Districting and Gerrymandering

Andrea Scozzari
University Niccolò Cusano

Caen, July 8-12 2014

Definitions

Political Districting (PD) consists of subdividing a given territory into a fixed number of districts in which the election is performed. A given number of seats, generally established on the basis of the population of the district, is allocated to each district. These seats must be assigned to parties within the district according to the adopted electoral system that rules out how the citizens' votes are transformed into seats.

The PD problem has been studied since the 60's and many different models and techniques have been proposed with the aim of preventing districts' manipulation to favor some specific political party (*gerrymandering*).

Definitions

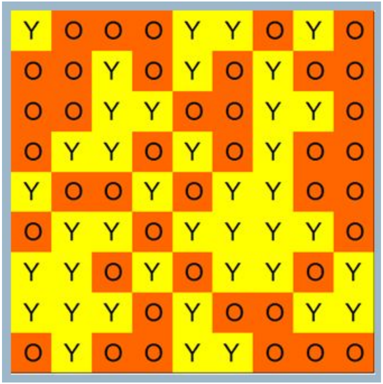
Given the vote distribution, different district plans may reverse the outcome of an election (see R.J. Dixon and E. Plischke (1950), American Government: Basic Documents and Materials, New York, Van Nostrand.)

Neutral district plans are necessary to oppose partisan manipulation of electoral district boundaries (*gerrymandering*)

The aim is to provide automatic procedures for political districting, designed so as to be as neutral as possible

Example

1. A territory divided into 81 elementary units (sites) of equal population
2. Each site is colored yellow (40 sites) or orange (41 sites) that constitute the vote distribution
3. 9 (uninominal - 1 seat at stake) districts must be drawn, each formed by 9 contiguous sites (perfect population equality)

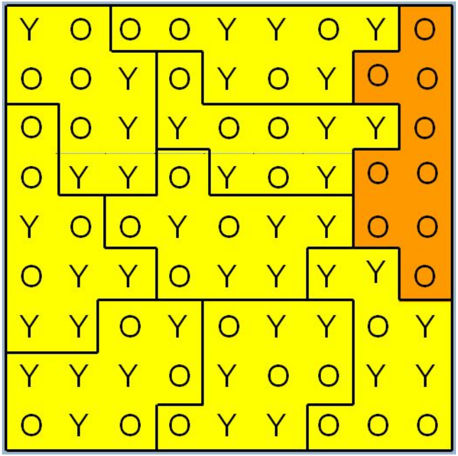


Example

Try to make **Yellow/Orange** party win as many seats as possible!! by drawing 9 districts

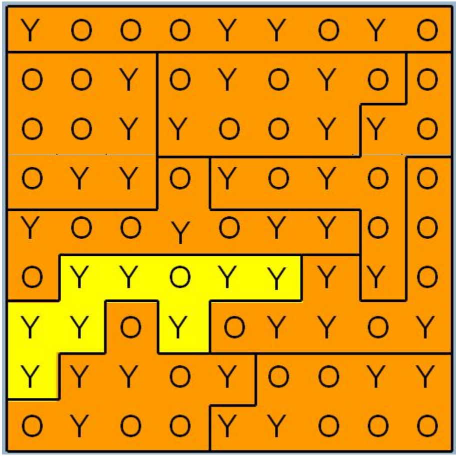
Example

Try to make **Yellow/Orange** party win as many seats as possible!! by drawing 9 districts



Yellow party wins 8 seats, Orange party wins 1 seat!

Example

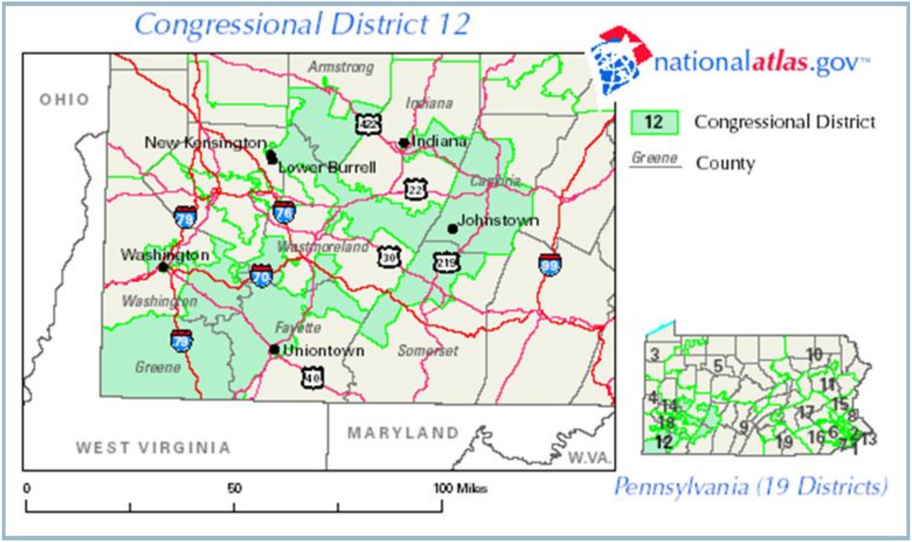


Orange party wins 8 seats, Yellow party wins 1 seat!

