SPATIAL DISTRIBUTIONS: DENSITY-EQUALIZING MAP PROJECTIONS, FACILITY LOCATION, AND TWO-DIMENSIONAL NETWORKS

by

Michael T. Gastner

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Physics)
in The University of Michigan
2005

Doctoral Committee:

Associate Professor Mark E. Newman, Chairperson
Professor James W. Allen
Professor Leonard M. Sander
Associate Professor Cagliyan Kurdak
Associate Professor Dragomir R. Radev

"As a young man, my fondest dream was to become a geographer. However, while working in the customs office I thought deeply about the matter and concluded it was far too difficult a subject. With some reluctance, I then turned to physics as a substitute."

- Attributed to Albert Einstein in at least one refereed journal [1].

(According to Matt Rosenberg, "this quote was actually written by Duane F. Marble, Professor of Geography at Ohio State University. Professor Marble wrote the quote and put it on his office door at SUNY at Buffalo in response to the cool welcome received by the geography department which had taken over part of the physics building." Professor Marble confirmed this anecdote in a personal communication with the author.)

DEDICATION

To my parents

ACKNOWLEDGEMENTS

I would like to thank my adviser Dr. Mark Newman for his guidance. His quick wit and sense for interesting research projects are truly remarkable. Mark is a great source of inspiration with his ability to approach problems of practical relevance from unusual points of view. I also thank Dr. Leonard Sander, Dr. James Allen, Dr. Cagliyan Kurdak, and Dr. Dragomir Radev for serving on my dissertation committee.

I would further like to thank Elizabeth Leicht for helping me with the data on the geographic positions of facilities used in Chapter III. The staff of the University of Michigan's Numeric and Spatial Data Services also provided valuable help by patiently answering my questions about GIS software. Thanks also to Charles VanBoven and Matthew Golobish for proofreading the manuscript. This work was funded in part by the National Science Foundation under grant numbers DMS-0234188 and DMS-0405348 and by the James S. McDonnell Foundation. I gratefully acknowledge further financial support from the Max Kade Foundation, the Horace H. Rackham School of Graduate Studies, and the University of Michigan's Physics Department.

A collective thanks to the members of Renaissance Coop for providing a friendly environment during my time in Ann Arbor, and especially our chef Lynn Noellert.

Finally, I would like to thank my parents for their love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

SPATIAL DISTRIBUTIONS: DENSITY-EQUALIZING MAP PROJECTIONS, FACILITY LOCATION, AND TWO-DIMENSIONAL NETWORKS

by

Michael T. Gastner

Chairperson: Mark E. Newman

Geography is a strong force behind data of statistical, economical and technological interest. The distribution of human population and the location of industries, for example, follow non-trivial geographic patterns. Service facilities such as department stores and hospitals are constrained by geography since they must be in geographic proximity to consumers and patients. And in transportation networks, geography is important since nearby places can be more easily connected than those far apart.

We use tools from statistical physics to analyze the influence of geography on several social and technological phenomena. Variations in population density often go a long way in explaining geographic data. This can be visualized with "cartograms," maps in which the sizes of geographic regions such as countries or provinces appear in proportion to their population. The challenge in creating cartograms is to scale regions and still have them fit together. Here we present a new technique based on linear diffusion that is conceptually simple and produces useful and easily readable maps. A number of applications, including the results of the 2004 US presidential

election, illustrate the technique.

Many geographic problems of practical interest involve the optimization of point locations. Here we treat in detail a problem related to the distribution of service facilities, the p-median problem. It consists of finding the positions of p facilities in geographic space such that the mean distance between a member of the population and the nearest facility is minimal. An algorithm for approximate numerical solutions and analytical results are presented.

The remainder of this dissertation concentrates on networks whose vertices have definite geographic positions. This includes transportation systems, utility networks, or the Internet. We focus on the cost of a network, as represented by the total length of all its links, and its efficiency in terms of the directness of routes from point to point. Although minimizing the cost and maximizing the efficiency are often conflicting goals, real distribution networks achieve remarkably good compromises. Models and numerical simulations are presented to explain this observation. We close with an analysis of structure and flow in optimally designed spatial networks.

CHAPTER I

INTRODUCTION

In the past few years, the scientific community has realized the relevance of statistical physics for many disciplines outside of physics. Exciting new results have emerged by applying concepts and methods of statistical physics to fields as different as evolutionary biology, economics, and sociology. In this dissertation, we hope to contribute to this young field of interdisciplinary work by exploring several topics linking statistical physics with geography.

We are by no means the first to draw parallels between both sciences. In fact, physics metaphors were common practice during the quantitative geography movement of the 1960s and early 1970s. One of the central works of that period, Bunge's "Theoretical geography" [14], has many references to physics. He describes a gravity theory of human settlement, an electric circuit theory of movement in transportation networks, a heat flow model of human migration, a fluid mechanics analogy for traffic flows, and a kinetic gas theory for the geographic spread of ideas. Beyond Bunge's direct comparisons to physics however, this general zeitgeist of the geographic literature of those days was characterized by a tendency towards abstraction, modeling, and mathematics—a typical approach for a physicist, but rather rare in the overall history of geography.

Imposing a physics attitude on other disciplines is not without its risks. "The most distinguishing feature of the physicists' character ... is the idea that everything that can be analysed quantitatively should be modelled (and if it cannot be analysed

it should be modelled anyway)" [15]. The quantitative geography community sometimes stepped into this same pitfall of producing "hot air" by making mathematical reasoning an end in itself. Some leading scholars, therefore, started rejecting quantitative geography in the 1970s: "The quantitative revolution has run its course and diminishing marginal returns are apparently setting in as ... [it] serve[s] to tell us less and less about anything of great relevance. ... There is a clear disparity between the sophisticated theoretical and methodological framework which we are using and our ability to say anything really meaningful about events as they unfold around us" [16].

This led not only to the decline of the quantitative movement, but also to a serious identity crisis in the geography community. The University of Michigan's geography department for example, once a hot spot of that movement, was disbanded in 1982, following a similar trend at other universities. On the positive side, this brief period established the idea that social geography can go beyond previous, mostly qualitative and case-based studies. The recent surge of complex systems research, triggered by the ability to perform large-scale numerical simulations has, mostly unwittingly, taken up the torch with its work on land use patterns [17]. In our opinion, much can still be gained by returning to some of the questions posed by the earlier pioneers of quantitative geography.

In Chapter II we will take a closer look at what might be considered the paradigmatic problem of that era [18], the construction of "cartograms", i.e. maps in which the sizes of geographic regions such as countries or provinces appear in proportion to their population. Such maps are helpful for the representation of census results, election returns, disease incidence, and many other kinds of human data. Unfortunately, in order to scale regions and still have them fit together, one is normally forced to distort the regions' shapes, potentially resulting in maps that are difficult to read. Here we will introduce a technique for making cartograms based on ideas borrowed from elementary physics that is conceptually simple and produces easily readable

maps. We will illustrate this method with applications to disease and homicide cases, Autonomous Systems in the Internet, the 2004 US presidential election, energy consumption and production in the United States, and the geographic distribution of stories appearing in the news.

While most data in Chapter II are of a probabilistic nature, Chapter III will concentrate on data obtained by optimization. The particular problem we will study there is related to the distribution of service facilities such as hospitals or post offices. We will try to find the positions of p facilities in geographic space such that the mean distance between a member of the population and the facility closest to him or her is minimal. Solving this so-called p-median problem is very difficult; no efficient method is currently known that can solve it exactly. We will present an approximate numerical algorithm as well as approximate analytical calculations, and will show how these results relate to the spatial distribution of real facilities.

We will then move on to facilities which are not simply isolated points in space, but have connections to other facilities in the form of a network. Chapter IV will consider the case of a growing distribution or collection network with one "root node," a facility that acts as a source or sink of the commodity distributed. Examples include pipelines delivering oil from a well to neighboring towns, or urban transit networks transporting passengers to a central train station. In particular, we will focus on the cost of a network, as represented by the total length of all its connections, and its efficiency in terms of the directness of routes from point to point. Using data from several real-world examples, we will find that distribution networks appear remarkably close to optimal where both these properties are concerned. We will propose two models of network growth that offer explanations of how this situation might arise. Similarities with fractal growth models in the physics literature will be discussed.

Not all networks, however, have a single source or sink. In Chapter V we will investigate the case where many facilities contribute to the flow. Similar to Chap. IV,

the efficiency of such networks depends on their total length as well as the minimal distances between all facilities. These two criteria are generally at odds with one another: A network with few and short connections will not provide many direct links from point to point, while a network with all possible links is usually prohibitively expensive to construct. Finding an optimal solution is difficult, so we will introduce a heuristic Monte Carlo optimization scheme. We will analyze near-optimal networks for different user preferences and distance metrics in terms of their efficiency and flow distribution. With our algorithm to solve the p-median problem from Chap. III, we can create optimized networks for realistic population distributions, a technique with potential use in real-life network design problems.

A brief summary of our results in Chapter VI will conclude this dissertation.

CHAPTER II

DENSITY-EQUALIZING MAP PROJECTIONS: DIFFUSION-BASED ALGORITHM AND APPLICATIONS

2.1 Introduction

Suppose we wish to represent on a map some data concerning, to take the most common example, the human population. For instance, we might wish to show votes in an election, incidence of a disease, number of cars, televisions, or phones in use, numbers of people falling in one group or another of the population, by age or income, or any other variable of statistical, medical, or demographic interest. The typical course under such circumstances would be to choose one of the standard (near-)equal-area projections for the region of interest and plot the data on it either as individual data points or using some sort of color code. The interpretation of such maps however can be problematic. A plot of disease cases, for instance, will inevitably show high incidence in cities and low incidence in rural areas, solely because there is a higher density of people living in cities. Fig. 2.1(a), for example, shows the distribution of lung cancer cases among males in the state of New York between 1993 and 1997. Cases are particularly dense in New York City and its suburbs and sparse in the rural north of the state, but this could be purely a population effect and nothing to do with the disease itself.

