

Szavazókörzetek újraosztása

Szerző:

András Réka

(Babeş-Bolyai Tudományegyetem,
Kolozsvár, Informatikus szak, 3. év)

Témavezető:

dr. Szöllősi István

(adjunktus, Babeş-Bolyai
Tudományegyetem, Kolozsvár,
Matematika és Informatika Kar)

MTDK 2018

Temesvár, 2018. április 26-28.

Kivonat

A demokratikus államformák fő célja a választópolgárok érdekeinek a képviselete kellene legyen, de a valóságban azt tapasztaljuk, hogy az egyéni ambíciók gyakran előtérbe helyeződnek a politikum részéről. Azonban úgy gondolom – ahogyan Esterházy Péter is írta: „a legrosszabb demokrácia is jobb, mint az, ami eddig volt” – és akkor ennek a kevésbé rossznak a jobbá tételére kell törekednünk. Ennek a törekvésnek a kinyilatkoztatása a szakdolgozatban bemutatásra kerülő QGIS modul, amelyet a szavazókörzetek objektív és átlátható újraosztásának céljából írtunk.

1. fejezet

Bevezetés

A modern democráciák működésének elengedhetetlen alapfeltétele (többek között) egy átlátható választási rendszer. Egy pártatlan és átlátható választási folyamat hiánya kétségbe vonhatja a közigazgatási rendszer (kormány, intézmények, stb.) legitimitását. Sajnos mindig vannak olyan folyamatok, amelyek azt az érzetet ébresztik a szavazóban, hogy a döntések tőle függetlenül, zárt ajtók mögött zajlanak. Ezen folyamatok közé tartozik a szavazókörzetek újraosztása. Jobb esetben az újraosztási tervet egy függetlennek nevezett bizottság készíti el, amelyet majd valamely törvényhozó szerv elfogad. Az emberek jogosan sejtenek rejtett szándékokat a gyakran földrajzi szempontból irracionálisnak tetsző határvonalak mögött: *redistricting*¹ és *gerrymandering*² – ez a két kifejezés elválaszthatatlanná vált. A körzetek újraosztása sokszor elkerülhetetlen, ezért az átlátható és az igazságos választások érdekében a választókerület-manipulációt kell megszüntetni.

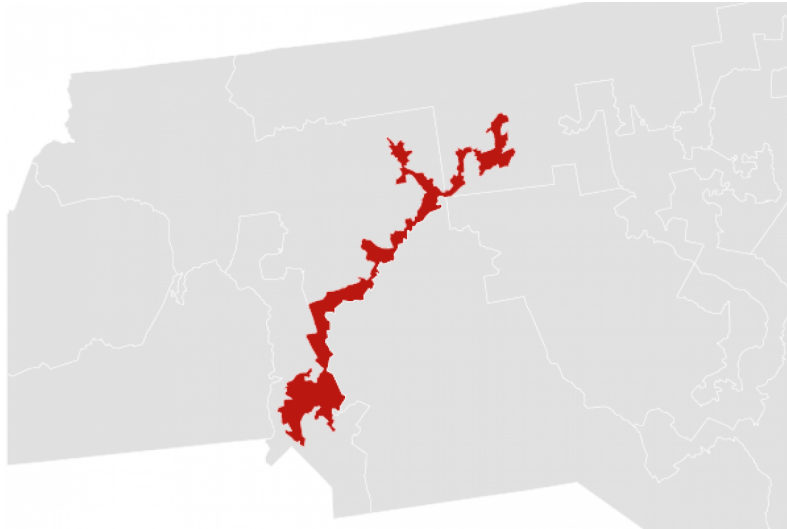
Az általunk készített és a dolgozatban bemutatott QGIS³ kiterjesztés (plug-in) célja az újraosztási folyamat teljes automatizálása, ami mindenki számára követhetővé és átláthatóvá teszi a folyamatot. A felosztást egy többszörös célfüggvényt optimalizáló algoritmus végzi, a szimulált hűtés⁴ módszerét használva. Azáltal, hogy eltűnik a folyamatból az emberi tényező, elkerülhetővé válnak az olyan felosztások, amelyek valamely érdekcsoportnak szándékosan kedveznek. Ugyanakkor az, hogy a program milyen kritériumok alapján végzi a felosztást, mindenki számára áttekinthető, így az újraosztás körüli gyakori bizalmatlanság is megszűnne. Ez remélhetőleg csökkentené a kételyeket a többségi elven működő demokratikus rendszerek legitimitását illetően, és megszüntetné a történelem során oly sokszor előforduló, önkényes „határvonalakat”.

¹A választási körzetek újraosztása

²A választási körzetek újraosztási folyamatának befolyásolása annak érdekében, hogy bizonyos érdekcsoportoknak kedvező választási eredmény szülessen.

³Korábbi nevén: Quantum GIS.

⁴Angolul *simulated annealing*.



1.1. ábra. Észak-Karolina jelenlegi 12. választói körzete

A 2. fejezetben megmagyarázunk néhány, a probléma átlátásához szükséges politikai fogalmat. Majd a 3. fejezet röviden ismerteti az alkalmazás által használt technológiákat: a térinformatikai (GIS) rendszereket általában, majd konkrétan a QGIS térinformatikai szoftvercsomagot ([2, 3, 4, 5, 6] alapján).

A szavazókörzetek felosztását végző algoritmus részletes bemutatása a 4. fejezetben található. A többszörös célfüggvény néhány elengedhetetlen kritériumnak megfelelően van megválasztva, amit szintén ebben a fejezetben tárgyalunk. Végül az 5. fejezetben – a módszer alkalmazásaként – generáljuk Románia megyéinek, majd körzeteinek lehetséges felosztásait.

2. fejezet

Egy kis politika

2.1. Parlamenti választások

A demokratikus rendszerekben a törvényhozás tagjait legtöbb esetben közvetlenül választják meg. A törvényhozó testület lehet egykamarás vagy kétkamarás. A kétkamarás rendszerek gyakoribbak, ezeket a Képviselőház és a Szenátus alkotja. Ez nagy számú képviselő¹ és szenátor² megválasztását jelenti, azonban az országok ennél jóval kevesebb adminisztrációs egységre (megye, régió, tartomány, stb.) vannak felosztva. Közigazgatási szempontból egy ország túlzott „felaprózása” fölösleges bürokráciához vezet. Ugyanakkor, minél nagyobb a parlament tagjainak száma, annál inkább beszélhetünk képviseleti demokráciáról. Tehát a közigazgatási egységek száma nem feleltethető meg a képviselőket és szenátorokat megválasztó körzetek számával.

Ahhoz, hogy parlamenti képviselőket válasszunk meg, az ország már meglévő közigazgatási egységeit kisebb részekre kell osztani. Ezeket a felosztásokat nevezzük választói körzeteknek. A körzeteket az ország legkisebb egységei (falvak, városok, stb.) alkotják. Kétféle típust különböztetünk meg: többmandátumos választókörzet és egyéni (egymandátumos) választókörzet. Az egyéni körzetes választási rendszerek esetén minden választókörzet egyetlen képviselőt delegál. Ebben az esetben a körzetek száma megegyezik a parlamenti mandátumok számával. Ilyen szavazói rendszert alkalmaznak többek között Franciaországban, az Egyesült Királyságban és az Amerikai Egyesült Államokban. A kizárólag egymandátumos választási rendszerekben a választások kimenetele sokkal inkább manipulálható, mint a többmandátumos rendszerek esetében. Ennek okát a későbbiekben részletesebben is megmagyarázzuk. A többmandátumos választókörzeteket az arányos (pártlistás) választási rendszerekben használják. Ez esetben az állampolgárok a pártok jelöltlistáira szavaznak, a pártok a kapott szavazatokkal arányos mértékben jutnak be a parlamentbe. Sok állam a fent említett választási rendszerek valamilyen kombinációját használja, ilyen például Románia és Magyarország.