Gerrymandering

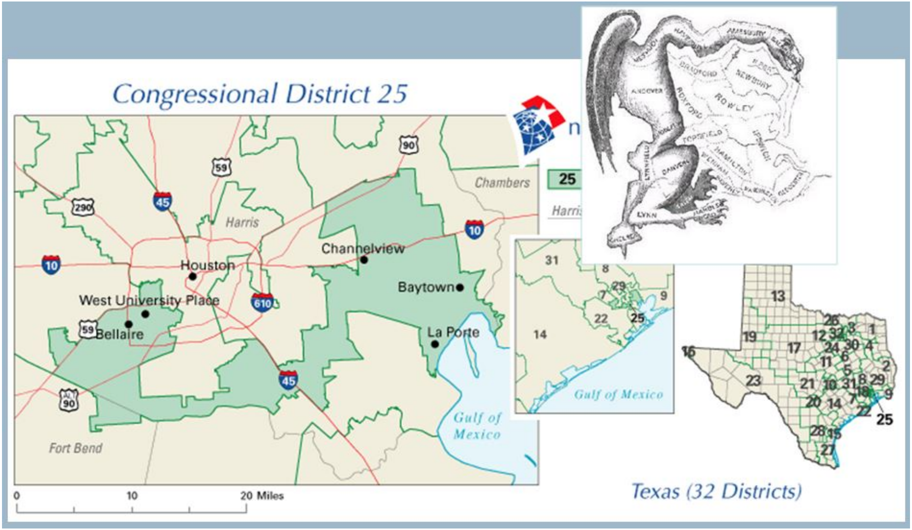
This was what happened in Massachusetts in 1821 when the governor Elbridge Gerry drew the electoral districts in order to be re-elected. In this way, he was able to take advantage from the territorial subdivision in order to gain seats. This bad malpractice is known as gerrymandering from a particular salamander-shape of one of the districts obtained.

Presidential election USA 2004: Pennsylvania Districts 12

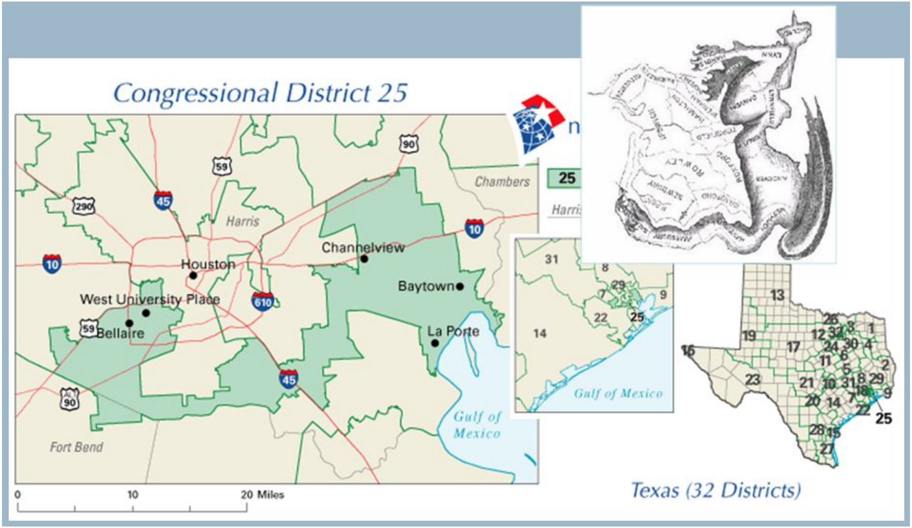


Presidential election USA 2004: Pennsylvania Districts 12

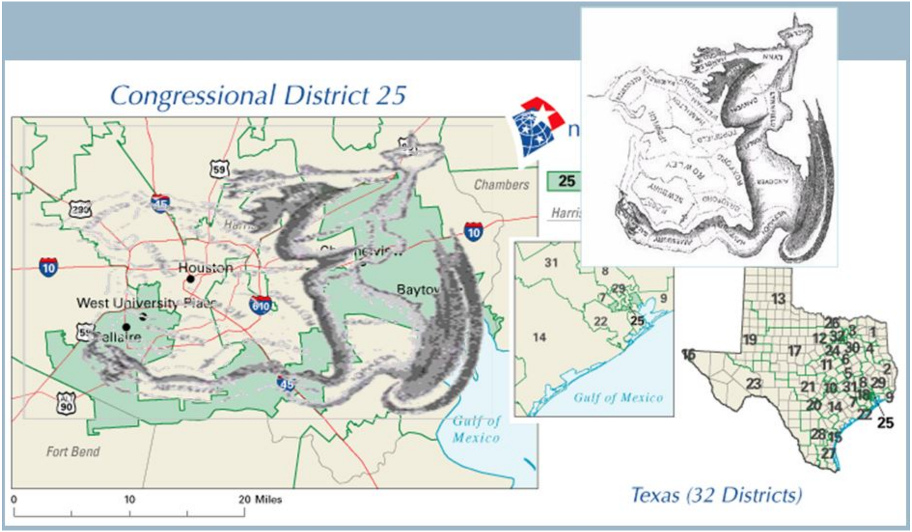
Presidential election USA 2004: Pennsylvania Districts 12



Presidential election USA 2004: Pennsylvania Districts 12

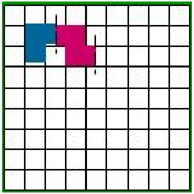


Presidential election USA 2004: Pennsylvania Districts 12

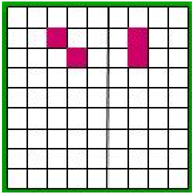


Political Districting Criteria

1. *Integrity*: Each territorial unit belongs to only one district and it cannot be split between two different districts.

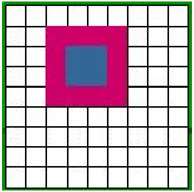


2. *Contiguity*: A district is formed by a set of geographically contiguous units.

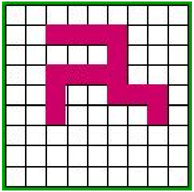


Political Districting Criteria

3. *Absence of enclaves*: No district can be fully surrounded by another district.



4. *Compactness*: The districts must have *regular* geometric shapes. Octopus- or banana-shaped districts must be avoided.



Political Districting Criteria

5. *Population equality*: Districts populations must be as balanced as possible.

Political Districting Criteria

5. *Population equality*: Districts populations must be as balanced as possible.