To get a clearer impression of the situation we can instead plot a fractional measure of disease incidence rather than raw incidence data; we plot some measure of the

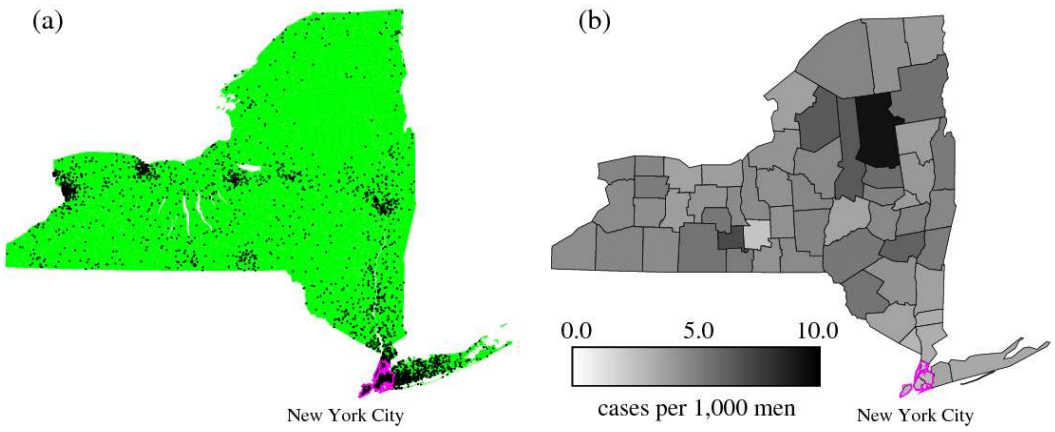

Figure 2.1: Lung cancer cases among males in the state of New York, 1993-1997. (a) Each dot represents 10 cases, randomly placed within the zip-code area of occurrence. Since there are 1598 zip codes in the state, each case can be located quite accurately. (Data from the New York State Department of Health.) (b) Incidence rates by county.

number of cases per capita, binned in segments small enough to give good spatial resolution but large enough to give reliable sampling. Then we can use a color code to indicate different per capita rates on the map, as in Fig. 2.1(b). This procedure however has its own problems since it discards all information about where most of the cases are occurring. In Fig. 2.1(b), for example, there is now no way to tell that a large fraction of all cases occur in the New York City area.

Ideally, we would like some geographic representation of the data that allows us to see simultaneously where each individual case occurs as well as the per capita incidence. Though it appears at first that these two goals are irreconcilable, this is not in fact the case. On a normal area-preserving or approximately area-preserving projection, such as that used in Fig. (2.1), they are indeed irreconcilable. But if we can construct a projection in which areas on the map are proportional not to areas on the ground but instead to human population, then we can have our cake and eat it. Disease cases or other similar data plotted on such a projection will have the same density in areas with equal per capita incidence regardless of the population, since both the raw numbers of cases and the area will scale with the population.

However, each case or group of cases can still be represented individually on such a map, so it will be clearly visible where most of the cases occur. Projections of this kind are known as value-by-area maps, density equalizing maps, anamorphoses, or cartograms [19, 3, 20, 21].

The construction of cartograms turns out to be a challenging undertaking. A variety of methods have been put forward, but none of them have been entirely satisfactory. In particular, they often produce highly distorted maps that are difficult to read or projections that are badly behaved under some circumstances, with overlapping regions or strong dependence on coordinate axes. In many cases the methods proposed are also computationally demanding, sometimes taking hours to produce a single map. Here we describe a new method based on the physics of diffusion which is, we believe, intuitive, while also producing elegant, well-behaved, and useful cartograms whose calculation makes relatively low demands on our computational resources.

2.2 Literature review

Cartograms have been used by geographers for about one century. Early examples, like that in Fig. 2.2, had to be prepared laboriously by hand for each individual map. Several methods have been proposed since then that can be implemented as computer programs. Before deciding for any particular technique, the user should clearly define his objective. If, for example, one is willing to live with a non-contiguous cartogram, one in which regions adjacent in real life are not adjacent on the cartogram, then several quite simple methods give good results, such as Dorling's circular cartograms [3] (Fig. 2.3). If, on the other hand, the rescaled regions are supposed to fit together correctly, one is forced to use more involved strategies.

Some of these strategies begin by breaking down the map into a finite set of non-intersecting polygons and vertices which describe the boundaries of interest. Then

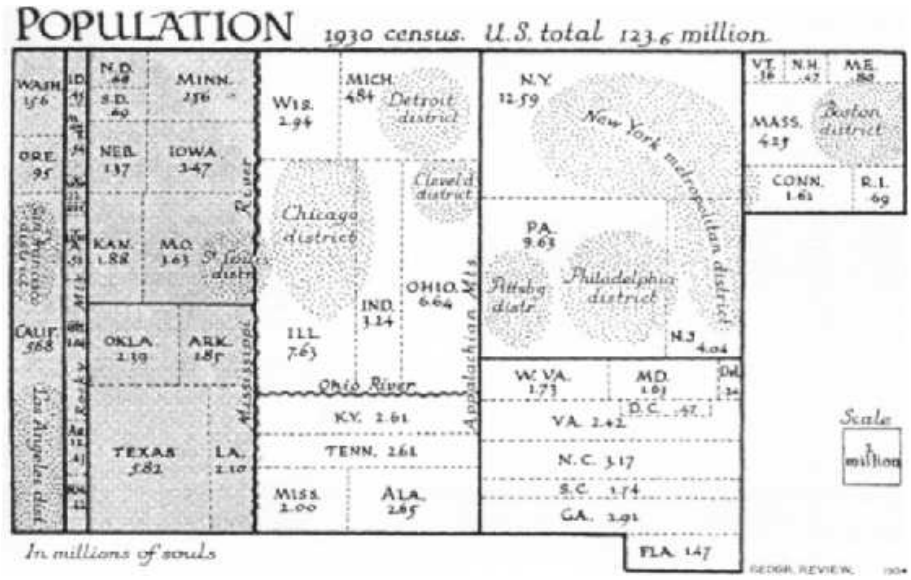

Figure 2.2: Rectangular cartogram by Raisz (1934) [2]. States appear in proportion to their population in 1930. Note that some states adjacent in reality are not adjacent on this cartogram and vice versa.

the vertices are moved to new positions such that the polygon areas are proportional to the population inside them, but the original topology is left intact, i.e. polygons that were neighbors on the original map remain neighbors on the cartogram. This can, of course, only be accomplished by distorting the polygon's shapes. To keep the shapes more or less similar to the original ones, certain features of the polygons, such as angles or length ratios, should be preserved.

Kocmoud [4] proposed a mass-and-spring model in which some of the springs attempt to change the areas, while other springs try to preserve the shapes. The springs, however, do not follow the equations of motion for simple harmonic oscillators because then the cartogram would not settle down in a stationary state, but oscillate about the equilibrium position. To prevent this, Kocmoud changes the equations of motion so that they contain only first, but not second derivatives with respect to time.[1] The computation then repeatedly alternates between changing the polygon

[1]The differential equation at the center of our method below, Eq. (2.7), is similarly first-order in

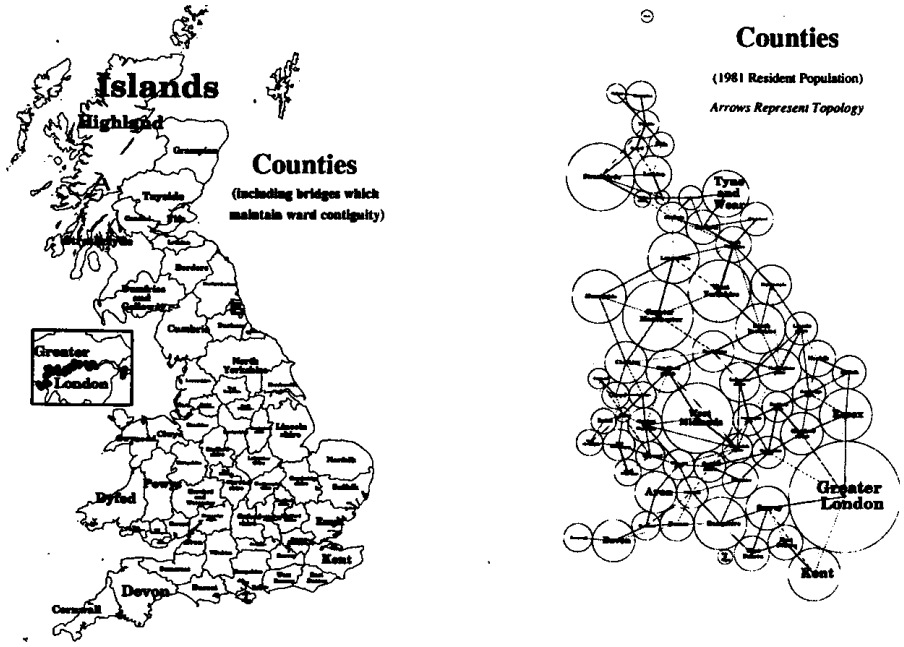

Figure 2.3: Map and circular population cartogram of Great Britain. Reproduced with the author's permission from [3].

areas and restoring their shapes. To reduce the complexity of the problem, the polygons are initially simplified to contain fewer vertices; the missing vertices are inserted again only at later stages of the calculation to form the final cartogram. The results are easily readable cartograms (Fig. 2.4(a)), but this method appears to be rather slow: Kocmoud reports a run-time of 16 hours.

Keim *et al.* [5] achieve considerably shorter run-times, 25 seconds, with a similar vertex relocation technique. Their "CartoDraw" algorithm is faster because it divides the input map into small sections, and only vertices within one section are moved during one iteration whereas Kocmoud's method considers all vertices simultaneously. Some of the speed-up, however, seems to be due to a substantial simplification of the input polygons before beginning the computations which unfortunately dispenses with many useful cartographic details. The missing vertices are not, as in Kocmoud's scheme, put back later during the calculation which gives the "CartoDraw" results a

time.

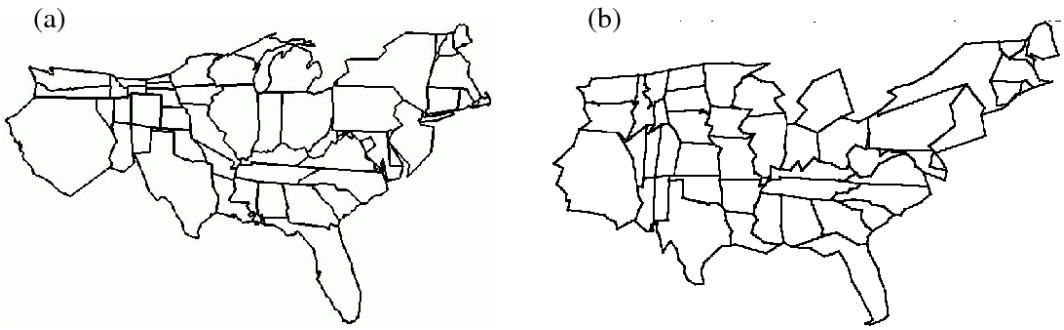

Figure 2.4: US population cartogram, (a) by Kocmoud [4], (b) by Keim *et al.* [5] (©2004 IEEE).

