoritmus újraosztáshoz

A több-célfüggvényű algoritmus különböző kritériumokat figyelembe véve optimalizál. A mi célunk olyan (K,U) párosításokat találni, amelyek legnagyobb mértékben megfelelnek az általunk definiált objektív függvényeknek. K a körzeteket jelöli, U a unitokat. A unitok a létező legkisebb közigazgatási egységei

a felosztandó országnak. Az ország egy lehetséges felosztása, vagyis egy megoldás, olyan (K,U) párok összessége, ahol minden körzet előre meghatározott számú unitot foglal magába , és egy unit csak egy körzethet tartozhat. Amikor összehasonlítunk két megoldást, ahol V az aktuális megoldás, V' a szomszédos megoldás, három lehetséges eset van:

1. V' minden objektív függvény szerint jobb megoldásnak bizonyúl
2. V' csak bizonyos objektív függvényekre mutat jobb eredményt
3. V' minden objektív függvény szerint rosszabb megoldásnak bizonyúl

A megoldásainkat egy olyan halmazba gyűjtjük, amely a fenti eseteknek megfelelő szabályok szerint bövűl. Ha V' dominál más megoldásokat (1 eset), átveszi a leggyengébb dominált megoldás helyét. Ha az új megoldás nem dominál (nem igaz az 1. eset) és nem dominált (nem igaz a 3. eset), akkor a halmazhoz adódik. Ha V' dominált valamely megoldás által nem kerül a halmazba, viszont a szimulált hűtésnél leírt valószínűség szerint felválthatja a jelenlegi megoldást. Amikor egy új megoldás kerül a halmazba, mindig hozza rendelődik egy random választott súlyvektor az előre meghatározott súlyok listájából. Az algoritmus végére minden súlyvektorhoz fog legalább egy nem dominált megoldás tartozni. Egy lehetséges algoritmus annak eldöntésére, hogy egy megoldás dominált-e más megoldások által, illetve dominál-e más megoldásokat, k darab célfüggvény esetén:

Algorithm 3.1 Dominálás

Require: $k \geq 1$
dominalt ←false
dominaltak ←∅
for $s \in \mathcal{P}areto$ **do**
    UdominaltSaltal ←true
    SdominaltUaltal ←true
    for $i = 1, .., k$ **do**
        UdominaltSaltal ←UdominaltSaltal and s.objektiv(i)<u.objektiv(i)
        SdominaltUaltal ←SdominaltUaltal and u.objektiv(i)<s.objektiv(i)
    end for
    dominalt ←dominalt or UdominaltSaltal
    if SdominaltUaltal **then**
        $dominaltak \leftarrow dominaltak \cup \{s\}$
    end if
end for

A szimulált hűtés több célfüggvény esetén:

Algorithm 3.2 Szimulát hűtés több célfüggvénnyel

```
1: legyen U a kezdeti megoldas
2: legyen T a kezdeti kezdeti homerseklet
3: pareto ← ∅
4: pareto ← pareto ∪ {U}
5: while T ≠ fagyaspont do
6:     for i = 1, .., maxiteracio do
7:         V ← szomszed(U)
8:         if V dominans then
9:             pareto.update(V)
10:            U ←V
11:        else
12:            if V nem dominalt then
13:                pareto ← pareto ∪ {V}
14:                U ←V
15:            else
16:                if random(0,1) <= exp(obj(U)-obj(V)/T) then
17:                    U ←V
18:                end if
19:            end if
20:        end if
21:    end for
22:    T ← T ∗ hutes
23: end while
```