There are other PD criteria that are seldom used, among the others we mention:

- *the respect of natural boundaries*
- *fair representation of ethnic minorities*
- *respect of integrity of communities* ⋮

Political Districting Indicators

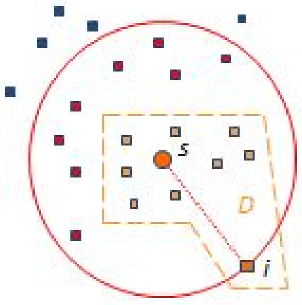
There is the need to define correct indicators to measure the above criteria.

Compactness

The compactness of a district depends on its area, on the distances between territorial units and the district center, the perimeter, the geometrical shape, its length and its width, the district population, and so on.

1. Dispersion measures: district area compared with the area of canonical compact figure (for example the circle);
2. Perimeter based measures: perimeter compared with area;
3. Population measures: district population compared with the population of the smallest compact figure (for example a circle) which contains the whole district.

Political Districting Indicators



This is a measure of compactness of a district D obtained by computing the percentage of sites in the circle centered in the center s of radius d_{is} that do not belong to D (Arcese, Battista, Biasi, Lucertini, Simeone, 1992).

Political Districting Indicators

The above index can be refined so as to evaluate compactness also with respect to population since each territorial unit can be weighted by its population.

Let P_h^d be the total population of the units within the circle of radius d, then the compactness index is:

$$\sum_{h=1}^{K} \frac{P_h^d - P_h}{P_h^d}$$

K = the total number of districts

P_h = the population of district h

Political Districting Indicators

Moment of Inertia

Let c be a point in district D. The moment of inertia of district D with respect to c is the weighted sum of the squared distances of all territorial units in D from c. The weight of each distance is given by the population of the corresponding territorial unit.

The moment of inertia is minimized by setting c equal to the center of gravity g of the district.

$$\sum_{i=1}^{n_h} p_i \cdot (d_i^{g_h})^2$$

n_h = number of units in district h

$d_i^{g_h}$ = the distance between unit i in h and the center of gravity of h

p_i = population in the territorial unit i

Political Districting Indicators

Population Equality

The most popular indexes of population equality are global measures of the distance between the populations of the districts and the mean district population $\overline{P}$.

$$\frac{\sum_{h=1}^{K} |P_h - \overline{P}|}{K}$$

K = the total number of districts

P_h = the population of district h

Political Districting Indicators

Population Equality

Other indexes can be built simply by replacing the L_1 norm by other norms:

$$\frac{\sum_{h=1}^{K} (P_h - \overline{P})^2}{K}$$

K = the total number of districts

P_h = the population of district h

Political Districting Indicators

Population Equality

Unfortunately, the range of these measures depends on the size of the total population, so relative measures with values in the $[0, 1]$ interval are usually preferred (Arcese, Battista, Biasi, Lucertini, Simeone, 1992):

$$\frac{\sum_{h=1}^{K} |P_h - \overline{P}|}{2(K-1)\overline{P}}$$

K = the total number of districts

P_h = the population of district h

Political Districting Indicators

Population Equality

A very different index is given by the *inverse coefficient of variation* (ICV) (Shubert and Press, 1964)

$$\sqrt{\frac{\sum_{h=1}^{K}(\frac{P_h}{\overline{P}}-1)^2}{K}}$$

K = the total number of districts

P_h = the population of district h

Political Districting Problem

Suppose that political districts must be designed for a territory divided into n elementary units *population units*. Let $k < n$ be the total number of districts to be obtained; denote by p_i the population resident in unit i, $i = 1, \ldots, n$.

Let $P = \sum_{i=1}^{n} p_i$ be the total population of the territory, and $\overline{P} = \frac{P}{k}$ the average district population

Find a compact partition of a given territory into k connected components such that the weight of each component (i.e., the sum of the weights p_i of the units in the component) is as close as possible to $\overline{P}$.

Political Districting Approaches

1. *Integer Linear Programming (ILP) approaches*

Hess et al. 1965 is considered the earliest Operations Research paper in political districting. The idea is to identify k units representing the centers of the k districts, so that each territorial unit must be assigned to exactly one district center (*Location approach*). Assume d_{ij} the distance between unit i and unit j:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2\, p_i\, x_{ij} & \\
& \sum_{j=1}^{n} x_{ij} = 1 & i = 1,\ldots,n \\
& \sum_{j=1}^{n} x_{jj} = k & \\
& a\,\bar{P}\,x_{jj} \le \sum_{i=1}^{n} p_i\, x_{ij} \le b\,\bar{P}\,x_{jj} & j = 1,\ldots,n \\
& x_{ij} \in \{0,1\}, & i,j = 1,\ldots,n
\end{aligned}
\qquad (1)
$$

ILP Approaches

$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^2 \, p_i \, x_{ij}$: is a measure of *compactness*.

$a \, \bar{P} \, x_{jj} \leq \sum_{i=1}^{n} p_i \, x_{ij} \leq b \, \bar{P} \, x_{jj}$: measures the *Population equality*, with $a < 1$ and $b > 1$ the minimum and the maximum allowable district population fractions.

NOTE The above integer programming model does not consider contiguity of the units belonging to the same district, so that a revision for spatial contiguity is required a posteriori.

ILP Approaches

Garfinkel and Nemhauser 1970, proposed a two-phases approach based on a set partitioning technique. In phase I, they generate all possible feasible districts w.r.t. three types of constraints related to contiguity, population equality and compactness, respectively, and denote this set by J;

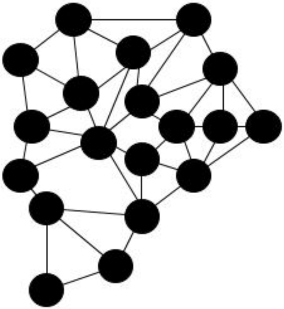
$$
\begin{aligned}
\min \quad & \sum_{j \in J} f_j \, x_j \\
& \sum_{j \in J} a_{ij} \, x_j = 1 \qquad i = 1, \dots, n \\
& \sum_{j \in J} x_j = k \\
& x_j \in \{0, 1\} \qquad j \in J
\end{aligned} \tag{2}
$$

where $f_j = \frac{|P_j - \bar{P}|}{\alpha \, \bar{P}}$ ($\alpha \in [0, 1]$ is the tolerance percentage of deviation for the population of a district from $\bar{P}$); $a_{ij} = 1$ if unit i is in district j and $a_{ij} = 0$ otherwise; $x_j = 1$ if district $j \in J$ is included in the partition.