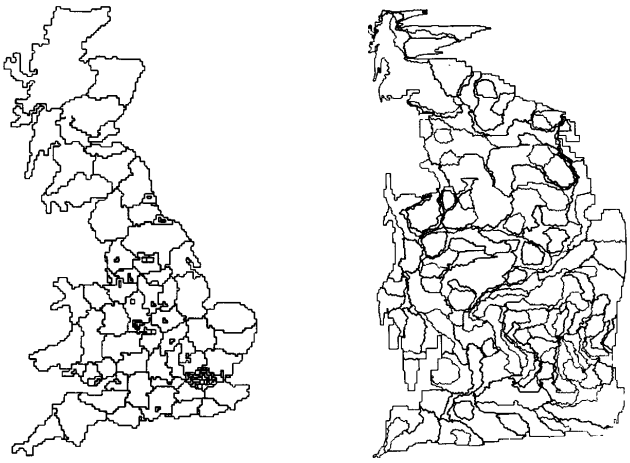

Figure 2.5: Population cartogram of Britain by county. Left: the original map. Right: cartogram generated using the cellular automaton algorithm of Dorling. Reproduced with the author's permission from Dorling [3].

less polished appearance (Fig. 2.4(b)).

Not all methods operate directly on the polygon vertices. Appel *et al.* [22] and recently Dorling [3] have both proposed methods based on cellular automata that draw the initial map on a fine grid. In their algorithms, for each iteration, cells lying on or close to the boundaries of regions are identified, and if a neighboring region needs extra area, then those cells are reassigned to the neighbor. The procedure is repeated and the regions with greatest population grow slowly larger until an equilibrium is reached and no further changes are needed. The procedure is elegant and simple, but in practice it can distort shapes quite badly—see Fig. 2.5. One can add constraints

on the shapes to make the maps more readable, but then the method quickly loses its main advantage, namely its simplicity.

The techniques we have reviewed so far only determine the boundaries, but not the positions of any other points on the map. Other techniques construct a continuous "displacement field" which gives the displacements needed for every point on the map to form a density equalizing projection. Mathematically, the displacements form a transformation $\mathbf{r} \to \mathbf{T}(\mathbf{r})$ of one plane to another plane such that the Jacobian $\partial(T_x, T_y)/\partial(x, y)$ of the transformation is proportional to some specified population density $\rho(\mathbf{r})$ thus:

$$\frac{\partial(T_x, T_y)}{\partial(x, y)} \equiv \frac{\partial T_x}{\partial x}\,\frac{\partial T_y}{\partial y} - \frac{\partial T_x}{\partial y}\,\frac{\partial T_y}{\partial x} = \frac{\rho(\mathbf{r})}{\bar{\rho}}. \tag{2.1}$$

where $\bar{\rho}$ is the mean population density averaged over the area to be mapped. (This choice of normalization for the Jacobian ensures that the total area before and after the transformation is the same.)

Eq. (2.1) does not determine the cartogram projection uniquely. To do that we need one more constraint; two constraints are needed to fix the projection for a two-dimensional cartogram. Different choices of the second constraint give different projections, and no single choice appears to be the obvious candidate. One idea is to demand conformal invariance under the cartogram transformation, i.e., to demand that angles be preserved locally. This requirement is equivalent to demanding that the Cauchy–Riemann equations be satisfied:

$$\frac{\partial T_x}{\partial x} = \frac{\partial T_y}{\partial y},\ \frac{\partial T_x}{\partial y} = -\frac{\partial T_y}{\partial x}, \tag{2.2}$$

but this imposes two, not one, additional constraints, and hence it is normally not possible to construct a conformally invariant cartogram.

In an attempt to minimize the distortion of angles, Tobler [18, 6] took the first steps in the automated computer generation of cartograms in the late 1960s (Fig. 2.6(a)).

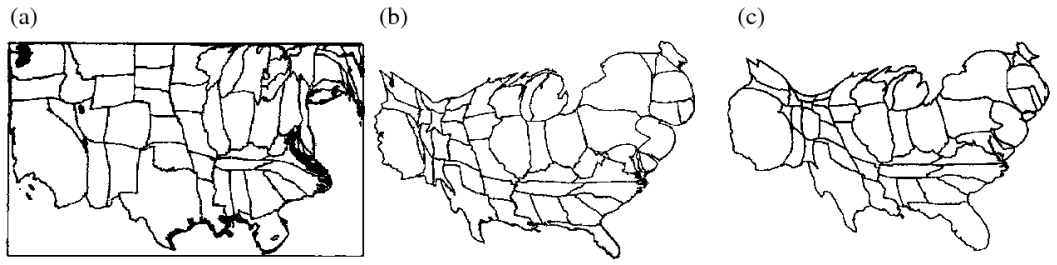

Figure 2.6: US Population cartograms (a) by Tobler [6] (©1973 New York Academy of Sciences, U.S.A.), (b) by Dougenik *et al.*(reproduced with permission from [7]), (c) by Gusein-Zade and Tikunov [8] (reproduced with permission from [8]).

The initial map is divided into small rectangular or hexagonal cells, each of which is then independently dilated or shrunk to a size proportional to its population content. Since each cell is scaled separately, the corners of adjacent cells do not match afterwards. To re-establish a match, Tobler's method takes a vector average over the positions of corresponding corners and draws a new map with the resulting distorted cells. The process is then iterated until a fixed point of the transformation is reached. Although the principle is simple, it runs into practical problems. First, convergence tends to be rather slow because a node a few cells away from a population center will feel the effect of that center only after several iterations. Second, under some circumstances the transformation can produce overlapping or "folded" regions of the map, thereby ruining the topology. This problem can be corrected by introducing additional constraints, but the result is a more complex algorithm with slower run-times.

To increase the speed of the calculations, Dougenik *et al.* [7] introduced an algorithm where the borders of a cell move in response not only to local space requirements but also to "forces" exerted by other cells. Cells create force fields that diminish with distance from the cell and that are larger for cells that contain larger populations. These forces "push" other cells away from areas of high population in a manner reminiscent of the behavior of charged objects in electrostatics (although the authors do not use this metaphor). Again the positions are relaxed iteratively to achieve the final

cartogram, and convergence is substantially faster than Tobler's algorithm, although topological errors still cannot be ruled out.

Gusein-Zade and Tikunov [8] suggested a further twist that does away with the cells altogether by taking the limit of infinitely small cell size. The displacements can then be expressed as integrals over the mapped area such that regions with high population densities exert a repulsive force and those with lower densities an attractive force:

$$\mathbf{T}(\mathbf{r}) - \mathbf{r} = \int_{\text{map}} \frac{\rho(\mathbf{r}') - \bar{\rho}}{\bar{\rho}} \frac{\mathbf{r} - \mathbf{r}'}{2\pi(\mathbf{r} - \mathbf{r}')^2} \, \mathrm{d}^2 r'. \tag{2.3}$$

This integral is akin to Coulomb's law in two dimensions and we can apply the usual tools of vector analysis to it. In particular, if the density $\rho(\mathbf{r})$ is a piecewise constant function, Stokes theorem allows us to rewrite the expression above as a sum of line integrals that can be analytically solved. The method, though somewhat arcane, produces some of the most attractive cartograms among the existing algorithms—see Fig. 2.6(c).

Several other mathematical techniques have been proposed [23, 24, 25, 26] as well as some experimental methods, using mechanical or electrostatic devices [27, 28, 29]. The method we will describe in the next section, although also based on physics, has the advantage that it can be easily and quickly simulated on a computer.

2.3 The diffusion cartogram

It is a trivial observation that on a true population cartogram the population is necessarily uniform: once the areas of regions have been scaled to be proportional to their population then, by definition, population *density* is the same everywhere. Thus, one way to create a cartogram given a particular population density is to allow population somehow to "flow away" from high-density areas into low-density ones until the density is equalized everywhere. This immediately brings to mind the diffusion process of elementary physics, and in fact it is not difficult to show that

simple Fickian (or linear) diffusion achieves just such a density equalization. This is the basis for our method of constructing a cartogram.

We describe the population by a density function $\rho(\mathbf{r}, t)$, where $\mathbf{r}$ represents geographic position and t time. At $t = 0$ the population density is the one observed in reality, and then we allow this density to diffuse. The current density, measuring the amount and direction of flow, is given by

$$\mathbf{J} = \mathbf{v}(\mathbf{r}, t)\, \rho(\mathbf{r}, t), \tag{2.4}$$

where $\mathbf{v}(\mathbf{r}, t)$ and $\rho(\mathbf{r}, t)$ are the velocity and density respectively at position $\mathbf{r}$ and time t. In Fickian diffusion the current density follows the gradient of the density field thus:

$$\mathbf{J} = -\nabla\rho, \tag{2.5}$$

meaning that the flow is always directed from regions of high density to regions of low density and will be faster when the gradient is steeper. Conventionally, there is a diffusion constant in Equation (2.5) that sets the time scale of the diffusion process. But since we are only interested in the limit $t \to \infty$ we can set this diffusion constant equal to one without loss of generality.

The diffusing population is conserved locally so that

$$\nabla \cdot \mathbf{J} + \frac{\partial\rho}{\partial t} = 0. \tag{2.6}$$

Combining Eqs. (2.4), (2.5), and (2.6) we then arrive at the familiar diffusion equation:

$$\nabla^2\rho - \frac{\partial\rho}{\partial t} = 0 \tag{2.7}$$

and the expression for the velocity field in terms of the population density:

$$\mathbf{v}(\mathbf{r}, t) = -\frac{\nabla\rho}{\rho}. \tag{2.8}$$

The calculation of the cartogram involves solving Eq. (2.7) for $\rho(\mathbf{r}, t)$ starting from

the initial condition in which ρ is equal to the given population density of the region of interest and then calculating the corresponding velocity field from Eq. (2.8). The cumulative displacement $\mathbf{r}(t)$ of any point on the map at time t can be calculated by integrating the velocity field thus:

$$\mathbf{r}(t) = \mathbf{r}(0) + \int_0^t \mathbf{v}(\mathbf{r}, t')\, \mathrm{d}t'. \tag{2.9}$$

In the limit $t \to \infty$ the set of such displacements for all points on the original map defines the cartogram.