Chapter 4

Redistricting

4.1 Általánosan

A megírt program a szimulált hűtés módszerével generálja egy ország megyéinek/államainak/tartományainak lehetséges körzetekre való felosztását. A célfüggvények és a hozzájuk tartozó súlyok tetszőlegesen megadhatóak. A pythonban írt plug-in a QGIS-ből futtatható. A programnak szüksége van egy shapefájlra, amely tartalmazza a felosztandó megyéket, és egy másik fájlra amely a felosztandó ország közigazgatási egységeit (unit) tartalmazza. A két fájlhoz tartozó adatbázis tábláknak összekapcsolhatóknak kell lenniük a megye azonosítók szerint, tehát tudnunk kell, hogy egy közigazgatási egység melyik megyéhez tartozik. A későbbiekben egy unit szomszédainak a bejárására gyakran kerül sor, ezért a shapefájlok első használata alkalmával egy scriptet is le kell futtatni, amely bővíti egy attribútummal a unitokat tartalmazó adatbázis táblát. Az új attribútum az adott unit szomszédos unitjainak az azonosítóját tárolja. Ez később rengeteg futási időt nyer majd a körzetek változtatásához szükséges műveletek során. A plug-in könyvtár-struktúrája tartalmaz egy config.yaml elnevezésű fájlt, ezt változtatva állíthatjuk be a paramétereinket. Elsősorban megkell adni a shapefájlainkban szereplő attribútumok elnevezéseit, ezek helyes ismerete nélkül a program nem lesz futtatható. Megkell mondanunk összesen hány körzetet akarunk látni a végső felosztásban. A további paraméterek az optimalizáló algoritmus viselkedését befolyásolják, ezek változtatása nélkűl is használható a program. A szimulált hűtés módszerének olyan paramétereit változtathatjuk, mint a kezdeti és végső hőmérséklet, a hűtés mértéke és a hőmérsékletenkénti iteráció. Beállítható a megoldás halmaz maximális mérete és a költségfüggvényekhez rendelhető lehetséges súlyok listája. Az is megmondható, hogy egy körzethez minimum hány unitnak kell tartoznia. A használt költségfüggvényeket az objectives.py fájl tartalmazza, ezek bővítése értelemszerűen nem adható meg a konfigurációs fájlban. Amennyiben újabb függvényt szeretnénk egy megoldás kiértékeléséhez adni azt implementálnunk kell az objectives.py fájlban. A program indítás után iterációnként loggolni fogja a megoldáshalmaz viselkedését. A hőmérséklet lehűlése után a legjobb megoldás megjelenik a QGIS grafikus felületén, úgy hogy az ugyanazon körzetekhez tartozó unitok azonos színűek lesznek.

4.2 Objektív Függvények

A program alapértelmezetten két függvényt használ a megoldások kiértékelésénél. Ezek a függvények a gerrymandering és a malapportionment elkerülését szolgálják. A C_1 függvény azt méri, hogy a populáció mennyire oszlik el egyenletesen a körzetek között. Minnél alacsonyabb a függvény értéke, annál egyenletesebb az eloszlás. A használt függvényt az IFE (Instituto Federal Electoral) határozta meg, és használta a 2006-os mexicói újra osztások során:

$$C_1(P) = \sum_{s \epsilon S} (\frac{P_E}{d(P_N/300)})^2 (\frac{P_S}{P_E} - \frac{1}{n})^2$$

Ahol$P = \{Z_1, Z_2, ..., Z_n\}$ egy felosztása az országnak, úgy hogy$Z_s = \{U_1, U_2, ..., U_{s_k}\}$.$Z_s$ az s körzet unitjainak a halmaza.P_N az ország ,P_E a megye lakosságának a száma, d a megengedett szórás a megyén belül, n a megyéhez tartozó körzetek száma. A másik objektív függvény a körzetek formájának a kompaktságát méri. Abból kiindulva, hogy a legkompaktabb forma a kör, mérjük a körzet kompaktságát. Legyen a körzethez tartozó unitok által formált alakzat kerülete P_s , területe A_s. Minnél kompaktabb a körzetünk, a kerülete annál inkább megegyezik egy A_s területű kör kerületével:

$$C_2(P) = \sum_{s \epsilon S} (\frac{P_s}{4\sqrt{A_s}} - 1)$$

4.3 Románia

Chapter 5

Összefoglalás, eredmények

Bibliography

[1] E. A. Ringcón-García, M. A. Guiterrez-Andrade et al., A Multiobjective Algorithm for Redistricting, Journal of Applied Research and Technology Volume 11, Issue 3, June 2013, Pages 324–330.

[2] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi , Optimization by Simulated Annealing , Science, New Series, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.