¹Romániában 329, az Egyesült Államokban 435 képviselőt választanak

²Romániában 136, az Egyesült Államokban 100 szenátort választanak

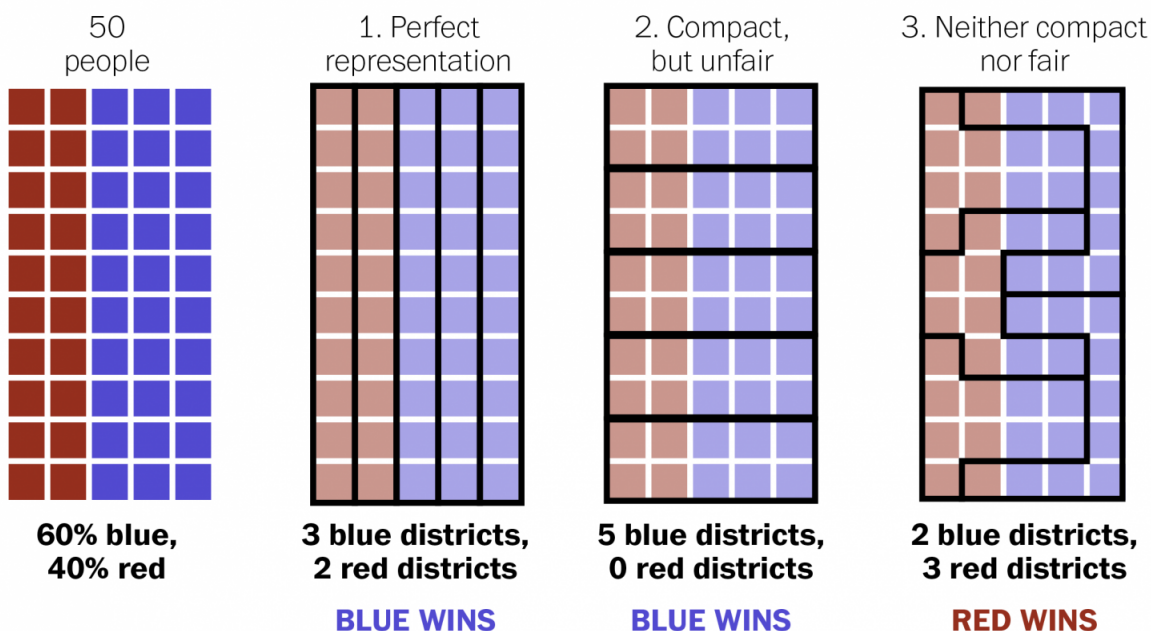
2.2. Redistricting

Különböző demográfiai változások következtében a választási körzetek populációja csökkenhet vagy növekedhet. Ez a választói rendszeren belüli aránytalanságokhoz vezet, ami azt eredményezi, hogy a sűrűbben lakott körzetek polgárainak szavazata kevesebbet ér, mint a gyéribben lakott körzetekben élő polgároké. Ez ellentmond a demokrácia „egy ember egy szavazat” elvével. Ennek elkerülése érdekében a választási körzeteket szükség szerint újraoszthatják. Ezenkívül törvényi követelmény is lehet, hogy a körzeteket újra kell osztani bizonyos időközönként.

Korábban az újraosztást a törvényhozó hatalom végezte el különösebb megkötések nélkül. Az így bekövetkezett folyamatos visszaélések eredményeként az országok elkezdtek törvényeket hozni az újraosztási tervekre vonatkozólag, például: a körzeteknek folytonosnak és kompaktnak kell lenniük, a körzetek között a populációnak egyenletes eloszlást kell követnie, meg kell őrizniük az ország adminisztrációs felbontását, stb. Mindezek ellenére, általában nem az igazságos választási előfeltételek megteremtését társítják a szavazókörzetek újraosztásához. Gyakran merül fel a gerrymandering fogalma az újraosztások után, ugyanis különböző felosztási tervek, különböző választási kimenetekhez vezethetnek.

Gerrymandering, explained

Three different ways to divide 50 people into five districts



WASHINGTONPOST.COM/**WONKBLOG**

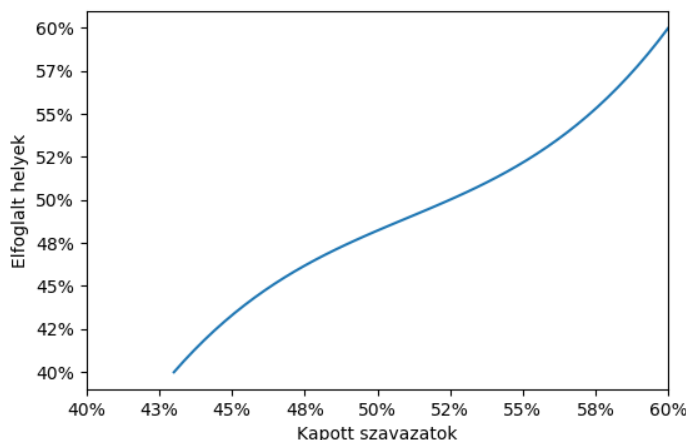
Adapted from Stephen Nass

2.1. ábra. Három különböző kimenetelt eredményező felosztás ([13, 12])

2.3. Gerrymandering

1812-ben Massachusetts kormányzója, Elbridge Gerry, jóváhagyott egy olyan körzetfelosztási tervet, amely jelentős előnyhöz juttatta a demokratákat. A különös alakú választókerület egy szalamandrához hasonlított. A Gerry név és az angol „salamander” szó összetételéből megkapjuk a *gerrymandering* kifejezést, ami a választói körzeteknek egy olyan felosztását jelenti, amely kedvező/kedvezőtlen helyzetbe hozhat valamely pártot, etnikai vagy vallási csoportot. A gerrymandering két fő stratégiája a „cracking” (szétosztás) és a „packing” (összerakás). Mindkét esetben az ellenfél kárba vesztett szavazatainak a maximalizálása a cél. Ez közvetlenül az egymandátumos körzeteket érinti. Az arányos választási rendszerekben minden szavazat beleszámít a végső eredménybe, így nem beszélhetünk kárba ment szavazatokról.

A probléma megértéséhez vegyünk például egy 100 szavazóból álló egymandátumos körzetet, ahol az A párt 90 szavazatot kap, a B pártra pedig 10-en szavaznak. Ebben az esetben a B párt vesztesége 10, a A párté pedig 39 (90 mínusz a győzelemhez szükséges 51 szavazat). Látható, hogy a szavazatveszteség eredménytől független. Akkor is gerrymandering-et végzünk, ha az ellenfél szavazóit egy körzetbe gyűjtjük (packing) és akkor is, ha megosztjuk őket (cracking), úgy hogy a lehető legkevesebb helyen ériék el a szavazati többséget. Így megtörténhet az, hogy egy párt sokkal több parlamenti helyre tesz szert, mint amennyit a választók közötti támogatottsága alapján érdemelne. A gerrymandering mérésének egyik módja a parlamentben elfoglalt helyek és a szavazók közötti támogatottság kapcsolatának az elemzése. Ezen kapcsolat ábrázolására vezették be a hely-szavazat görbét ³ (lásd [1]). Például az A párt esetében a hely-szavazat görbét az S függvény ábrázolásával szemléltetjük. Az S függvény megadja a parlamentben elfoglalt helyek $S(V_A)$ arányát a párt által kapott szavazatok V_A arányának függvényében.



2.2. ábra. Hely-szavazat görbe

³Angolul „seat-vote curve”

Az így kapott görbék összehasonlításával kimutatható a részrehajlás⁴ a felosztási tervben. Egy teljesen igazságos felosztás azt jelentené, hogy $S(V_A) = S(V_B)$ bármely két A és B párt esetén. A hely-szavazat görbe elemzése a választások lebonyolítása után megtehető, de ezzel nem biztosítottunk egy igazságos választást, csupán utólagos bírálatnak tehetjük ki az eredményeket. Annak garantálása, hogy a felosztási tervek teljesen igazságosak, továbbra is megoldatlan feladat. Az érintett államok különböző módszerekkel próbálkoznak. Van ahol egy politikamentes bizottság végzi a felosztást, máshol egy szoftver generálja ezeket, és mindeközben újabb és újabb törvényeket hoznak, amelyeket egy felosztási tervnek be kell tartania.