ILP Approaches

Network flow approach

Many authors adopt a graph-theoretic model for representing the territory on which districts must be designed. The territory can be represented as a connected n-node graph $G = (N, E)$, where the nodes correspond to the elementary territorial units and an edge between two nodes exists if and only if the two corresponding units are neighboring (they share a portion of boundary). The graph G is generally known as *contiguity graph*.



ILP Approaches: Network flow approach

$$
\begin{aligned}
\min \quad & u - l & & \\
& l \leq \sum_{i=1}^{n} p_i \, z_{ih} \leq u & h = 1, \ldots, k & \\
& \sum_{a \in \delta^-(v_i^h)} f(a) = \sum_{a \in \delta^+(v_i^h)} f(a) & i = 1, \ldots, n & \quad h = 1, \ldots, k \\
& f(a) \geq 0 & a \in \bar{A} & \\
& f(s^h, v_i^h) = \beta \, y_{ih} & i = 1, \ldots, n & \quad h = 1, \ldots, k \\
& \sum_{i=1}^{n} y_{ih} = 1 & h = 1, \ldots, k & \\
& y_{ih} \in \{0, 1\} & i = 1, \ldots, n & \quad h = 1, \ldots, k \\
& \sum_{a \in \delta^-(v_i^h)} f(a) \leq \beta \, z_{ih} & i = 1, \ldots, n & \quad h = 1, \ldots, k \\
& z_{ih} \leq f(v_i^h, t_i), & i = 1, \ldots, n & \quad h = 1, \ldots, k \\
& \sum_{h=1}^{k} z_{ih} = 1 & i = 1, \ldots, n & \\
& z_{ih} \in \{0, 1\} & i = 1, \ldots, n & \quad h = 1, \ldots, k
\end{aligned}
\tag{3}
$$

ILP Approaches: Network flow approach

$\sum_{a \in \delta^-(v_i^h)} f(a) = \sum_{a \in \delta^+(v_i^h)} f(a)$; this is the classical *flow balance* constraints that guarantees *Contiguity*

$l \leq \sum_{i=1}^{n} p_i z_{ih} \leq u$; along with the objective function *control* the *Population equality* criterion

The model takes into account integrity, contiguity and population equality, but **compactness** of the districts is not guaranteed.

Heuristic Approaches

Due to the *computational* difficulty in solving the above models, in the literature alternative *non exact* or *heuristic* solution approaches have been proposed (see Ricca, Scozzari, Simeone 2013 for a comprehensive review, and the references therein).

1. *Local Search Techniques* They are very general methods which are usually adopted to find solutions for computationally difficult combinatorial problems when an exact algorithm cannot be applied. Some of them are very simple to implement:

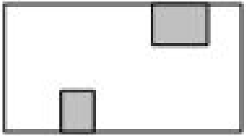
- **Multi-Kernel** growth strategy: a district map can be obtained in an incremental fashion. A set of territorial units is generally selected at the beginning as the set of centers (or potential centers) of the districts and the algorithm proceeds by adding neighboring units to the district under construction in order of increasing distance, until a certain population level is reached, and stopping when all units are assigned to some district.

Heuristic Approaches

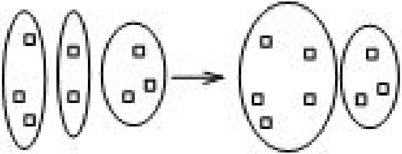
2. *Computational Geometry*

There is a new class of methods that borrow notions and techniques from the computational geometry area. Specifically, some papers refer to Voronoi regions or diagrams. These methods perform a discretization of the territory and use the (weighted) discrete version of the Voronoi regions. All these techniques are heuristics and generally take into account contiguity, compactness and balance of the populations of the districts.

Heuristic Approaches General Strategies

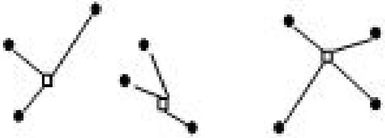


Eat-up



Agglomerative clustering

Multi-kernel growth



Location

General Local Search Procedure

Descent algorithm

1. Select an initial solution s;
2. While some feasible migration in s exists, do:
2.1. try to perform one of the feasible migrations, obtaining a new solution s'. Let $\Delta = f(s') - f(s)$ be the variation of the objective function when going from s to s';
2.2. if $\Delta \leq 0$:
 the migration is performed;
2.3. else [$\Delta > 0$]:
 the migration rejected;
 endif;
3. endwhile.
4. Output the best solution s found up to now.

Tabu Search

Tabu search [Glover, 1989 and 1990]

1. Select an initial solution s ;
2. Do:
   - 2.1. generate a set of feasible migrations from s ;
   - 2.1.1. if there are non-tabu moves which produce a new solution s' such that $\Delta = f(s') - f(s) \leq 0$:
     then perform the one with the largest absolute value of Δ ;
   - 2.1.2. else [every non-tabu move produces a new solution s' such that $\Delta = f(s') - f(s) > 0$]:
     perform the one which produces the lowest $\Delta > 0$;
     endif;
   - 2.2. update the tabu list;
3. until the stopping condition is satisfied.
4. Output the best solution s found up to now.

Old Bachelor Acceptance

Old Bachelor Acceptance [Hu et al., 1995]

1. Select an initial feasible solution s ;
2. Fix an initial threshold T_0 .
3. For $i = 1, \ldots, m$:

3.1. try to perform a feasible migration from s obtaining a new feasible solution s' . Let $\Delta = f(s') - f(s)$ be the variation of the objective function when going from s to s' ;

3.2. if $\Delta \leq 0$ (downhill move):

the move is performed;

$T_{i+1} := T_i - \Delta^-(i)$;

3.3. else [$\Delta > 0$] (uphill move):

the move is not performed;

$T_{i+1} := T_i + \Delta^+(i)$;

endif;

endfor.