Most of the time, we are not interested in mapping the entire globe, but only some part of it, which means that the area of interest will have boundaries (e.g., country borders or coastlines) beyond which we don't know or don't care about the population density. It would be inappropriate to represent the regions outside these boundaries as having zero population, even if they are, like the ocean, unpopulated, since this would cause arbitrary expansion of the cartogram as the population diffused into its uninhabited surroundings. (This is true of essentially all methods for constructing cartograms.) Instead, therefore, we apply a "neutral buoyancy" condition, floating the area of interest in a "sea" of uniform population density equal to the mean density of the map as a whole. This keeps the total area under consideration constant during the diffusion process.

The whole system, including the sea, is then enclosed in a box.[2] For simplicity we consider only rectangular boxes, as most others have done also.[3] Provided the dimensions L_x and L_y of the box are substantially larger than the area to be mapped,

---

[2]Note that we do not fix the shape of the borders or coastlines in our cartogram, as others have occasionally done. Doing so can create bottlenecks in the diffusion flow, which we avoid by allowing free motion of all points, whether they are near a border or not. Fixing external boundaries in fact often produces worse cartograms because it requires more substantial distortions of internal boundaries.

[3]We discuss spherical cartograms briefly in Section 7.

the dimensions themselves do not matter. In the limit $L_x, L_y \to \infty$ the cartogram will be a unique deterministic mapping, independent of the coordinate system used, with no overlapping regions. In practice, we find that quite moderate system sizes are adequate—dimensions two to three times the linear extent of the area to be mapped appear to give good results. We also need to choose boundary conditions on the walls of the box. This choice too has no great effect on the results, provided the size of the box is reasonably generous. We use Neumann boundary conditions in which there is no flow of population through the walls of the box.

The considerations above completely specify our method and are intuitive and straightforward. The actual implementation of the method, if one wants a calculation that runs quickly, involves more work. We solve the diffusion equation in Fourier space, where it is diagonal, and back-transform before integrating over the velocity field. With the Neumann boundary conditions, the appropriate Fourier basis is the cosine basis, in which the solution to the diffusion equation has the form

$$\rho(\mathbf{r},t) = \frac{4}{L_x L_y} \sum_{\mathbf{k}} \tilde{\rho}(\mathbf{k}) \cos(k_x x) \cos(k_y y) \exp(-k^2 t), \tag{2.10}$$

where the sum is over all wave vectors $\mathbf{k} = (k_x, k_y) = 2\pi(m/L_x, n/L_y)$ with m, n non-negative integers, and $\tilde{\rho}(\mathbf{k})$ is the discrete cosine transform of $\rho(\mathbf{r}, t = 0)$:

$$\tilde{\rho}(\mathbf{k}) = \frac{1}{4} \left(\delta_{k_x,0} + 1\right) \left(\delta_{k_y,0} + 1\right) \int_0^{L_x} \int_0^{L_y} \rho(\mathbf{r}, 0) \cos(k_x x) \cos(k_y y) \, \mathrm{d}x \, \mathrm{d}y, \tag{2.11}$$

where $\delta_{i,j}$ is the Kronecker symbol. The velocity field $\mathbf{v}$ is then easily calculated from Eqs. (2.8) and (2.10) and has components

$$v_x(\mathbf{r},t) = \frac{\sum_{\mathbf{k}} k_x \tilde{\rho}(\mathbf{k}) \sin(k_x x) \cos(k_y y) \exp(-k^2 t)}{\sum_{\mathbf{k}} \tilde{\rho}(\mathbf{k}) \cos(k_x x) \cos(k_y y) \exp(-k^2 t)}, \tag{2.12a}$$

$$v_y(\mathbf{r},t) = \frac{\sum_{\mathbf{k}} k_y \tilde{\rho}(\mathbf{k}) \cos(k_x x) \sin(k_y y) \exp(-k^2 t)}{\sum_{\mathbf{k}} \tilde{\rho}(\mathbf{k}) \cos(k_x x) \cos(k_y y) \exp(-k^2 t)}. \tag{2.12b}$$

Equations (2.11) and (2.12) can be evaluated rapidly using the fast Fourier transform and its back-transform respectively, both of which in this case run in time of order

$L_x L_y \log(L_x L_y)$. We then use the resulting velocity field to integrate Eq. (2.9), which is a nonlinear Volterra equation of the second kind and can be solved numerically by standard methods [30]. In practice it is the Fourier transform that is the time-consuming step of the calculation and with the aid of the fast Fourier transform this step can be performed fast enough that the whole calculation runs to completion in a matter of seconds or at most minutes, even for large and detailed maps.

It is straightforward to see that our diffusion cartogram satisfies the fundamental definition, Eq. (2.1), of a cartogram. In the limit $t \to \infty$, Eq. (2.10) is dominated by the $\mathbf{k} = 0$ term and gives

$$\rho(\mathbf{r}, \infty) = \frac{4\tilde{\rho}(0)}{L_x L_y} = \frac{1}{L_x L_y} \int_0^{L_x} \int_0^{L_y} \rho(\mathbf{r}', 0)\, \mathrm{d}x'\, \mathrm{d}y' = \bar{\rho}, \tag{2.13}$$

where $\bar{\rho}$ is again the mean population density over the area mapped. Furthermore, by definition, the total population within any moving element of area does not change during the diffusion process, and hence, denoting by $\mathbf{T}(\mathbf{r})$ the final position of a point that starts at position $\mathbf{r}$, we have $\rho(\mathbf{r})\, \mathrm{d}x\, \mathrm{d}y = \bar{\rho}\, \mathrm{d}T_x\, \mathrm{d}T_y$, and, rearranging, the Jacobian is given by $\partial(T_x, T_y)/\partial(x, y) = \rho(\mathbf{r})/\bar{\rho}$, in agreement with Eq. (2.1).

Conceptually our algorithm is in some respects similar to the cellular automaton method of Dorling [3]. Our description of the diffusion method has been entirely in terms of macroscopic variables and equations, but one could equally look at the method as a microscopic diffusion process in which each individual member of the population performs a Gaussian random walk about the surface of the map. Over time the population will diffuse until it is uniform everywhere within the box enclosing the map, except for statistical fluctuations. The cartogram is derived by moving all boundaries on the map in such a way that the net flow passing through them is zero at all times during the diffusion process. This resembles Dorling's method in the sense that different regions trade their area until a fair distribution is reached.

Our method, however, has the advantage of being based on a global, lattice-

independent process. The exchange of area between regions in Dorling's method occurs only between nearest-neighbor squares along the principal axes of a square lattice and this introduces a strong signature of the lattice topology into the final cartogram (Fig. 2.5). Furthermore, the cellular automaton method gives only the displacements of region boundaries whereas our method gives the displacement of any point on the map. In this respect our algorithm is more like the methods by Tobler [6], Dougenik *et al.* [7], and Gusein-Zade and Tikunov [8].

2.4 Population density function

The description of our method tells, in a sense, only half the story of how to create a cartogram. Before applying this or indeed any method, we need to choose the starting density $\rho(\mathbf{r})$ for the map. We can, by defining $\rho(\mathbf{r})$ in different ways, control the properties of the resulting cartogram, including crucially the balance between accurate density equalization and readability.

Population density is not strictly a continuous function, since people are themselves discrete and not continuous. To make a continuous function the population must be binned or coarse-grained at some level. All methods of constructing cartograms require one to do this, and no single accepted standard approach exists. Part of the art of making a good cartogram lies in shrewd decisions about the definition of the population density.

If we choose a very fine level of coarse-graining for the population density, then the high populations in centers such as cities will require substantial local distortions of the map in order to equalize the density. A coarser population density will cause less distortion, resulting in a map with features that are easier to recognize, but will give a less accurate impression of the true population distribution. The most common choice made by others has been to coarse-grain the population at the level of the (usually political) regions of interest. For example, if one were interested in the United States,

one might take the population of each state and distribute it uniformly over the area occupied by that state. This method can be used also with our cartogram algorithm and we give some examples below. But we are not obliged to use it, and in some cases it may be undesirable, since binning at the level of states erases any details of population distribution below the state level.

On the other hand, if we use a finer resolution and allow for density variation within states then not only can the local distortions of the cartogram become severe, but rapidly varying densities can also slow down the numerical calculations. Regions with population density zero are particularly difficult to deal with because the denominator in Equation (2.8) becomes zero and hence the velocity is undefined. In our work we circumvent these issues by first adding a small constant to the density to get rid of zero values, and then applying a spatially uniform Gaussian blur to the population density thus:

$$\rho(\mathbf{r}) = \frac{1}{2\pi\sigma^2} \int_0^{L_x} \int_0^{L_y} \rho_{\text{raw}}(\mathbf{r}') \exp\left[-\frac{(\mathbf{r}' - \mathbf{r})^2}{2\sigma^2}\right] \mathrm{d}x' \, \mathrm{d}y', \tag{2.14}$$

where σ is the width of the Gaussian and ρ_{raw} is the unblurred density. Varying the width σ of the blurring function is a convenient way to tune the cartogram between accuracy and readability.

This blur can be performed rapidly in Fourier space, where the convolution becomes a simple multiplication. Alternatively, we note that Equation (2.14) is equal to the density we would get if we simply allowed ρ_{raw} to diffuse for a time $\frac{\sigma^2}{2}$, so that we can also use Equation (2.10) to calculate the Gaussian blur.

Ultimately the choice of population density function is up to the user of the method, who must decide what particular features are most desirable in his or her application. One advantage of our diffusion-based algorithm is that it is entirely agnostic about this choice; the process of computing the cartogram is decoupled from the calculation of the population density and, hence, is not slanted in favor of one

choice or another.

2.5 Applications I: Population cartograms and aggregation

We give several examples of the use of diffusion cartograms, focusing on the United States and using population data from the 2000 U.S. Census. First, we reexamine the New York lung cancer map, Fig. 2.1(a), by transforming it to population cartograms, applying Gaussian blurs of varying width to the underlying population density.[4] In Fig. 2.7(a), we show the cancer distribution on a population cartogram with only moderate coarse-graining. Although the map is visibly distorted, a reader familiar with the state of New York would still be able to identify different regions because of the shape of the state boundaries. The distribution of cancer cases however is still visibly "clumped."