⁴Angolul „partisan bias”

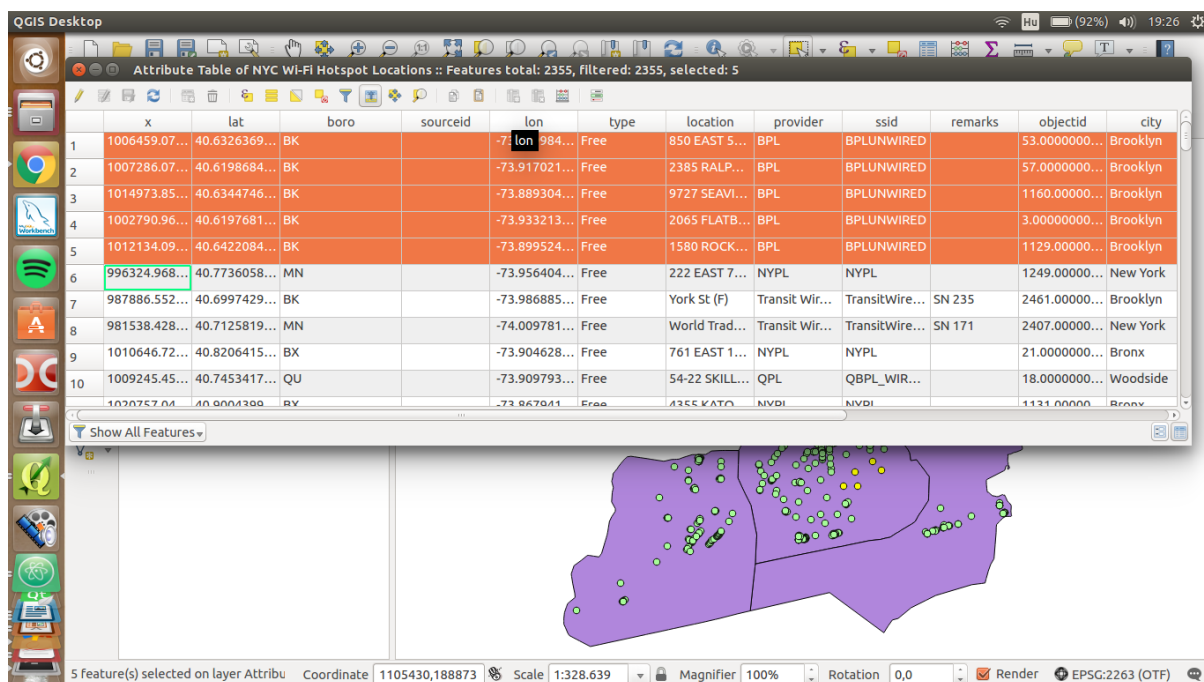
3. fejezet

Térinformatikai rendszerek

A GIS (Geographic Information System) hardver, szoftver és módszerek rendszere, amely segíti a komplex tervezési és irányítási feladatok megoldására szolgáló térbeli adatok gyűjtését, kezelését, feldolgozását, elemzését és megjelenítését. A GIS technológiák könnyedebé teszik a térbeli adatok rendszerezését és térbeli kapcsolataik megértését, azáltal, hogy egy rendszerbe integrálják a térképi és leíró adatokat. Egy térinformatikai rendszerben az adatok elérése földrajzi helyzetük alapján is lehetséges, a digitális térképen megjelölve egy pontot, a szoftver kikeresi a térképhez tartozó adatbázisból az adott ponthoz tartozó adatokat. Ami igazán fontos ezekben a rendszerekben az a térképek közötti átfedések megvalósítása, a térképek úgynevezett „összelinkelése”. Így különösebb nehézség nélkül lehet egymás viszonylatában vizsgálni a különböző forrásból érkező, különböző típusú adatokat.

3.1. Adatmodellezés

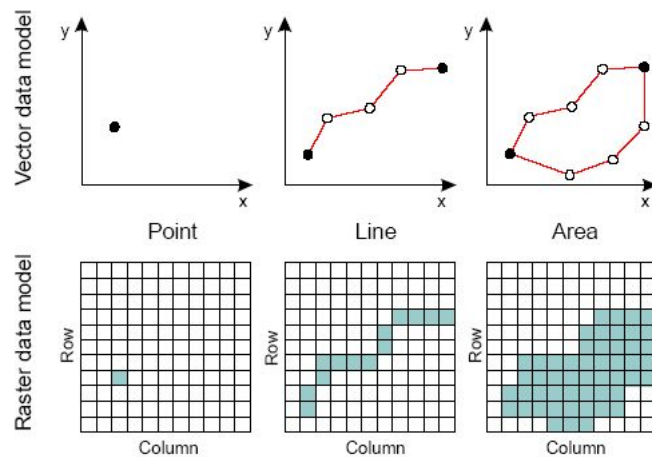
A GIS által feldolgozott adatok két típusát különböztetjük meg: térbeli és leíró adatok. A térbeli jellemzők alapján köthetjük az adatainkat a Föld pontjaihoz (*georeferencing*), láthatjuk a formájukat és az egymáshoz viszonyított helyzetüket, méretüket. A térbeli adatokat úgynevezett *layer*-ekben (rétegek, fóliák) tároljuk. A *layer*-ekben tárolt objektumokhoz kapcsolódik egy attribútumtábla, amely a leíró adatainkat tartalmazza. Minden egyes objektumhoz tartozik egy sor a táblázatban, az oszlopok pedig az egyes elemekhez tartozó további adatokat (attribútumokat) tartalmazzák.



3.1. ábra. A térképen látható sárga pontoknak megfelelő sorok vannak az attribútumtáblában kijelölve

A GIS kétféle modellt használ a valós világ reprezentálására. Az egyik a raszteres (grid) adatmodell. A raszteres adatmodellt azonos méretű cellák (pixelek) alkotják, amelyek egy szabályos rácson helyezkednek el. Minden cella egy értéket tartalmaz, a rácshálózat az egész teret kitölti és a tér minden pontjáról információt hordoz. Raszteres adatnak számít például egy műholdfotó, ahol minden színhez különböző jelentések társulnak vagy egy topográfiai térkép. Legjellemzőbb fájlformátusai: .jpg, .tif (GeoTiff), .img stb.

A második adatmodell a vektoros adatmodell, amely az ábrázolandó területet és a rajta lévő objektumokat pontok és a köztük lévő egyenesek együtteseként fogja fel. A raszteres adatmodellel ellentétben, nem tölti ki a teret. Csupán pontokat tárol koordinátákként, valamint a vonalak vagy poligonok (zárt vonalak) összekötésére vonatkozó információkat. Mindig csak azonos típusú geometriai elemeket tartalmazhat. A vektoros adatok tárolása leggyakrabban shapefájlokban (ESRI shapefile) történik. Egy shapefájl a fájlrendszerben nem egyetlen fájl, hanem rendszerint 3-6, azonos nevű, de eltérő kiterjesztésű fájlrendszer: a .dbf kiterjesztésű fájl (dBase file) a leíró adatokat tartalmazza, az .shp kiterjesztésű az objektumokat tárolja, az .shx fájl a kapcsolatot határozza meg az objektumok és az adattábla összetartozó részei között, a .prj kiterjesztésű fájl pedig a használt vetítési rendszerre vonatkozó információkat tartalmazza. További vektoros fájlformátusok: .dwg és .dxf (AutoCAD fájl), .kml (Google Earth fájlformátum), .gpx (GPS fájlcsere-formátum).



3.2. ábra. Vektoros és raszteres adatok tárolása közti alapvető különbségek ([15])

Összefoglalva azt mondhatjuk, hogy a raszteres adatmodell elsősorban az entitások jelentését vizsgálja (terep, növényzet, épületek, utak, stb.), ezért sokkal inkább alkalmas területi elemzésre és topográfiai térképek ábrázolására. Ezzel szemben a vektoros adatmodell grafika-központú modell, azaz nagyobb geometriai pontossággal tárolja az adatokat. Nagy előnye, hogy a kép tetszőlegesen nagyítható. A két formátum közötti konverzió lehetséges. A vektor-raszter konverzió jól megoldott, viszonylag egyszerű feladat, melynek során a vektoros információk a raszter felbontásának megfelelő méretű képelemmé konvertálódnak, elveszítve eredeti struktúrájukat. A raszter-vektor konverzió lényegesen összetettebb feladat, nehezebb automatizálni, megoldása nagymértékben függ az alkalmazott konverziós algoritmustól és a konverzió paramétereitől.