4. Output the best solution s found up to now.

F. Ricca, A. Scozzari, B. Simeone (2013). Political Districting: from classical models to recent approaches. ANNALS OF OPERATIONS RESEARCH, vol. 204, p. 271-299

Approach	60's	70's	80's	90's	2000's
Multi kernel growth	Vickrey, 1961	Bodin, 1973		Arcese et al., 1992	
Location	Hess et al., 1965		Hojati, 1996 George et al., 1997		
Exact appr.		Garfinkel et el., 1970	Nygreen, 1988	Merhotra et al., 1998	Nemoto and Hotta, 2003; Li et al., 2007
Local search				Bourjolly et al., 1981; Ricca, 1996	Ricca and Simeone, 2008; Bozkaya et al., 2003; Forman and Yue, 2003; Baçao et al., 2005; Yamada, 2009
Comput. geometry					Kalcsics et al., 2005; Ricca et al., 2007, 2008; Miller, 2007

Table 1. PD algorithmic approaches.

Districting and Gerrymandering: an algorithm

Consider:

- a connected contiguity graph $G = (V, E)$, whose nodes represent the territorial units and there is an edge between two nodes if the two corresponding units are neighboring;
- a positive integer r, the number of districts;
- a subset $S \subset V$ of r nodes, called centers (all remaining nodes will be called sites);
- a positive integral node weights p_i, representing territorial unit populations;
- positive real distances d_{is} from a site i to a center s, $\forall\, i, s$.

Districting and Gerrymandering:

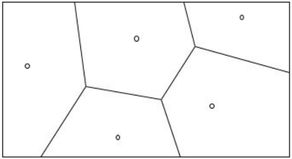
F. Ricca, A. Scozzari, B. Simeone (2008) Weighted Voronoi region algorithms for political districting, Mathematical and Computer Modelling, vol. 48, 1468-1477.

Multiobjective graph-partitioning formulation:

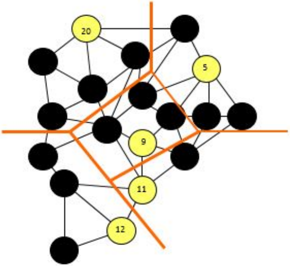
Given the contiguity graph G, partition its set of n nodes into r classes such that the subgraph induced by each class is connected (connected r-partition of G) and a given vector of functions of the partition is minimized.

NOTE: Compactness and population equality are generally taken as OBJECTIVES, while integrity, contiguity and absence of enclaves are commonly taken as CONSTRAINTS.

Voronoi Approach

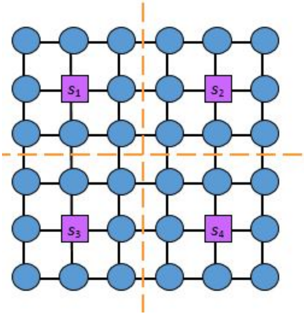


We adopt the traditional graph partitioning formulation and design the district map by drawing the graph Voronoi diagram w.r.t. the distances d_{is} (Discrete Voronoi regions)



Voronoi Approach

If one takes as districts the ordinary Voronoi regions w.r.t. the distances d_{is}, a good compactness is usually achieved.



The district-map obtained by the Voronoi regions is compact, but a poor population balance might ensue!

Voronoi Approach

The *initial Voronoi region* (or diagram) of center a center s is the set of all nodes i such that the closest center to i is s.

In order to re-balance district populations, one would like to promote site migration out of heavier districts (populationwise) and into lighter ones.

Site migration can be performed by considering weighted distances. We consider two different approaches: At a given iteration k of the procedure we (re)-compute:

- Static: $d_{is}^k = (frac P_s^{k-1} \overline{P}) d_{is}$

- dynamic: $d_{is}^k = (frac P_s^{k-1} \overline{P}) d_{is}^{k-1}$

Voronoi Approach

Site migration can be also performed via two other strategies:

- Single transfer: Voronoi regions are calculated at the beginning. At each iteration only one site moves to a new district.

- Full transfer: Voronoi regions are calculated iteratively. At each iteration a number of sites move from its own district to a new one.

- Partial transfer: Voronoi regions are calculated iteratively. Only a particular subset of sites (suitably selected according to some rule) migrates at each iteration.

Voronoi Approach

In particular, the implementation of the single transfer procedure is the following: at iteration k, some district D_t with minimum population, $P_t^{k1} \min\{P_s^{k1} : s = 1, \ldots, r\}$ is selected as the destination district. Then, a subset of sites, say M, that are candidates for migrating into D_t is selected according to the following rule: site $i \notin D_t$ is a candidate for migrating into D_t if $d_{it}^k = \min\{d_{is}^k : s = 1, \ldots, r\}$.

Finally, site i is chosen for migrating from D_q (the district it belongs to) to D_t if the following two conditions hold:

1. $d_{it}^k = \min\{d_{js}^k : j \in M, j \notin D_t\}$
2. $P_t^k < P_q^k$

The algorithm stops when there is no site i in M that satisfies conditions 1. and 2.