In Fig. 2.1(b) we use a much finer Gaussian blur, creating a cartogram with better population equalization and significantly greater distortion. Now the virtue of this representation becomes strikingly clear. As the figure shows, when we use a projection that truly equalizes the population density over the map, there is no longer any obvious variation in the distribution of cases over the state—the pattern appears random with more or less the same density everywhere. The shape of the map in Fig. 2.1(b) does not much resemble the shape of the original any more, but this is the price we pay for equalizing the population almost perfectly.

We do not need to rely on eyesight for judging whether the distribution of points in Fig. 2.1(b) is clustered. There are several statistical methods to test spatial distributions for randomness. A particularly simple measure is the Hopkins statistic H, a number between 0 and 1 whose expectation value for random point patterns is $\frac{1}{2}$.

[4] A similar study with a different technique and for a smaller area was carried out by Merrill [31]

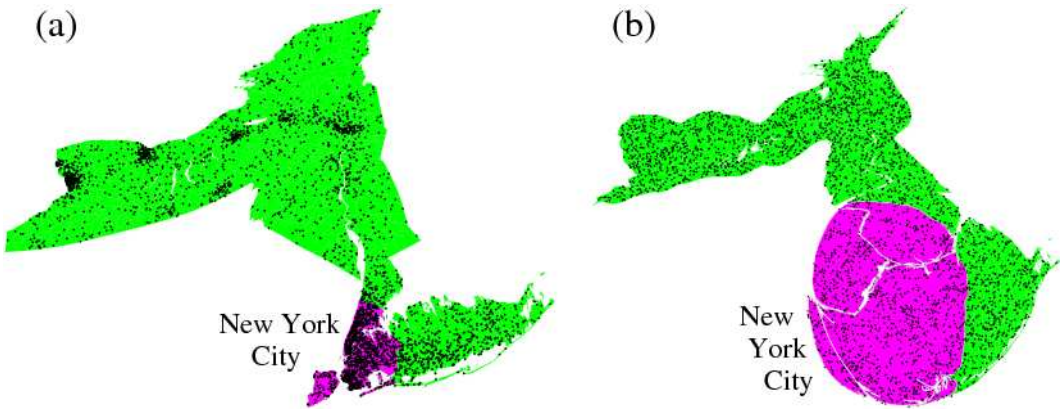

Figure 2.7: Lung cancer cases among males in the state of New York, 1993-1997, like in Fig. 2.1(a), but now on population cartograms (a) with a coarse-grained population density $\sigma = 50$ km, (b) with a much finer-grained population $\sigma = 1$ km.

Higher values are a sign of clustering; lower values indicate repulsion between points. Furthermore, it can be shown that the value of the Hopkins statistic on random points is beta-distributed which allows us to calculate a p-value—the probability that the observed value of the Hopkins statistic would occur if point positions were purely random. Details can be found in App. A.

Calculating the Hopkins statistic for Fig. 2.1(a) gives a value of $H = 0.89 \pm 0.03$ and a p-value $< 10^{-16}$. This is of course hardly surprising since the distribution shows very clear clustering around the major urban centers. But when we perform the same calculation for the points on the cartogram Fig. 2.7(b) we obtain $H = 0.50 \pm 0.03$ consistent with a random distribution. Thus, the per capita risk of lung cancer appears, by this calculation, to be the same everywhere. One might imagine that environmental effects such as pollution in big cities could influence cancer rates, but we see no evidence of this. A more careful analysis of course would have to take into account that people might move between the time when they are exposed to a carcinogen and the time when they develop cancer. However, Fig. 2.1(a), despite its impressive clustering, definitely cannot be taken as evidence that there are high-risk regions of New York state for cancer. This finding is consistent with medical evidence

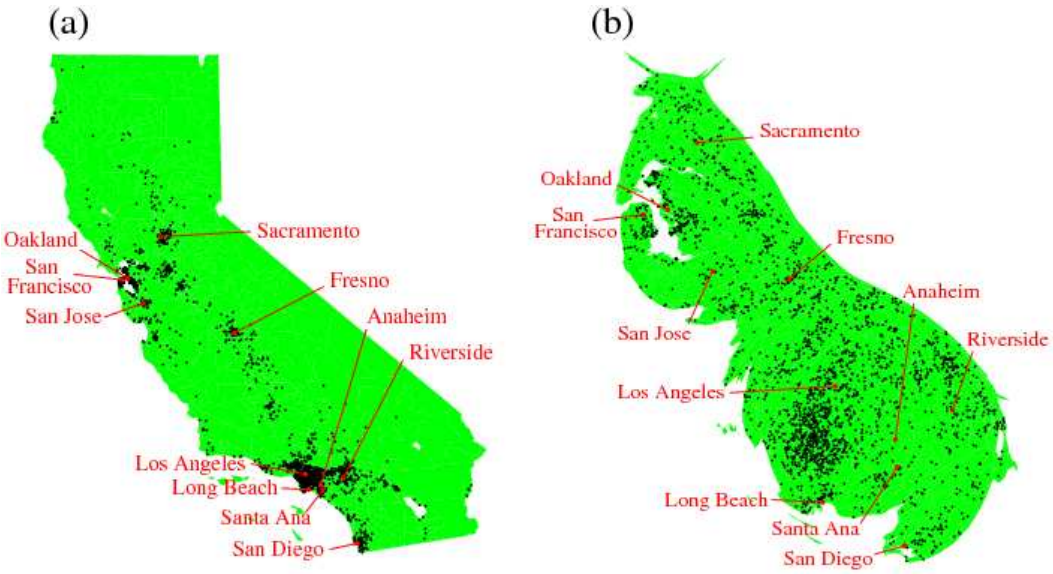

Figure 2.8: Homicides in California in 2001. (a) Equal-area projection. (b) Population cartogram. Data from the California Department of Health Services.

that lung cancer is caused by individual behavior rather than environmental causes, with smoking alone responsible for 90% of all cases worldwide (Spira *et al.*, 2004).

Not all threats to our well-being, however, are uniformly distributed. In Fig. 2.8(a) we plot each of the 2301 homicides that occurred in the state of California in the year 2001 on an equal-area map, and the map again shows clear clustering around urban areas, particularly the Greater Los Angeles area. Switching once again to a high-resolution population cartogram, Fig. 2.8(b), we see that the points become more spread out, as before, but this time it is clear that the distribution is far from random. Especially in the southwestern part of Los Angeles the density is clearly higher than average even on the cartogram. The Hopkins statistic confirms this impression. Its value of 0.68 ± 0.04 for the cartogram indicates that there is clustering even on the cartogram and the p-value for complete spatial randomness is below 10^{-8}, meaning the probability that we would get a point pattern such as this purely by chance is below one in a hundred million. Undoubtedly, some parts of the state experience a statistically significant higher number of homicides per capita than others.

For the last example in this section we look at technological rather than social data and analyze the geographic distribution of the Internet in the contiguous United States. The Internet is a network of computers and routers connected by optical fiber. Computers belonging to the same company or organization are typically grouped together into subnetworks called Autonomous Systems (ASes) or routing domains that share a single external routing policy. Here we treat each AS as a single point in geographic space; we have created a map of the Internet using the software tool NetGeo,[5] which can return approximate latitude and longitude for a specified AS. The positions of the 7049 ASes in the lower 48 states in March 2003 is shown in Fig. 2.9(a).

That ASes on this *equal-area map* appear heavily clustered in cities comes as no surprise since there are more people in the cities. Switching to a cartogram, Fig. 2.9(b), we find in this case that, like the homicides, the ASes become more uniform, but are still concentrated around the cities. Some urban areas appear particularly dense, such as Silicon Valley (west of San Jose), Manhattan (New York City), or Washington, DC. Other cities, by contrast, appear to have no more ASes per capita than average—for example, Detroit, Memphis, and Jacksonville. The Hopkins statistic for the distribution on the cartogram takes a value of $H = 0.80 \pm 0.04$, and the random distribution is firmly ruled out (the p-value is $< 10^{-16}$). In the early days of the Internet it was often claimed that geographic position would become unimportant thanks to high-speed data transmission and easy access to information from everywhere. Apparently, this has not happened.

[5] `http://www.caida.org/tools/utilities/netgeo`

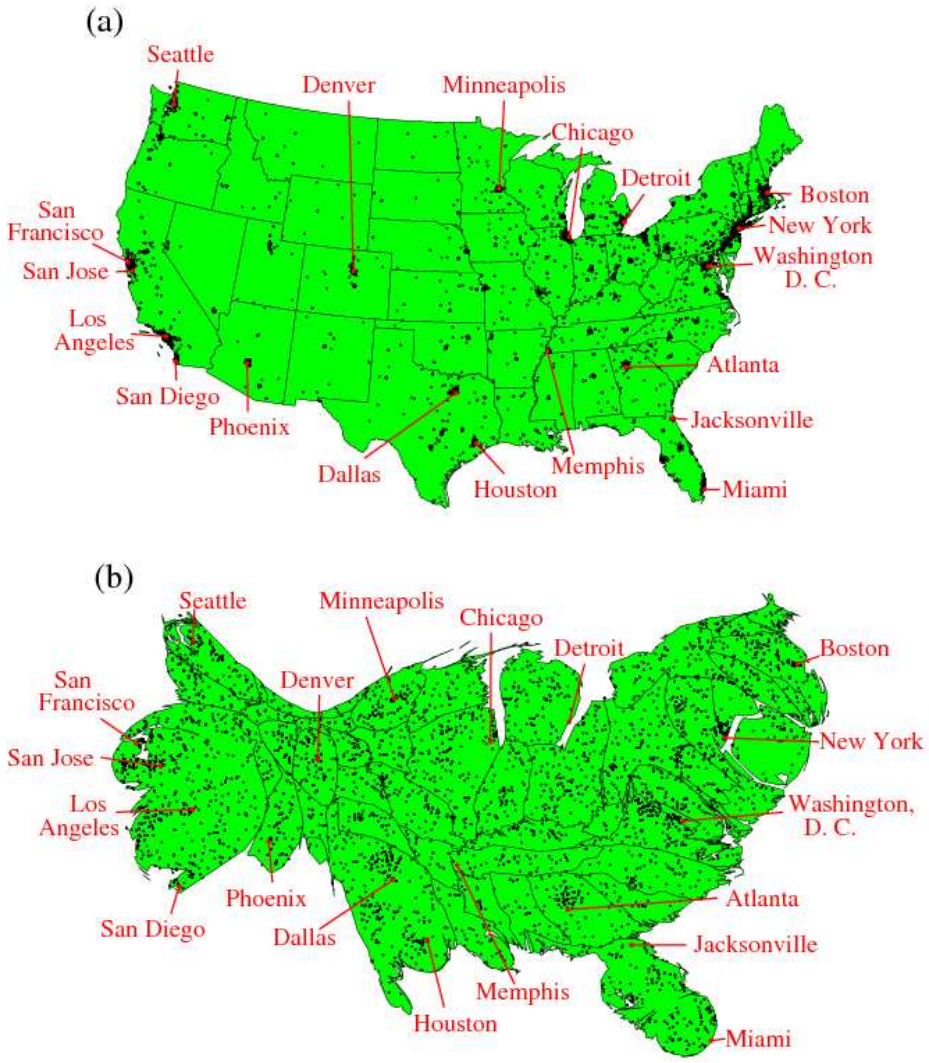

Figure 2.9: The Internet's Autonomous Systems in the contiguous United States (March 2003). (a) Equal-area projection. (b) Population cartogram.