3.2. Vetületi rendszerek

A térképvetítés olyan matematikai transzformációk sorozata, amelyek során a Föld felszínén található pontok a síkkoordináta-rendszerbe vetítődnek. Bizonyos torzulások, mint szög-, távolság-, vetületi irány-, vonalas aránymérték- és terület-torzulás mindig nemkívánatos eredményei a folyamatnak. Néhány leképezés ezen tulajdonságok valamelyikének torzulását minimálisra csökkenti más, jelentős méretű torzulást okozva valamely más paraméterben. Némely vetület pedig csak mérsékelten igyekszik torzítani ezen tulajdonságok mindegyikét.

A Föld felszíne kitüremkedésekkel és mélyedésekkel teli, ezért ezt a szabálytalan idomot helyettesíteni kell egy matematikailag jól meghatározható, a Föld formáját nagy mértékben megközelítő felülettel. Az így megválasztott felületet nevezik alapfelületnek. A leggyakrabban használt alapfelület a forgási ellipszoid (szferoid). Egy szferoidot leírhatunk a fél-kistengely hossza és a lapultság (gömb formájától való eltérés) által. Mivel az így kapott szferoidok egyike sem modellezheti teljes pontossággal a Földet, különböző referencia-szferoidokat használnak a vetítések során. Amikor csak egy adott terület

ábrázolása a cél, a szferoidnak olyan középpontot választunk, amivel az illeszkedési hiba minimális. Ezt a szakirodalom *geodéziai dátumnak* nevezi. A térbeli modell meghatározása után következik a pontok síkba való vetítése. A vetítés az alapfelületről történik a képfelületre (térkép). Általában nem közvetlenül vetítődnek az alapfelület pontjai a síkra, hanem a vetítés először egy síkba fejthető felületre történik, amelyet aztán „felvágva” a síkba terítenek. A vetületi rendszerek különbözhetnek az alapfelületben, a vetítés módszerében, az egységnyi hossz mértékében, stb. A GIS által használt vetületi rendszereket két csoportba sorolhatjuk:

1. Földrajzi koordináta-rendszer – ezek hosszúsági és szélességi koordinátákat használnak, közöttük találjuk például a WGS84-t (World Geodetic System 1984)¹
2. Vetületi koordináta-rendszer – ezek X, Y Descartes-koordinátákat használnak, közöttük találjuk az UTM-t (Universal Transverse Mercator)

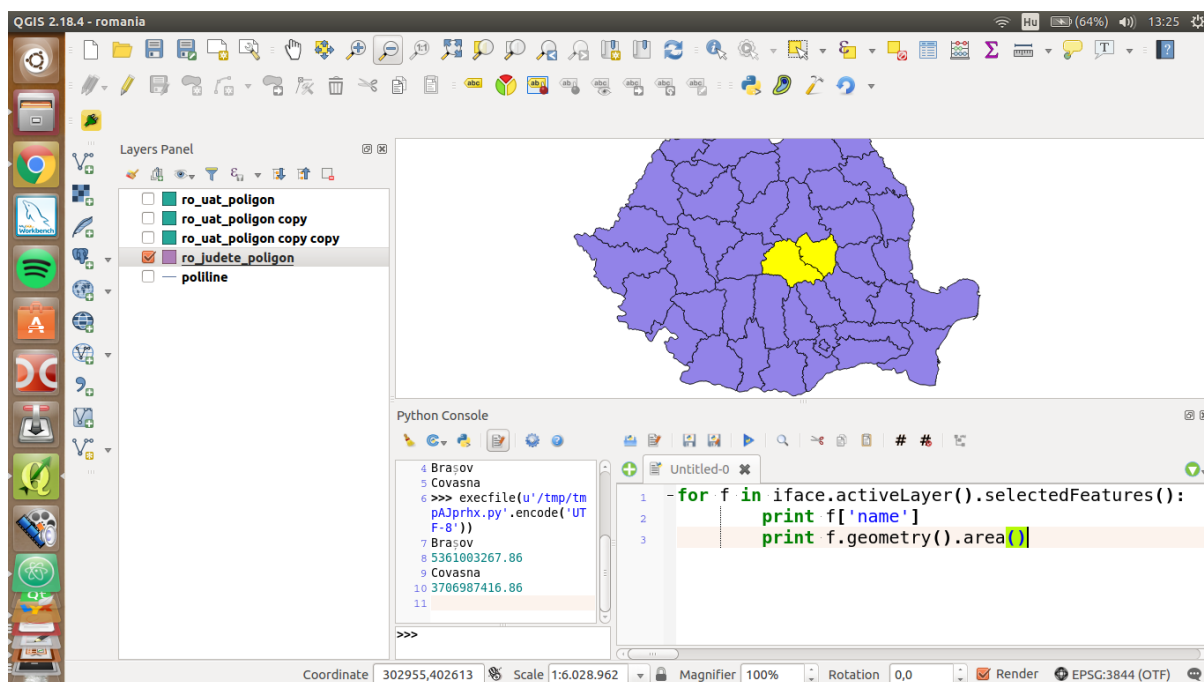
3.3. A QGIS-ről röviden

A QGIS az egyik legnépszerűbb nyílt forráskódú, platformfüggetlen térinformatikai szoftver, melynek fejlesztése még a 2000-es évek elején kezdődött el. Fejlesztése az OSGeo ² hivatalos projektje. A szoftver képességei és használati lehetőségeinek száma folyamatosan növekszik, köszönhetően a beépített alapvető függvények és a beépülő modulok gyarapodó számának. Így a program alkalmas térképek megjelenítésére, szerkesztésére, adatok elemzésére és nyomtatható térképek összeállítására. Támogatja a legtöbb vektoros formátumot, beleértve az ESRI shapefájlt, a Mapinfo, SDTS, és GML (GoogleEarth XML) formátumokat is. A QGIS programban megadhatjuk a layer-ek vetületi rendszerét, ez alapértelmezetten a WGS-84. Különböző rendszerek esetén, azonos vetületi rendszerbe transzformálhatjuk az adatainkat, ugyanakkor lehetőségünk van saját vetületi rendszer definiálására.

A szoftver indítása után az első feladat az objektumokat tartalmazó layer-ek betöltése. Amint betöltünk egy layer-t, az láthatóvá válik a QGIS térképnézegetőjében. A térképen történő navigálást legegyszerűbben a Térkép navigáció eszközsor elemeivel tehetjük meg. Betöltéskor a QGIS felépíti a térképen látható geometriai formáknak megfelelő objektumokat. Az adatokat manipulálhatjuk a menüsor által nyújtott eszközökkel vagy a QGIS által biztosított Python konzolból szkriptek segítségével. A Python kódrészletekkel lehetőségünk nyílik a program funkcionalitásainak a bővítésére.

¹A WGS-84 az egész Földre definiált dátum

²Open Source Geospatial Foundation



3.3. ábra. A konzolban látható a mellette lévő szövegszerkesztőben írt szkript végrehajtásának eredménye

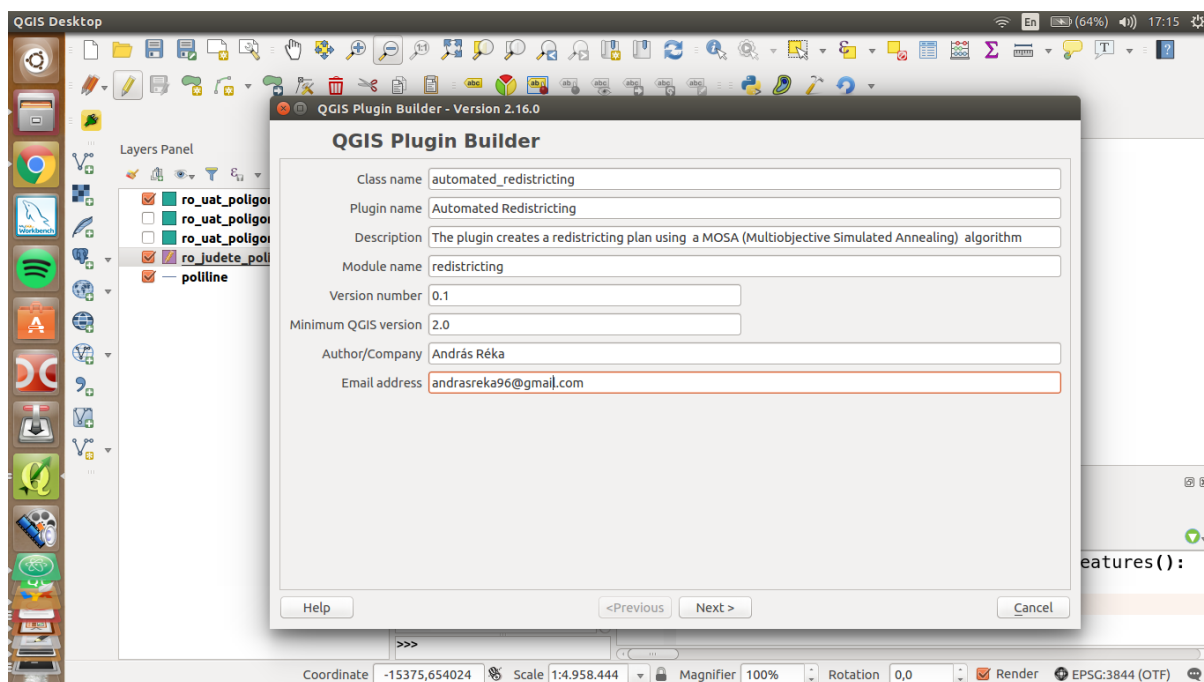
A konzol megnyitása előtt a QGIS az alábbi két parancsot végrehajtja:

```
from qgis.core import *
import qgis.utils
```