Properties of the strategies

General paradigm of a Voronoi Region Algorithm

WEIGHTED VORONOI REGION ALGORITHM

INPUT: G=(V,E), r, p_i $\forall i \in V$, d_{ij} $\forall i, j \in V$

OUTPUT: a connected partition of G

1. Locate the set S of the r centers in G

2. $k = 0$;

3. Let $d_{is}^{o} = d_{is}$, $\forall i \in V - S$, $\forall s \in S$

4. Compute the discrete Voronoi regions w.r.t. $[d_{is}]$ (*initial district map*)

5. **repeat**

$k = k + 1$

update (using either (1) or (2) throughout) the distances d_{is}^{k}, $\forall i \in V - S$,

according to P_s^{k-1}, $\forall s \in S$

compute the subset M' of sites that are candidates for migrating
and perform the corresponding migrations

compute the discrete Voronoi regions w.r.t. $[d_{is}^{k}]$ (*current district map*)

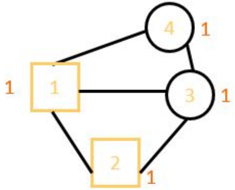
until M' is empty

Pathologies of the strategies

1. Lack of termination for the dynamic full transfer strategy (loop)

2. An example of lack of contiguity, where all the nodes have the same population and the site-to-center distances are given in the table.

distances		sites: 3	sites: 4
centers	1	1	2
centers	2	2	1



Voronoi regions $\{1,3\}$ and $\{2,4\}$ are perfectly balanced but $\{2,4\}$ is not contiguous!

Desirable properties of the strategies

1. Order invariance at each step of the algorithm, the order relation on the sites w.r.t. their distances to any given center s does not change:

$$d_{is}^k < d_{js}^k \Leftrightarrow d_{is} < d_{js} \qquad s \in S; i, j \in V \backslash S$$

2. Re-balancing at iteration k site i migrates from D_q to D_t only if $P_t^{k-1} < P_q^{k-1}$

3. Geodesic consistency: at any iteration, if site j belongs to district D_s and site i lies on the shortest path between j and s, then i also belongs to D_s.

4. Finite termination: the algorithm stops after a finite number of iterations.

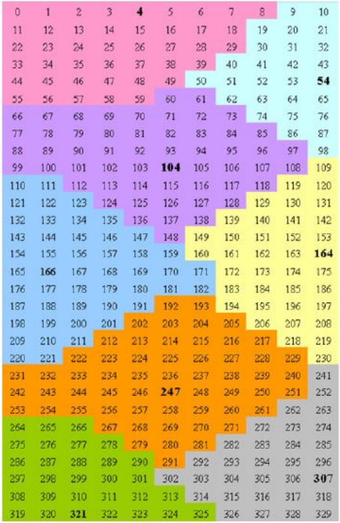
Properties of the strategies

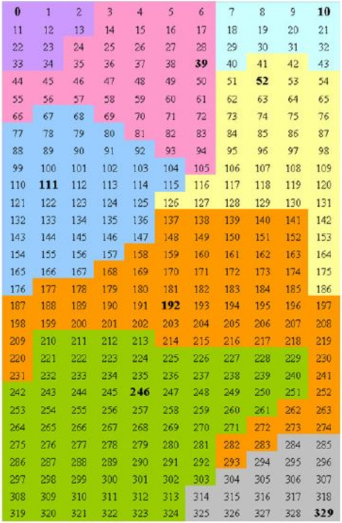
Properties of the Voronoi Region Approaches

Property	Static			Dynamic		
	Single transfer	Path transfer	Full transfer	Single transfer	Path transfer	Full transfer
Order invariance	Yes	Yes	Yes	Yes	Yes	Yes
Re-balancing	Yes	Yes	?	Yes	Yes	Yes
Geodesic consistency	Yes	Yes	Yes	Yes	Yes	Yes
Finite termination	Yes	Yes	?	Yes	Yes	No

Properties of the strategies

Different district maps obtained on a rectangular 30 11 grid graph according to different procedures for the location of the r centers.





Districting algorithms and criteria

PD model	Population equality	Compactness	Contiguity	Other criteria
Vickrey (1961)	*	*	*	*
Bodin (1973)	*	–	*	–
Arcese et al. (1992)	*	*	*	*
Hess et al. (1965)	**	*	**	–
Hojati (1996)	**	*	–	–
George et al. (1997)	**	*	–	–
Garfinkel and Nemhauser (1970)	*	*	*	–
Nygreen (1988)	*	*	*	*
Mehrotra et al. (1998)	*	*	*	–
Nemoto and Hotta (2003)	*	–	*	–
Li et al. (2007)	*	*	–	–
Bourjolly et al. (1981)	*	*	**	*
Ricca (1996)	*	*	*	*
Ricca and Simeone (2008)	*	*	*	*
Bozkaya et al. (2003)	*	*	*	*
Forman and Yue (2003)	**	*	*	–
Bação et al. (2005)	*	*	**	–
Yamada (2009)	*	–	*	–
Kalcsics et al. (2005)	*	*	*	–
Ricca et al. (2007, 2008)	*	*	*	–
Miller (2007)	–	*	*	–

Partitioning problems

I. Lari, J. Puerto, F. Ricca, A. Scozzari (2014) Partitioning a graph into connected components with fixed centers and optimizing different criteria, to be presented at the 20th Conference of the International Federation of Operational Research Societies (IFORS), Barcelona 13th-18th July 2014.

Problem definitions and notation

- Let $G = (V, E)$ be a connected graph with a set of n vertices V and a set of edges E. Suppose the subset $S \subset V$ is the set of $p = |S|$ fixed centers, which correspond to service points, while the subset $U = V \setminus S$ is the set of the np units/clients to be served.

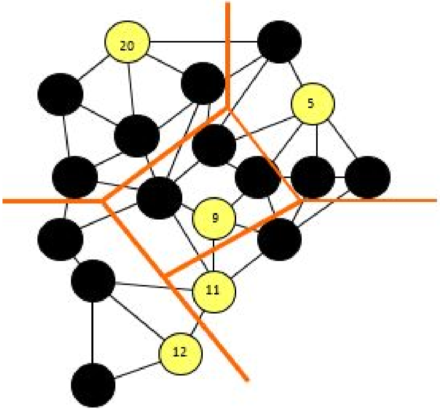
- We associate an assignment cost $c_{is} \geq 0$ to any pair $i \in U$, $s \in S$, and a weight $w_v \geq 0$ to each $v \in V$. In the general case such costs are assumed to be *flat*, i.e., they are independent from the topology of the network.