2.6 Applications II: Election cartograms

On November 2, 2004, the day of the United States presidential election, and in the months since then, many of us will have seen maps of the election results in which each state is colored (conventionally) red or blue to indicate whether more of their voters voted for the Republican presidential candidate (George W. Bush) or for the Democratic candidate (John F. Kerry). We show such a map in Fig. 2.10. The map gives the superficial impression that the "red states" dominate the country, since they

cover far more area than the blue ones. However, as Dorling pointed out [32], election maps are often misleading because the conservative (i.e. Republican) strongholds tend to be in the countryside, whereas cities tend to be politically more liberal (i.e. Democratic). The blue states may, hence, be small in area, but they are large in terms of the number of people, which is what matters in an election.

We can correct for this by making use of a cartogram in which the sizes of states are not proportional to their sheer topographic acreage—which has little to do with politics—but to the number of their inhabitants. Thus, on such a map, the state of Rhode Island, which has about a million inhabitants, would appear twice the size of Wyoming, which has half a million, even though Wyoming has 60 times the acreage of Rhode Island. Figure 2.10(b) shows the presidential election results on a cartogram of this type. Population data were taken from the 2000 US Census. The cartogram reveals what we know already from the news: that the country was quite evenly divided by the vote, rather than being dominated by one side or the other.

The presidential election is not decided on the basis of the number of people who vote for each candidate, however, but on the basis of the electoral college. Each state contributes a certain number of electors to the electoral college, who vote according to the majority in their state.[6] The candidate receiving a majority of the votes in the electoral college wins the election. The electoral votes are apportioned among the states roughly according to population, as measured by the census, but with a small but deliberate bias in favor of less populous states.

We can represent the effects of the electoral college by scaling the sizes of states to be proportional to their number of electoral votes. The result is shown in Fig. 2.10(c). This figure looks similar to Fig. 2.10(b), but it is not identical. Wyoming, for instance,

[6]In theory there are two exceptions: Nebraska and Maine have laws that allow them to divide up their electoral votes between the candidates. In practice, however, neither of them has ever done this.

has approximately doubled its size, precisely because of the bias in favor of states with smaller populations. The areas of red and blue on the cartogram are now proportional to the actual numbers of electoral votes won by each candidate. Thus, this map shows simultaneously which states went to which candidate, and also which candidate won more votes—something that you cannot tell easily from the normal election-night red and blue map.

But we can go further. We can do the same thing also with the county-level election results and the images are even more striking. Fig. 2.11(a) shows a map of US counties, again colored red and blue to indicate Republican and Democratic pluralities. Similar maps have appeared in the press and have been cited as evidence that the Republican party has overwhelming support. Again, however, the populations of counties vary significantly. The most populous county in the United States is Los Angeles County, CA, with over 9.5 million inhabitants, while the least populous is Loving County, TX, with just 67, so there are more than five orders of magnitude variation between the two extremes. The distribution of populations appears to be roughly log-normal; a histogram is shown in Fig. 2.12.

Redrawing the county-by-county results on a cartogram, as shown in Fig. 2.11(b), again gives a more accurate picture of the election. Once more, the blue areas are much magnified and the total areas of blue and red are nearly equal. However, there is still more red than blue on this cartogram, even after allowing for population sizes, while the percentages of voters nationwide voting for either candidate were by contrast almost identical, so what is going on here?

The answer seems to be that the amount of red on the map is skewed because, of the larger counties that were won by the Republican candidate, many were won by a relatively narrow margin. Fig. 2.13 shows a scatter plot of the vote counts by county. Each point in this figure represents one county and the point's position shows the number of votes cast for the two major candidates. The diagonal lines indicate where

counties would fall with 25%, 50%, and 75% of votes cast for the Republican candidate. Most of the points in the plot fall above the 50% line, indicating a Republican majority, but most of the points representing substantial Republican majorities are for small counties with fewer votes in total. Counties of medium or larger size—which account for a large portion of the total area on our cartogram—tend to be won (or lost) by narrow margins and should therefore be considered neither purely Republican nor purely Democrat. Nonetheless in Fig. 2.11(b) these counties all appear purely red or blue, which gives rise to a misleading impression of the vote.

One way to allow for this, suggested by Vanderbei [33], is to use not just two colors on the map, red and blue, but a range from red to blue via various shades of purple for the different percentages of votes. In Fig. 2.14, we show the normal map and the cartogram colored using this scheme. In the cartogram, it appears that only a rather small area is taken up by true red or blue counties, the rest being mostly shades of purple.

We believe that the cartograms presented here can go a long way towards correcting some of the most glaring problems encountered with simple geographic representations of election results.

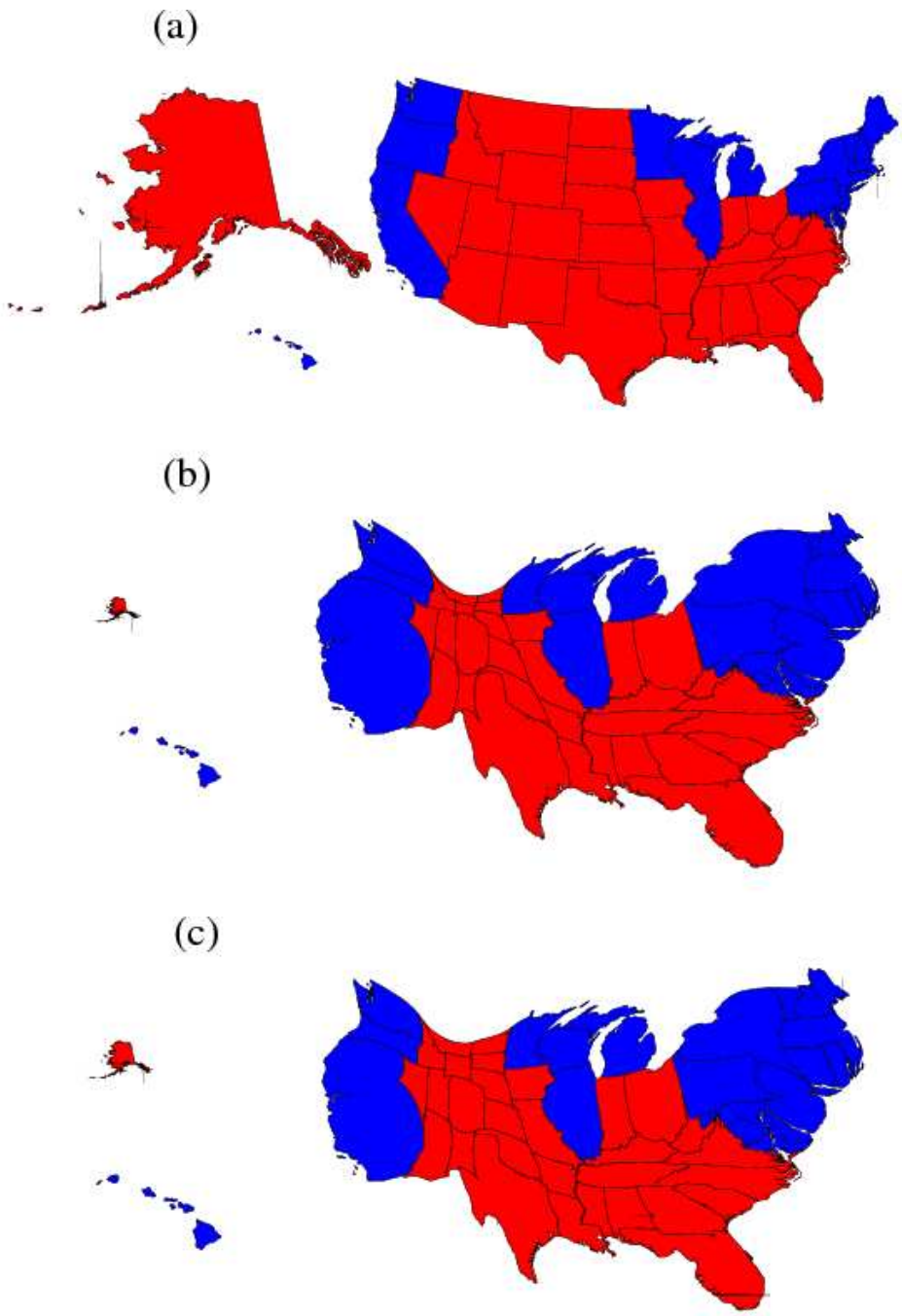

Figure 2.10: (a) The standard red and blue map of the results of the 2004 US Presidential election. The states are colored red if more voters voted for the Republican candidate than any other, and blue if more voters voted for the Democratic candidate than any other. (Because a small percentage of votes were taken by third-party candidates, this is not quite the same as saying a majority of voters voted Republican or Democrat.) (b) A cartogram in which the sizes of states are proportional to the states' populations. (c) A cartogram in which the sizes of states are proportional to the number of votes they have in the electoral college.