A `core` könyvtár tartalmazza az alap GIS funkciókat, a `utils` könyvtár tartalmazza a modulok kezeléséhez szükséges funkciókat. A betöltött layer-ekhez tartozó információkat a `QgsInterface` osztály `iface` nevű példányán keresztül érhetjük el. Az így kapott layer tartalmazza a térképet alkotó objektumokat. Egy objektumot „feature”-nek nevezünk. A QGIS által felépített feature egy olyan szótár (Python dictionary), ahol a kulcsok az attribútumtábla mezőinek nevei, az értékek pedig a mezők értékei. Tehát például az `f` feature *népesség* elnevezésű mezőjének az értékére `f['népesség']` formában hivatkozhatunk. A QGIS a geometriai műveleteket a GEOS³ könyvtár segítségével végzi. Minden feature rendelkezik egy `QgsGeometry` objektummal, ezt a `geometry()` elnevezésű függvény téríti vissza. Az így kapott objektumon már különböző geometriai műveleteket is végrehajthatunk, például: terület- vagy kerületszámítás, metszéspontok és középpont megkeresése, poligonok esetén a vonalakon belül elhelyezkedő pontok meghatározása, formák összeillesztése, stb.

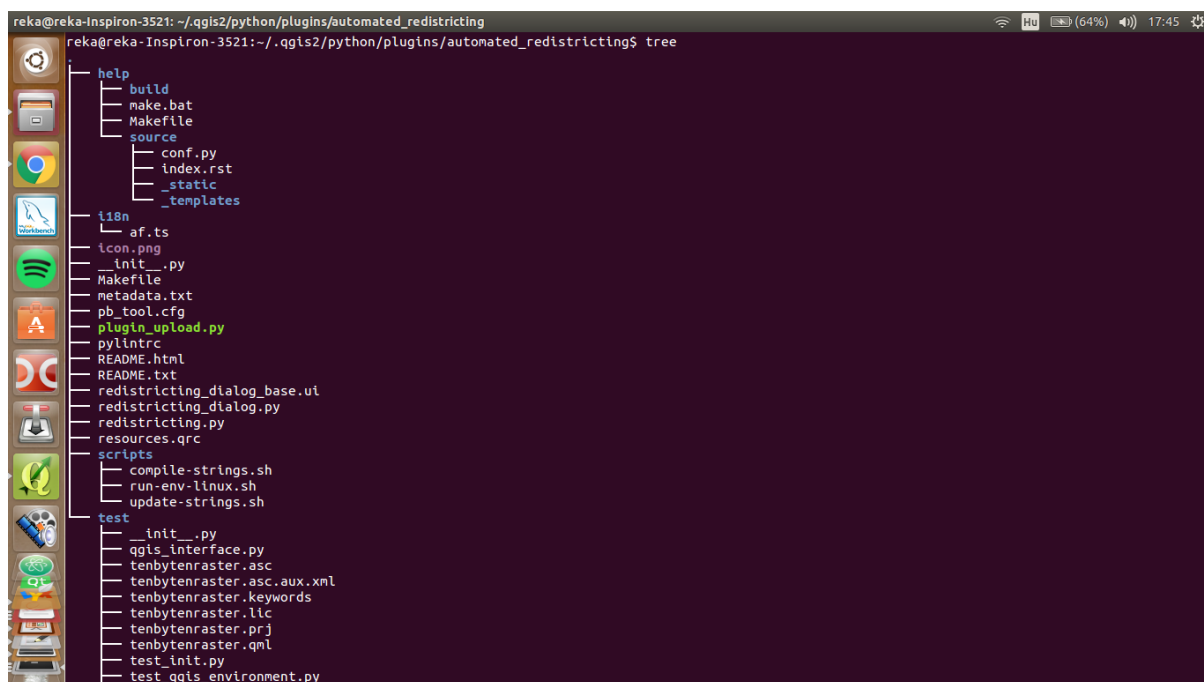
QGIS-ben létrehozhatunk saját modulokat, illetve letölthetünk mások által készítéseket. A saját modul készítését megkönnyíti a Plugin Builder, amely generál egy „üres” QGIS modult (amiben létrehozza a megfelelő könyvtárstruktúrát és a szükséges állományokat).

³(Geometry Engine, Open Source)



3.4. ábra. A Plugin Builder modul

A Plugin Builder az alábbi könyvtárstruktúrával rendelkező Python modult generálja:



3.5. ábra. generált könyvtárstruktúra

Az `__init__.py` állomány a modul belépési pontja, a `classFactory()` metódus hívódik meg a modul minden egyes betöltésekor. A metódusban lesz létrehozva a modulhoz tartozó osztály instanciája, amely paraméterként kapja a `QgisInterface` egy instanciáját. A `redistricting.py` fájl tartalmazza a főprogramot, a `run()` metódusa hívódik meg a

modul futtatásakor. A `redistricting_dialog_base.ui` a párbeszédablakot tartalmazza, a párbeszédablak-kezelő a `redistricting_dialog.py` nevű fájlban van. A generált modul továbbá tartalmazza a szintaktika ellenőrzéshez szükséges reguláris kifejezéseket (a `pylintrc` fájlban), a modul metainformációt és egy `makefajlt` a fordításhoz. Amint megvan a modul fejlesztéséhez szükséges keret, telepíthetjük is a modult a QGIS pluginkezelőjéből. A következő indításkor már az újonnan létrehozott modult is beolvassa a QGIS a `plugins` mappából, és a modul ikonja megjelenik a menüsorban. A modul további fejlesztése az általunk választott környezetek bármelyikében történhet, azonban a modul futtatása a QGIS programon belül lehetséges.

4. fejezet

A szavazókörzetek újraosztása

4.1. A szimulált hűtés módszere

A szimulált hűtés módszere egy metaheurisztikus algoritmus, amely alkalmas globális optimumok megközelítésére nagy megoldásterek esetén. 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 (lásd [7]), optimalizációs problémák megoldására. A cikkben leírják a termodinamikai rendszerek viselkedését, majd megmutatják hogy ezeket az elveket hogyan lehet felhasználni egy optimalizációs technika kifejlesztésére.

Egy termodinamikai rendszert különböző fizikai mennyiségek összessége ír le (hőmérséklet, energia, nyomás, entrópia, stb.), míg egy metaheurisztikus algoritmusnak csupán két eszközre van szüksége: *célfüggvényre* (vagy *költségfüggvényre*) és *szomszédos megoldást generáló módszerre*. A megfeleltetés a következő: az optimalizációhoz használt célfüggvény a rendszer energiaszintjének felel meg. Amikor a rendszer eléri az egyensúlyi állapotát, az energiaszint minimális. A cél a rendszer egyensúlyi állapotának az elérése, vagyis a célfüggvény minimumának a megtalálása. Egy szomszédos állapotot a jelenlegi állapoton eszközölt véletlen perturbációk által érünk el, ami 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 elegendő ha lehűtjük. Az analógiát követve, a keresés alacsony hőmérsékleten 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 kristályokba rendeződjenek. Nagy hőmérsékleten a molekulák kaotikus mozgást végeznek, ennek megfelelően az algoritmus is nagy ugrásokat tesz a megoldástérben, így elkerülhető az, hogy lokális minimumba ragadjunk. Azt, hogy a keresés mekkora valószínűséggel lép a jelenleginél rosszabb állapotba, a rendszer aktuális hőmérséklete befolyásolja. Az iterációk során akkor fogadjunk el egy új megoldást, ha az jobbnak bizonyul, mint a jelenlegi

megoldás. Amennyiben nem alacsonyabb az új megoldásunk célfüggvényének az értéke, akkor a jelenlegi megoldást az úgynevezett Metropolis-kritérium alapján

$$p = \exp((E - E_0)/T) \quad (4.1)$$

valószínűséggel cseréljük le, ahol E az aktuális megoldás objektív értéke (a célfüggvény értéke az aktuális megoldás esetén), E_0 a szomszédos megoldás objektív értéke, T pedig a rendszer hőmérséklete.