- A *p-centered partition* is a partition into $p = |S|$ connected components where each component contains exactly one center.

Partitioning problems

p-centered partition problem

find a p-centered partition of the graph optimizing a cost/weight based objective function.



Partitioning problems

We study several optimization problems in which we optimize different objective functions, based either on the costs c_{is}, or on the weights w_i, or on both of them.

1. The cost-based models are related to different cost-based objectives and optimization is aimed at minimizing such objectives.

2. The weight-based models concern with the problems of finding p-centered *uniform* and *most uniform* partitions (i.e., Population equality models). Actually, the objective of these problems is to have components of the partition as balanced as possible.

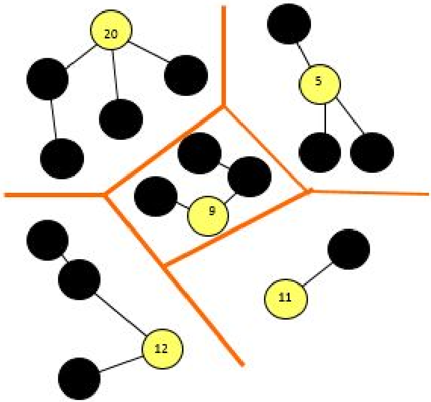
Objective functions

- *p-centered min-max* (pMM) partition problem: given a connected graph G, the sets $S, U \subset V$ and a cost function c, find a p-centered partition of G that minimizes the maximum assignment cost of a unit $i \in U$ to a center $s \in S$;
- *p-centered min-range* (pMR) partition problem: given a connected graph G, the sets $S, U \subset V$ and a cost function c, find a p-centered partition of G that minimizes the difference between the maximum and the minimum assignment cost of assigning a unit $i \in U$ to a center $s \in S$;
- *p-centered min-centdian* (pMCD) partition problem: given a connected graph G, the sets $S, U \subset V$ and a cost function c, find a p-centered partition of G that minimizes a convex combination of the maximum and the average cost of assigning a unit $i \in U$ to a center $s \in S$.

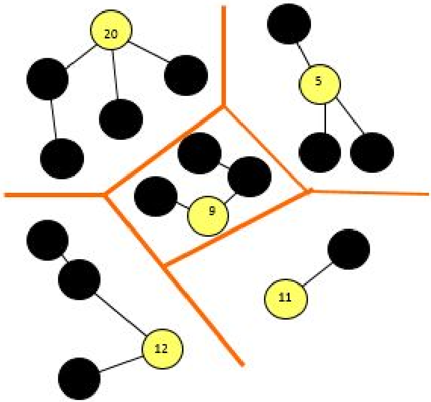
Objective functions

- *Capacitated p-centered min-sum* (C-pMS) partition problem: given a connected graph G, the sets $S, U \subset V$, a cost function c, a capacity $k_s \geq 0$ for each $s \in S$ and weights w_v, $v \in U$, find a p-centered partition of G that minimizes the total assignment cost and such that the total weight of a component centered in s does not exceed the capacity k_s of s.
- *p-centered uniform* (pU) partition problems: given a connected graph G, the sets $S, U \subset V$ and a cost function c, find a p-centered partition of G that: (i) minimizes the maximum assignment cost of a component (where the cost of a component is given by the sum of all the assignment costs of its units to the center of the component); (ii) maximizes the minimum assignment cost of a component; (iii) minimizes the difference between the maximum and the minimum cost of a component.

Problem formulation

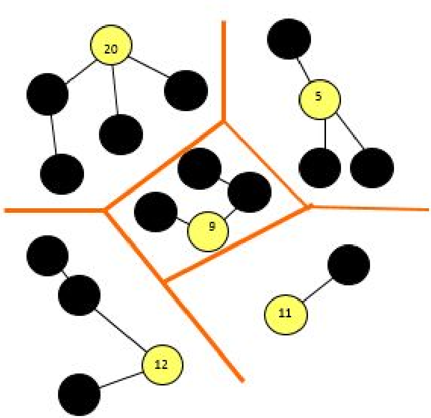


Problem formulation



Each component is a minimally connected component (*compactness*) and is a Tree. The set of trees forms a *Spanning Forest F* of G.

Problem formulation

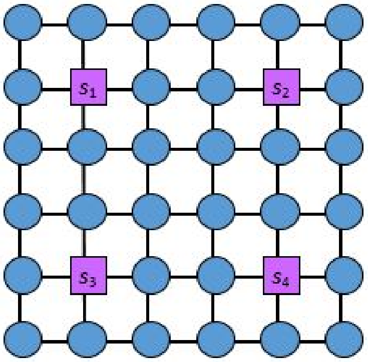


The partitioning problem can be stated as follows:
Find a spanning forest F of G such that each tree in F contains exactly one center and the (given) objective function is minimized.

Properties

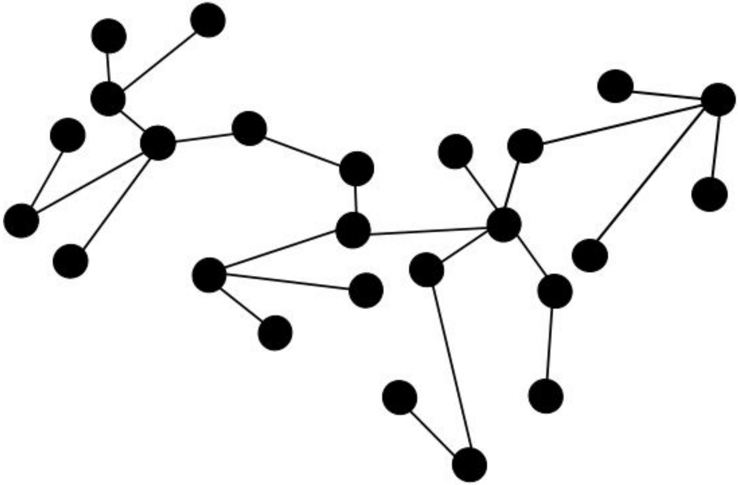
On general graphs G, the partitioning problem falls into the class of considerably difficult problems (NP-hard problems).