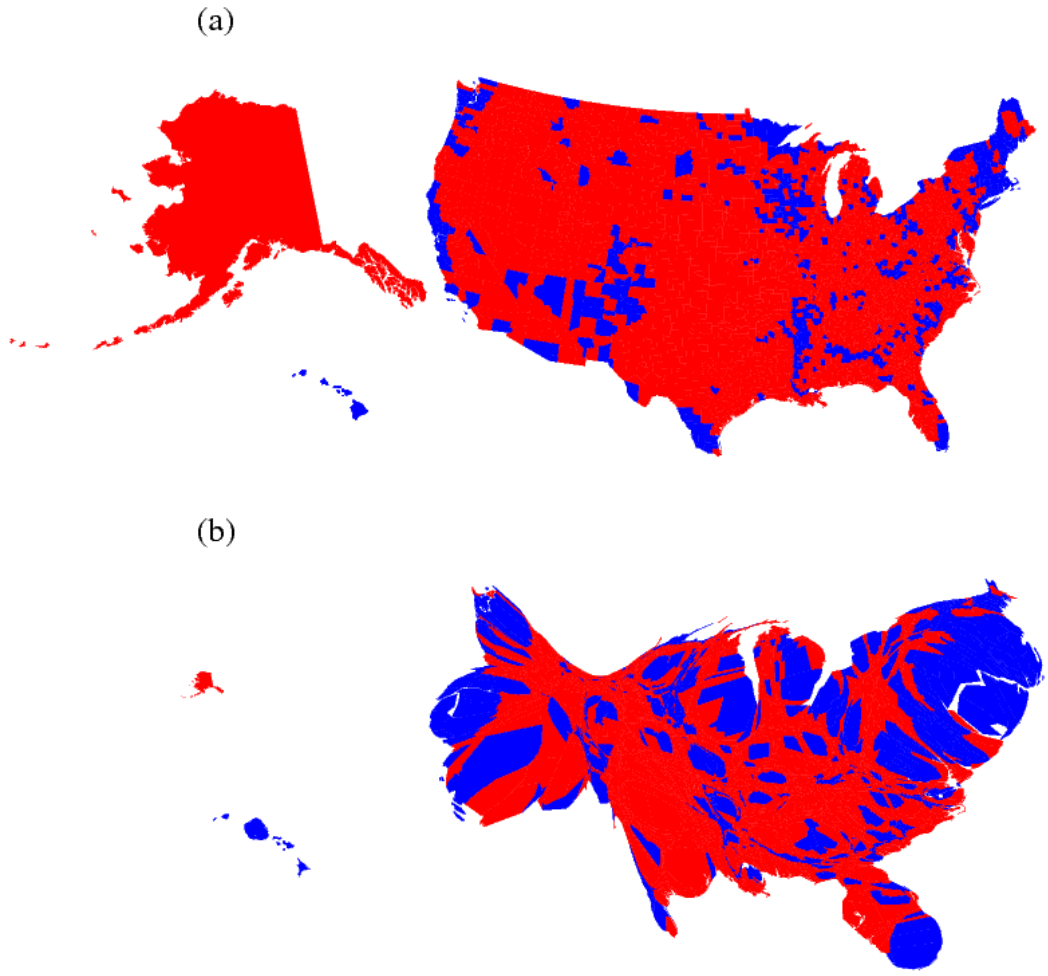

Figure 2.11: (a) A map of the counties of the United States, again colored red and blue to indicate Republican or Democratic pluralities. (b) The counties of the United States redrawn using a population cartogram.

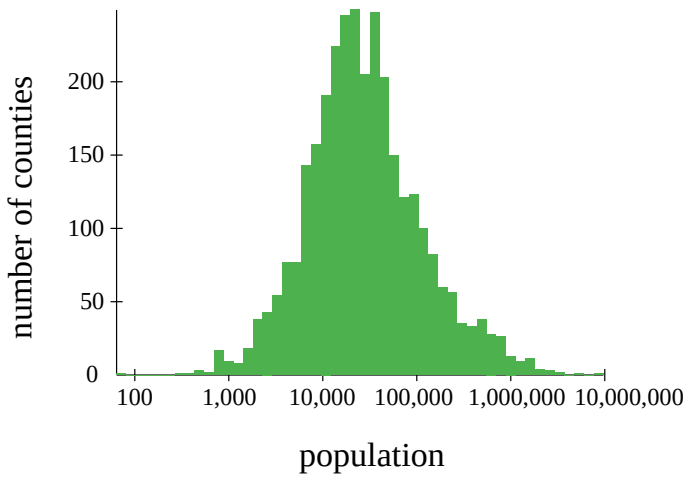

Figure 2.12: Histogram of the populations of US counties.

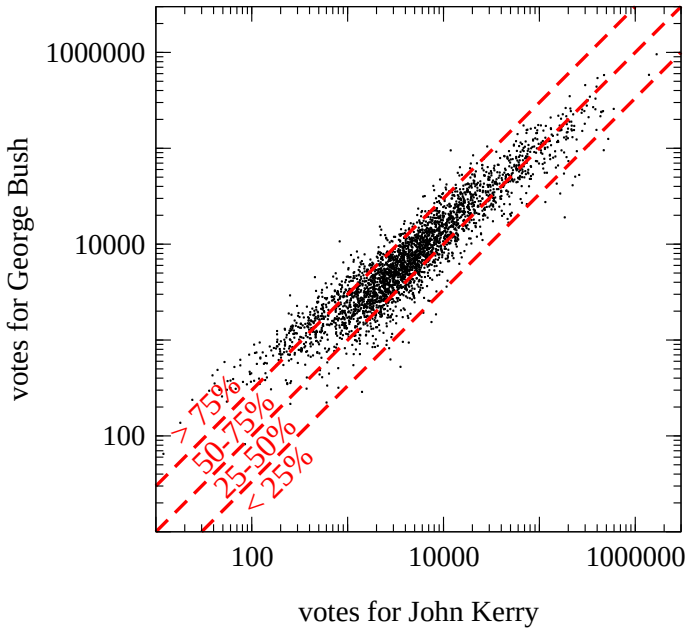

Figure 2.13: Scatter plot of votes, county by county. Each point represents the vote counts for the two major candidates in one county in the conterminous United States.

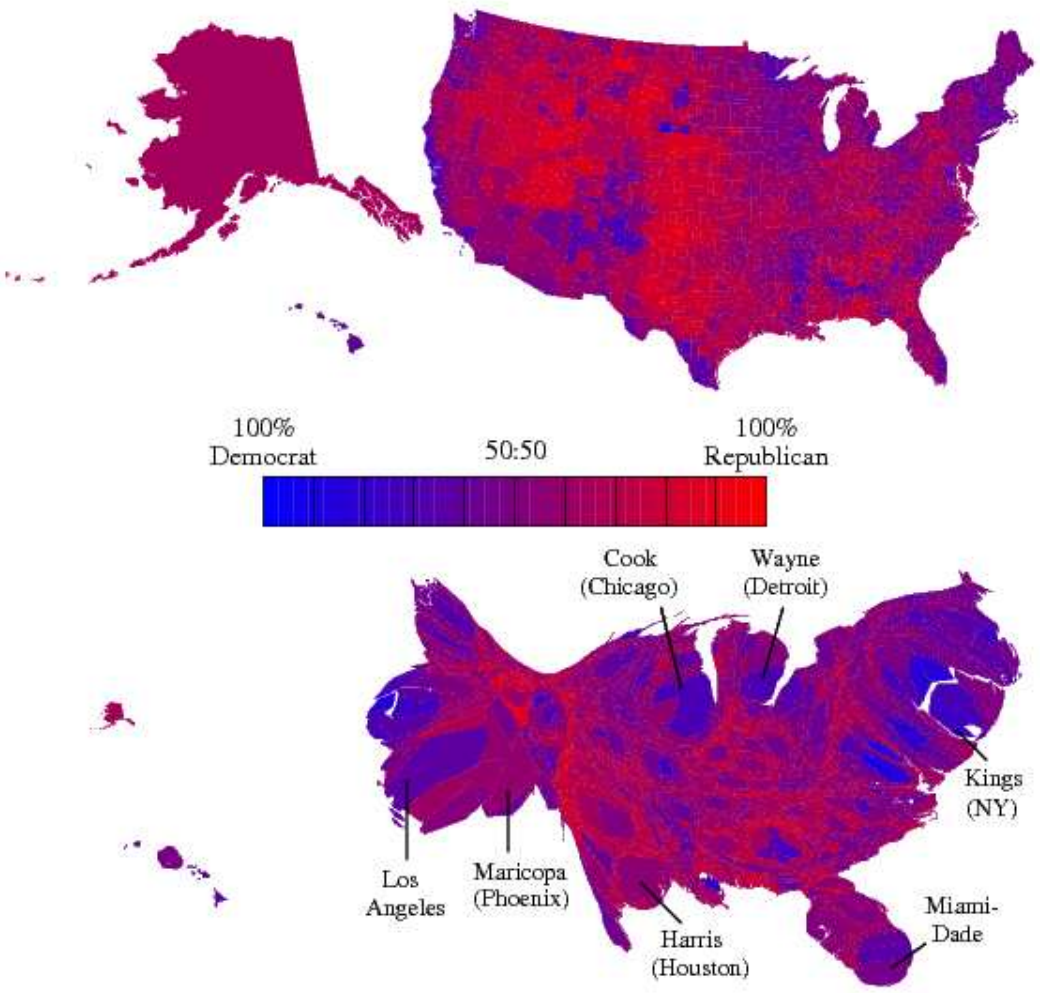

Figure 2.14: Map (top) and cartogram (bottom) showing the election results on the blue-purple-red scale in which the amount of blue or red in the color of each county is proportional to the fraction of votes going to the corresponding candidate (excluding votes for third-party candidates).

2.7 Applications III: Cartograms based on other density functions

The cartograms shown so far have all been based on human population density, which is certainly the most common type of cartogram. Other types, however, are also possible and we give some examples in this section

For our first example we examine the usage and production of energy in the United States. Each year, the US Energy Information Administration estimates each state's total energy consumption, including electricity, coal, gas, petroleum, wood, and alternative energy sources. For instance, in the year 2000 the United States consumed a total of 98 quadrillion British thermal units of energy. [34]. The use of energy varies greatly between the states, with Texas, for example, consuming 70 times as much energy as Vermont.

In Fig. 2.15(a) we show a cartogram in which states are scaled according to their total energy consumption during the year 2000. The cartogram appears quite similar to the population cartogram in Fig. 2.10(b), indicating that energy consumption per capita is roughly the same in most parts of the country. On closer inspection one might detect that the biggest state in Fig. 2.10(b), California, has been overtaken in Fig. 2.15(a) by what was formerly the second biggest state, Texas. New York and Florida have also become smaller, Pennsylvania and Louisiana larger, but all in all the changes are relatively minor—population density explains energy consumption quite well.