4.2. A körzetek újraosztása, többszörös célfüggvényt használó optimalizáció

A körzetek újraosztása két feladatot foglal magában. Először a meghatározott parlamenti helyeket (jelölés: m) szét kell osztani azon adminisztrációs egységek¹ között, amelyeken belül majd a körzeteink lesznek. A körzetek egységek közti szétosztására több módszer is létezik [8], mi a *legnagyobb maradék módszerét*² alkalmazzuk. Először kiszámítjuk a nemzeti átlagot: $N_m = p_{ország}/m$, ahol $p_{ország}$ az ország népességét jelöli. Kezdetben minden egységhez rendelünk $\lfloor p_e/N_m \rfloor$ darab körzetet, ahol p_e az adott egység népességét jelöli. A megmaradt körzeteket szétosztjuk a legnagyobb $frac(p_e/N_m)$ értékkel rendelkező egységek között, azaz mindegyikhez egy újabb körzetet adunk. A körzetek szétosztása után az újraosztási probléma felfogható úgy, mint egy egység, n darab alterületre, vagyis körzetre való felosztásának ismételt feladata. Egy érvényes felosztási terv alapfeltétele, hogy a körzetek folytonosak, és figyelembe veszik a már meglévő adminisztrációs határokat. Ezen feltételek a legegyszerűbben úgy biztosíthatóak, ha egy körzetet, olyan legkisebb adminisztrációs egységek (unitok) alkotnak, amelyek ugyanahhoz a nagyobb területi egységhez tartoznak. Az így kapott (Z, U) párokat egy érvényes felosztásnak nevezzük, ahol Z a körzeteket, $U = \{u_1, u_2, \dots, u_n\}$ a Z körzethez tartozó unitok halmazát jelöli.

Az olyan kritériumok, mint a körzetek folytonossága vagy a nagyobb adminisztrációs határok megőrzése, felfoghatóak úgy, mint az optimalizáció során betartandó megkötések. Azonban egy felosztási tervnek, olyan kritériumoknak is meg kell felelni, amelyeket függvényekkel mérhetünk. Gyakran előfordul, hogy ezek a kritériumok nem egyszerre teljesülnek, versengenek, vagy konfliktusban állnak egymással. Ilyenkor az optimalizációt nem végezhetjük egyetlen célfüggvény alapján, tehát nem is beszélhetünk egy optimális megoldásról. A többszörös célfüggvényt használó optimalizáció során a Pareto-hatékony megoldásokat [9] tartalmazó halmaz egy részhalmazának a megtalálására törekszünk. A Pareto-hatékony megoldás ebben az esetben azt jelenti, hogy $P = \{Z_1, Z_2, \dots, Z_n\}$ felosztás, és k darab célfüggvény esetén nem létezik $P' = \{Z'_1, Z'_2, \dots, Z'_n\}$, úgy hogy $f_p(P') < f_p(P)$, $p \in \{1, \dots, k\}$

¹Ezek általában nagyobb adminisztrációs egységek. Például: megyék, tartományok, zónák, stb..

²Angolul „largest remainder method”

és $f_p(P') \leq f_p(P)$, $\forall p \in \{1, \dots, k\}$.

Amikor több célfüggvény szerint optimalizálunk, egy megoldásról nem mindig állíthatjuk egyértelműen, hogy jobb vagy rosszabb, mint egy másik megoldás. Két megoldás összehasonlításánál három esetet különböztetünk meg (U jelöli az aktuális megoldást, V jelöli az új megoldást) :

1. V minden célfüggvény szerint jobb megoldás, mint U ;
2. V minden célfüggvény szerint rosszabb megoldás, mint U ;
3. V csak bizonyos célfüggvényekben teljesít jobban, mint U (az 1. és 2. eset nem teljesül).

Az első két eset a hagyományos optimalizáció eseteinek felel meg. Amikor V jobbnak bizonyul, akkor U -t lecseréljük, különben a szimulált hűtésnél leírt valószínűség szerint 4.1 felválthatja a jelenlegi megoldást. A 3. esetben a szakirodalom több lehetőséget is tárgyal [10, 11]. Legtöbbször az esetek összevonását tanácsolja, különböző skálázó függvényeket használva. A mi esetünkben azonban a megoldások is összevonhatóak, mivel két adminisztrációs egységen belüli felosztás egymástól független. Ilyenkor egy megoldást úgy kezelünk, mint az őt alkotó adminisztrációs egységek halmaza $P = \{A_1, A_2, \dots, A_n\}$. Az új megoldásunk: $P' = \{B_1, B_2, \dots, B_n\}$. A megoldások összevonása után egy olyan $P'' = \{C_1, C_2, \dots, C_n\}$ megoldást kapunk, ahol:

$$C_i = \begin{cases} B_i & \text{ha } f_p(B_i) \leq f_p(A_i) \ \forall p \in \{1, \dots, k\} \\ A_i & \text{különben} \end{cases}. \quad (4.2)$$

Tehát, amikor olyan megoldást találunk, amely nem dominál és nem dominált más megoldások által, a Pareto-halmazhoz adjuk [14], azonban amikor a halmaz eléri az általunk megadott maximális méretét, a 4.2 formula által kapott megoldás felváltja az aktuális megoldást. A fent felsorolt három eset elkülönítésére egy lehetséges algoritmus a következő oldalon leírt *pareto()* algoritmus, amit körzetek generálására alkalmazott több célfüggvényt használó szimulált hűtés módszerének algoritmusaként követ (*SimulatedAnnealing()* néven). A második algoritmusban az U bemeneti változó egy véletlenszerűen generált első megoldás, T a kezdeti hőmérséklet, $coolingschedule \in (0, 1)$, k a célfüggvények száma, $maxsize$ pedig a Pareto-halmaz maximális mérete.

Algorithm 1: dominance()

input : V , $pareto$, k , where $k \geq 1$

output: $dominated$, $dominates$

$dominated \leftarrow \text{False};$

$dominates \leftarrow \emptyset;$

for $S \in pareto$ **do**

$dominatedbyS \leftarrow \text{True};$

$dominatesS \leftarrow \text{True};$

for $i = 1, \dots, k$ **do**

$dominatedbyS \leftarrow dominatedbyS$ **and** $S.objective(i) < V.objective(i);$

$dominatesS \leftarrow dominatesS$ **and** $V.objective(i) < S.objective(i);$

$isdominated \leftarrow dominated$ **or** $dominatedbyS;$

if $dominatesS$ **then** $dominates \leftarrow dominates \cup \{s\};$

Algorithm 2: SimulatedAnnealing()

input : U , T , $maxiter$, $coolingschedule$, k , $maxsize$

output: $pareto$

$pareto \leftarrow \{U\}$

while T not frozen **do**

for $i = 1, \dots, maxiter$ **do**

$V \leftarrow neighbor(U);$

if V dominates **then**

$pareto.update(V);$

$U \leftarrow V;$

else

if V is dominated **then**

if $random(0, 1) \leq \exp((obj(U) - obj(V))/T)$ **then**

$U \leftarrow V;$

else

if $length(pareto) > maxsize$ **then**

$merge(U, V);$

else

$pareto \leftarrow pareto \cup \{V\};$

$U \leftarrow V;$

 layer-ek betöltése. Amint betöltünk egy layer-t, az láthatóvá válik

$T \leftarrow T * coolingschedule;$

$U \leftarrow pareto.random();$

4.3. Az implementáció részletei, a célfüggvények

A dolgozatban bemutatott program bemeneti adatai a következők: a generálandó körzetek száma (2), a felosztandó ország legkisebb adminisztrációs területeit tartalmazó shapefájl, a kulcsmező³ neve (2), a unitokat (3), az azonosítót (5), a terület népességét (4), a szomszédos unitokat (6), valamint a poligonokhoz tartozó vonalakat (7) tároló mezők nevei, illetve a területekhez tartozó határvonalakat tartalmazó shapefájl és a mező neve amely alapján a vonalakat a unitokhoz kapcsolhatjuk (8). A felsorolt adatok megadása a modulban található „config.yaml” elnevezésű fájlban tehető meg:

config.yaml

```
1 attributes:
2     join_id: 'regionId'
3     attribute_name: 'name'
4     attribute_population: 'pop2015'
5     attribute_id: 'natcode'
6     attribute_neighbours: 'neighbours'
7     attribute_lines: 'lines'
8     attribute_id_poliline: 'id'
```