This negative result holds also if we consider special classes of graphs such as the class of bipartite graphs.



Partitioning problems on Trees

When the graph is a tree $T = (V, E)$ the problem is polynomially solvable



Partitioning problems on Trees

The algorithms we propose follow the approach of reducing the given tree T to a set of subtrees $T_1, \ldots, T_j, \ldots, T_q$ such that:

(1) the union of all the T_j's is equal to the whole tree T;

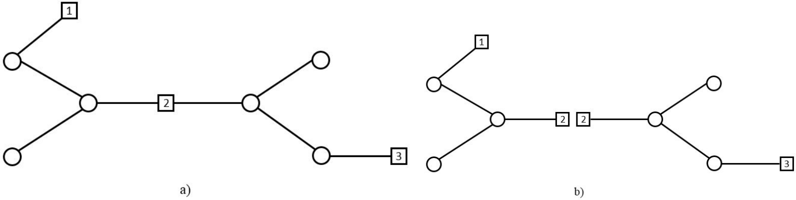
(2) any two subtrees T_k and T_j, $k \neq j$, intersect in at most one node, this node being a center;

(3) S_j is the set of leaves of T_j.

N. Apollonio, I. Lari, J. Puerto, F. Ricca, B.Simeone (2008), Polynomial Algorithms for Partitioning a Tree into Single-Center Subtrees to Minimize Flat Service Costs, Networks, vol. 51, 78-89.

Partitioning problems on Trees

Leaf property



This property allows the problem on T to be reduced, preserving optimality, to a set of independent instances on $T_1, \ldots, T_j, \ldots, T_q$.

Partitioning problems on Trees: Formulation

Based on the *Leaf property*, we can solve the problem on a single tree T_j, and then repeat the algorithm for all the subtrees obtained after the decomposition.

Partitioning problems on Trees: Formulation

Based on the *Leaf property*, we can solve the problem on a single tree T_j, and then repeat the algorithm for all the subtrees obtained after the decomposition.

Introduce the following binary variables:

$$y_{is} = \begin{cases} 1, & \text{if unit } i \text{ is assigned to center } s \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Partitioning problems on Trees: Formulation

$$
\begin{aligned}
\min \quad & f(y_{is}) \\
& y_{is} \leq y_{j(i,s)s} \qquad \forall\, i \in U,\ s \in S,\ (i,s) \notin E \\
& \sum_{s \in S} y_{is} = 1 \qquad \forall\, i \in U \\
& y_{is} \in \{0,1\} \qquad \forall\, i \in U,\ s \in S.
\end{aligned}
\tag{5}
$$

where we denote by $j(i,s)$ the vertex j that is adjacent to i in the (unique) path from i to s in T. Thus, to guarantee that the components of the p-centered partition are connected, for each pair $i \in U$ and $s \in S$ such that $(i,s) \notin E$, we impose that the vertex $j(i,s)$ is assigned to s whenever $y_{is} = 1$.

Partitioning problems on Trees: Formulation

Replace the integrality constraints on the y variables, thus obtaining

$$\begin{aligned}
\min \quad & f(y_{is}) \\
& y_{is} \leq y_{j(i,s)s} \qquad \forall\, i \in U,\ s \in S,\ (i,s) \notin E \\
& \sum_{s \in S} y_{is} = 1 \qquad \forall\, i \in U \\
& y_{is} \geq 0 \qquad \forall\, i \in U,\ s \in S.
\end{aligned} \tag{6}$$

Partitioning problems on Trees: Formulation

Write the model in a more compact form as:

$$\begin{aligned} \min \quad & f(y_{is}) \\ & y_{is} \in \mathbf{Q} \end{aligned} \tag{7}$$

$\mathbf{Q}$ is the set of feasible solutions of the above problem and it is integral, that is, all the vertices of the polytope representing (geometrically) the set of feasible solutions are integers.

Partitioning problems on Trees: Formulation

This allows the problems to be solved by Linear Programming with time complexity polynomial in the problem dimension.

Example

Consider the *p-centered min-max* partition problem, which is

$$\begin{aligned} \min \quad & \max_{s \in S} \max_{i \in U} \; c_{is} y_{is} \\ & y_{is} \in \mathbf{Q} \end{aligned} \tag{8}$$

Partitioning problems on Trees: Formulation

To solve the above problem we perform a binary search over all the possible values for the maximum of the objective function $\max_{s \in S} \max_{i \in U} c_{is} y_{is}$, and for each such value, say α, we solve a feasibility problem.

Actually, for a given α, the feasibility problem consists of finding a vector y that satisfies the following constraints

$$\begin{aligned} & y \in \mathbf{Q} \\ & y_{is} = 0 \qquad \text{if } c_{is} > \alpha \quad i \in U, \; s \in S. \end{aligned} \tag{9}$$

Partitioning problems on Trees: Formulation

The feasibility problem can be also solved by Linear Programming with time complexity polynomial in the problem dimension. The resulting algorithm for solving the above problem is the following

Algorithm 1

1. Sort the c_{is} values, $i \in U$, $s \in S$, in non-decreasing order
   1.1 Apply a binary search to generate all the possible different values α for the objective function of the problem
   1.2 for each α solve the feasibility problem

Partitioning problems on Trees

Capacitated *p-centered min-sum* (C-pMS) partition problem: given a connected graph G, the sets $S, U \subset V$, a cost function c, a capacity $k_s \geq 0$ for each $s \in S$ and weights w_v, $v \in U$, find a p-centered partition of G that minimizes the total assignment cost and such that the total weight of a component centered in s does not exceed the capacity k_s of s.

Theorem

The capacitated p-centered min-sum problem C-pMS is NP-complete on tree graphs.

Hint: Reduction from the 0-1 Knapsack Problem.