A későbbiekben egy unit szomszédainak a bejárására gyakran kerül sor, ezért a modul első használatakor a „neighbours.py” elnevezésű script lefuttatása is szükséges. A script meghatározza az egymással szomszédos unitokat. Az így kapott azonosítók tárolása sok futási időt spórol majd az új felosztások meghatározása közben. A további paraméterek beállításával az optimalizáló algoritmus viselkedését változtathatjuk: a kezdeti (4) és végső hőmérséklet (5), a hűtés mértéke (3), a hőmérsékletenkénti iteráció (6), valamint a hőmérsékletenkénti iterációk számának a növekedése (7). Továbbá megadható a megoldáshalmaz maximális mérete (11), a költségfüggvényekhez rendelhető lehetséges súlyok listája (9), és az egy körzeten belüli minimális unitszám (10). Valamit az új felosztást kereső algoritmus cseréinek a száma is változtatható (8).

config.yaml

```
1 parameters:
2     nr_of_districts : 120
3     cooling_schedule: 0.99
4     initial_temperature: 5
5     final_temperature: 0.001
6     iterations: 1800
7     iterations_increment : 0
```

³ezen értékek alapján csoportosíthatjuk a unitokat nagyobb adminisztrációs egységekbe (régión, megye, stb..)

```

8      neighbourhood: [-1]
9      weight_vectors : [[0.8 , 0.2]]
10     minnr_of_unit : 2
11     max_size_pareto: 100

```

Az optimalizáció során alkalmazott célfüggvényeket az „objectives.py” elnevezésű fájl tartalmazza, ezek bővítése nem adható meg a konfigurációs fájlban. Amennyiben újabb függvényt szeretnénk egy megoldás kiértékeléséhez adni, azt implementálnunk kell a függvényeket tartalmazó fájlban.

A dolgozatban bemutatott program, a mexikói újraosztási folyamatok mintájára, két célfüggvényt használ. Ezeket használta az IFE (Instituto Federal Electoral) a 2006-os választásokat megelőző újraosztások során. A C_1 célfüggvény a populáció egyenletes eloszlását méri:

$$C_1(P) = \sum_{e \in E} \left(\frac{P_e}{d_e \cdot N_m} \right)^2 \sum_{s \in S} \left(\frac{P_s}{P_e} - \frac{1}{n_e} \right)^2,$$

ahol $P = \{Z_1, Z_2, \dots, Z_n\}$ egy felosztási terv, N_m a nemzeti átlagot jelöli, P_e az adott egység populációja. Az egységen belül megengedett szórást d_e -vel jelöljük. A szorzat második tagjában levő P_s jelöli az adott körzet populációját, n_e az egységhez rendelt körzetek száma. Az, hogy egy egységben megengedett szórás értéke mekkora, annak a függvényében változik, hogy az egység átlaga mennyire tér el a nemzeti átlagtól. Amikor egy egységen belül a körzetek populációja megegyezik, a szorzat második tagja zéróval egyenlő. Ha ez minden egység esetén teljesül, elértük az optimális népesség egyenlőséget, tehát $C_1(P) = 0$.

A második célfüggvény a körzetek formájának a kompaktságát méri. A legkompaktabb geometriai alakzat a kör. A függvény azt méri, mennyire hasonlít a körzetek formája egy kör formájához:

$$C_2(P) = \sum_{s \in S} \left(\frac{PE_s}{4\sqrt{A_s}} - 1 \right)$$

Egy PE_s kerületű, és A_s területű körzet, annál kompaktabb, minél inkább megegyezik a kerülete egy A_s területű kör kerületével.

5. fejezet

Eredmények

5.1. Eredmények

A dolgozatban bemutatott program segítségével, a 2015-ben tartott népszámlálási adatok alapján, Romániát 136 körzetre osztottuk. Az így kapott körzeteket Románia legkisebb adminisztrációs egységei alkotják¹. A felosztást a 8 zónán, majd a 42 megyén belül is elvégeztük. A körzetek szétosztása a következő:

Megye	Populáció	Egész körzet	Direkt hozzárendelés	Törtrész szerinti hozzárendelés	Összesen
Alba	382537	2			2
Arad	475216	2		1	3
Arges	649014	3		1	4
Bacau	748043	4		1	5
Bihor	620531	3		1	4
Bistrita-Nasaud	329471	2			2
Botosani	458620	2		1	3
Brasov	630151	3		1	4
Braila	359721	2			2
Buzau	482716	2		1	3
Caras-Severin	331092	2			2
Cluj	720722	4		1	5
Constanta	770465	4		1	5
Covasna	229319	1	1		2
Dambovita	530745	3			3
Dolj	704097	4			4
Galati	634412	3		1	4

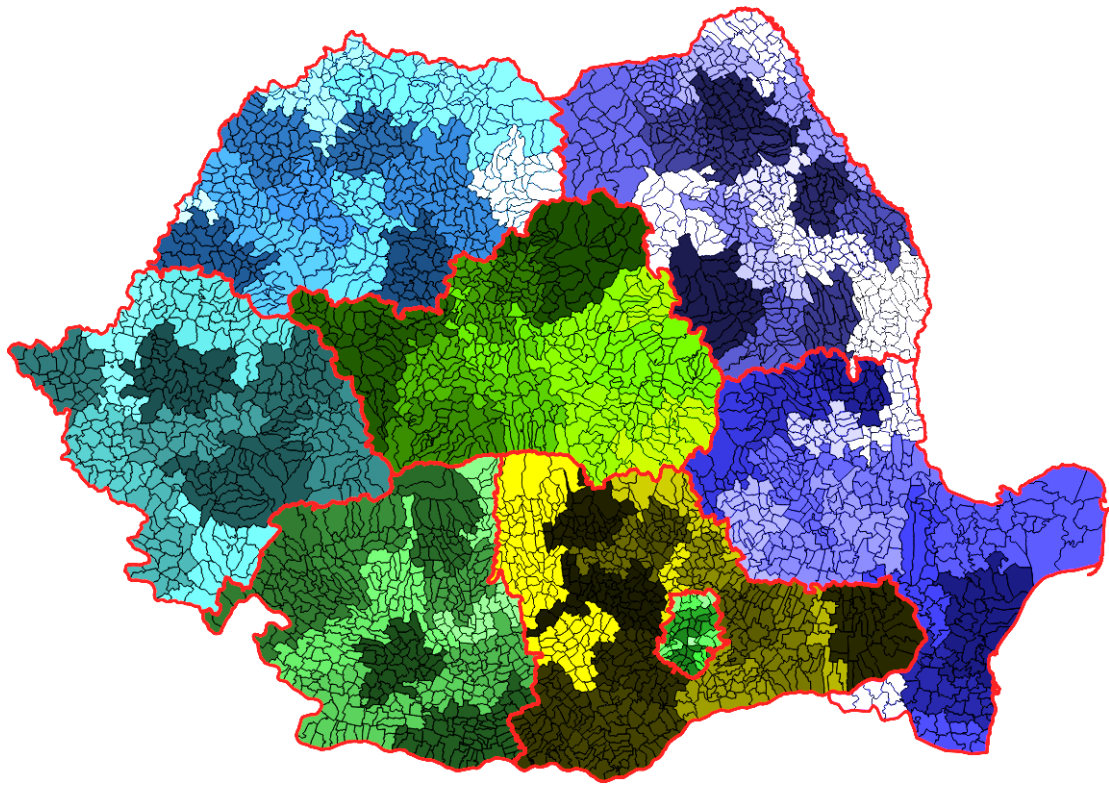
¹Románul „Unități administrativ-teritoriale”

Gorj	368786	2			2
Harghita	334481	2			2
Hunedoara	473635	2		1	3
Ialomita	295481	1	1		2
Iasi	907979	5		1	6
Ilfov	376536	2			2
Maramures	526952	3			3
Mehedinti	289080	1	1		2
Mures	597386	3		1	4
Neamt	579948	3		1	4
Olt	454686	2		1	3
Prahova	813759	4		1	5
Satu Mare	391840	2			2
Salaj	248431	1	1		2
Sibiu	463598	2		1	3
Suceava	741749	4		1	5
Teleorman	394499	2		1	3
Timis	740288	4		1	5
Tulcea	246309	1	1		2
Vaslui	478179	2		1	3
Valcea	405431	2		1	3
Vrancea	392591	2		1	3
Bucuresti	2103346	12	-6		6
Calarasi	319381	1	1		2
Giurgiu	277960	1	1		2

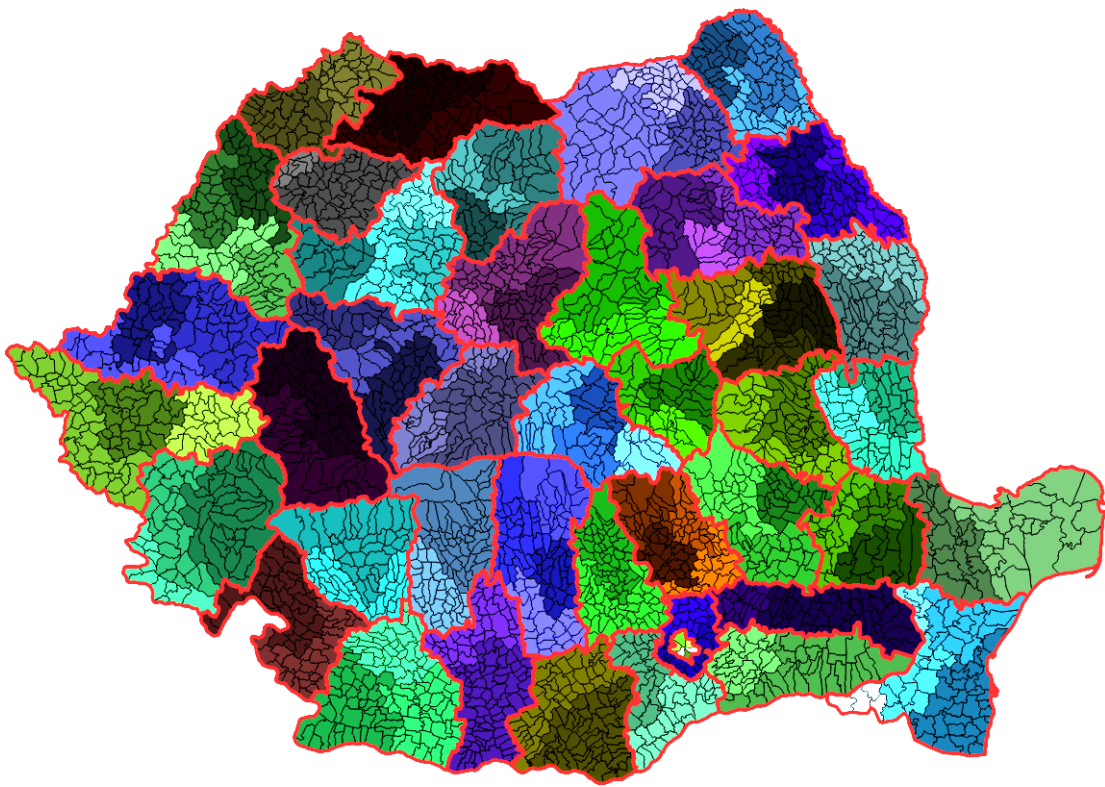
Körzetek zónák közötti szétosztására a következő eredményt kaptuk:

Zóna	Populáció	Egész Körzet	Törtrész szerinti hozzárendelés	Összesen
Észak-Kelet	3914518	23	1	24
Dél-Kelet	2886214	17	1	18
Dél	3280839	20		20
Dél-Nyugat	2222080	13	1	14
Nyugat	2020231	12		12
Észak-Nyugat	2837947	17		17
Központi	2637472	16		16
Bukarest-Ilfov	2479882	15		15

Az algoritmus a következő körzeteket generálta:



5.1. ábra. Régióként generált körzetek



5.2. ábra. Megyénként generált körzetek

S

A piros vonalak a régió-, illetve megyehatárokat mutatják. A határokon belül minden körzet különböző színárnyalatban látható. Az algoritmus 1700 iterációt végzett hőmérsékletenként. A populáció egyenletes eloszlását értékelő függvény súlya 0.8, míg a kompaktságot értékelő 0.2. Az optimalizációt egy véletlenszerűen generált megoldással kezdtük, ahol a populáció eloszlásának szórása 30%-os volt. Ez az optimalizáció után 10%-ra csökkent.

5.2. Továbbfejlesztési lehetőségek

Az elkövetkezendőekben szeretnénk az algoritmus olyan paramétereit megtalálni, amelyekkel a népesség szórása tovább csökkenthető. Ugyanakkor újabb célfüggvények behozatala is lehetséges. Például figyelembe lehetne venni a kisebbségek eloszlását, vagyis olyan körzeteket generálni ahol többségben vannak².

²positive gerrymandering

Irodalomjegyzék

- [1] S. Coate, B. Knight, *Socially Optimal Districting: A Theoretical and Empirical Exploration*, The Quarterly Journal of Economics, Vol. 122, Issue 4.
- [2] I. Elek, *Bevezetés a geoinformatikába*, ELTE Eötvös Kiadó (Budapest), 2006.
- [3] M. Belényesi, D. Kristóf, J. Magyari, *Térinformatika Elméleti Jegyzet.*, Szent István Egyetem, Mezőgazdaság- és Környezettudományi Kar, Gödöllő, 2008, http://zeus.nyf.hu/~foldrajz/geo/files/elmeleti_jegyzet.pdf.
- [4] G. Padányi-Gulyás, *Segédlet a Quantum GIS régészeti célú felhasználásához*, http://www.agt.bme.hu/gis/qgis/regesz_segedlet_qgis.pdf.
- [5] G. Szabó, *Quantum GIS Felhasználói kézikönyv*, 2010, <http://mek.oszk.hu/09200/09238/09238.pdf>.
- [6] Z. Siki, *QGIS Python programozás*, BME Geodézia, 2009, <http://oktatas.epito.bme.hu/local/coursepublicity/mod/resource/view.php?id=37215>.
- [7] 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.
- [8] I. McLean, R. Mortimore, *Apportionment and the Boundary Commission for England*, Electoral Studies, Volume 11, Issue 4, December 1992, pp. 293-309.
- [9] S. Yoo, M. Harman, *Pareto Efficient Multi-Objective Test Case Selection*, ISSTA '07, July 9-12, London, U.K.
- [10] E.L. Ulungu, J. Teghem, P.H. Fortemps, D. Tuyttens, *MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems*, Service de Mathématique & Recherche Opérationnelle, Faculté Polytechnique de Mons, Mons, Belgium.
- [11] J. Teghem, *Methodologies For Solving Multiobjective Combinatorial Optimization Problems*, Faculté Polytechnique de Mons, Belgium.
- [12] C. Ingraham, *How to steal an election: a visual guide*, <https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/>.

- [13] S. Nass, *A chart illustrating gerrymandering in its most basic form*, https://commons.wikimedia.org/wiki/File:How_to_Steal_an_Election_-_Gerrymandering.svg (CC-BY-SA-4.0).
- [14] E. A. Ringc3n-Garc3a, M. A. Guiterrez-Andrade et al., *A Multiobjective Algorithm for Redistricting*, Journal of Applied Research and Technology Vol. 11, Issue 3, June 2013, pp. 324–330.
- [15] S. Velasquez, *P3gina de l'assignatura ALE 'Cartografia i Noves Tecnologies Cartogr3fiques'*, http://www.catalonia.org/cartografia/Clase_03/Raster_Vector.html.