But now look at Fig. 2.15(b), which shows total energy *production* by state, again from figures compiled by the Energy Information Administration for the year 2000 and including crude oil, gas, and coal as well as electricity not generated from fossil fuels.[7] This cartogram gives a very different perspective on the country. A small number of

[7] `http://www.eia.doe.gov/emeu/states/2000StateEnergy_Sep2003.xls`

states, most notably Texas, Louisiana, and Wyoming, dominate the country's energy production, the first two because of their wealth in oil and gas, the third because of its abundance of coal. Total US energy production is 61 quadrillion British thermal units, which is only 62% of consumption—the difference is made up of imported energy—so the total area of Fig. 2.15(b) is smaller than that of Fig. 2.15(a) by the same factor. Fig. 2.15(a) and (b) together highlight the substantial redistribution of energy from producer to consumer in the United States.

For our last example we examine the media attention paid to different parts of the country. Anyone who reads or watches the news in the United States (and similar observations probably apply in other countries as well) will have noticed that the geographical distribution of news stories is not uniform. Even allowing for population, a few cities, notably New York and Washington, DC, get a surprisingly large fraction of the attention while other places get little. Apparently some locations loom larger in our mental map of the nation than others, at least as presented by the major media. We can turn this qualitative idea into a real map using our cartogram method.

We have taken about 72 000 newswire stories from November 1994 to April 1998 [35], and extracted from each the "dateline," a line at the head of the story that gives the date and the location that is the main focus of the story. Binning these locations by state, we then produce a map in which the sizes of the US states are proportional to the number of stories concerning that state over the time interval in question. The result is shown in Fig. 2.16.

The stories are highly unevenly distributed. New York City alone contributes 20 000 stories to the corpus, largely because of the preponderance of stories about the financial markets, and Washington, DC another 10 000, largely political stories.[8] We chose to bin by state to avoid large distortions around the cities that are the focus

[8]It should be noted that the dateline is the location of the news agency where the story is reported, which in some cases is not where the story takes place.

of most news stories. We made one exception however: since New York City had far more hits than any other location including the rest of the state of New York (which had around 1000), we split New York State into two regions, one for the greater New York City area and another for the rest of the state.

The cartogram is a dramatic depiction of the distribution of US news stories. The map is highly distorted because the patterns of reporting show such extreme variation. Washington, DC, for instance, which normally would be virtually invisible on a map of this scale, becomes the second largest "state" in the union. (The District of Columbia is not, technically, a state.) People frequently overestimate the size of the northeastern part of the United States by comparison with the middle and western states, and this map may give us a clue as to why. Perhaps people's mental image of the United States is not really an inaccurate one; it is simply based on things other than geographical area, such as the attention regions receive in the media.

Numerous other possible applications of cartograms come readily to mind, such as visualizations of gross regional products, number of people belonging to certain ethnic groups, sales of certain consumer goods, and so forth. Diffusion cartograms might also have applications outside geography. One possibility is the creation of a homunculus, a representation of the human body in which each bodily part is scaled in proportion to the size of the brain region devoted to it [36]. Such representations are usually constructed as two-dimensional plots, but there is no reason in theory why one could not create a fully three-dimensional homunculus; the diffusion process is easily generalized to any number of dimensions.

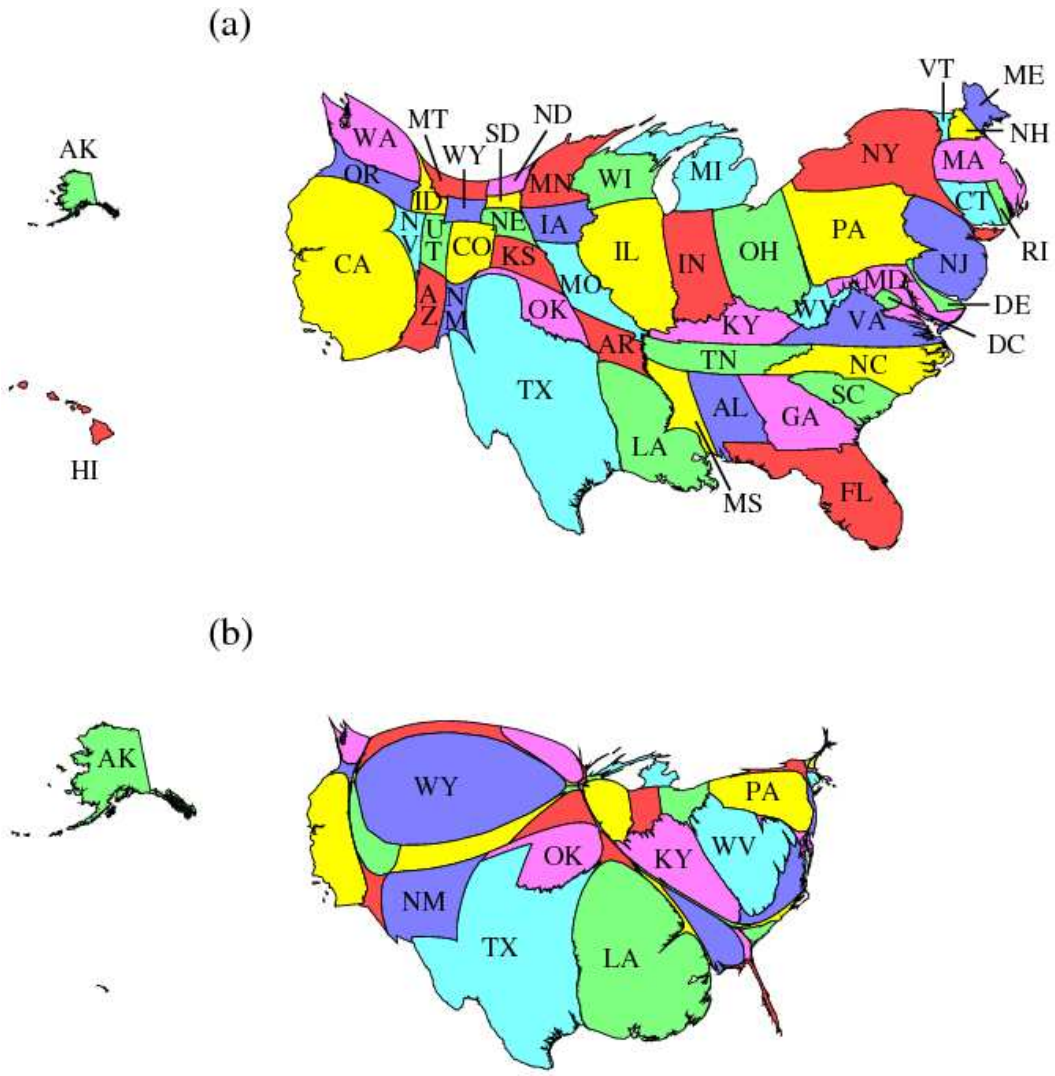

Figure 2.15: (a) A cartogram of the United States in which the sizes of states are proportional to their total energy consumption. (b) A similar cartogram for energy production. States are the same color in (a) and (b).

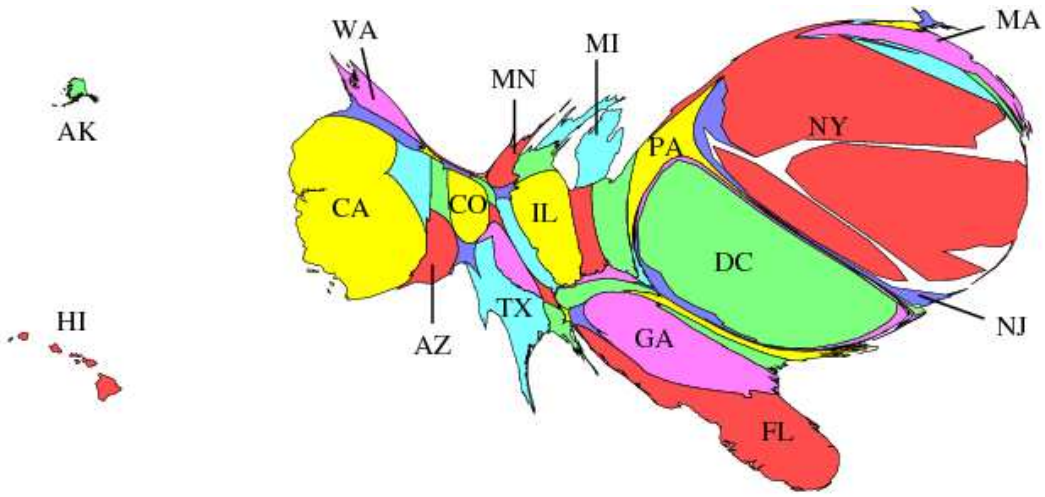

Figure 2.16: Cartogram in which the sizes of states are proportional to the frequency of their appearance in news stories. States are the same color as in Fig. 2.15.

2.8 Performance of the algorithm

An important consideration with any method for producing cartograms is efficiency. In many cases one would like to create cartograms interactively for data exploration, which means that program run times for producing them should be limited to seconds, or at most a few minutes for the most complex cartograms such as the larger country-sized ones shown above. In this section we provide some figures on run times and analysis of the efficiency of our cartogram method.

We have implemented our algorithm as a C program and it is this implementation that we analyze here. (No doubt faster implementations than ours are possible, given time and effort, but we believe ours to give a good general indication of the speeds attainable.) We analyze the performance of the program in calculating a cartogram of the lower 48 states and the District of Columbia with each region scaled according to population, which is the cartogram in Fig. 2.10(b) (without Alaska and Hawaii).

The program goes through the following steps in creating the cartogram (see App. B). First, the polygons making up the regions of the input map are read from ASCII files, along with data on the population of each region. Then, a fine square grid is created and filled with the initial density distribution. The grid spacing has to be chosen such that the smallest polygon is covered by at least a few grid points. For the present example we chose a 1024×512 grid, which is adequate to resolve the smallest region, the District of Columbia. The program then evaluates the cartogram transformation for each grid point by solving the diffusion equation as outlined in Section 2. The positions of the vertices of each polygon are calculated by piecewise bilinear transformations from the points of the distorted grid.[9] For this example our

[9]It would be entirely possible—and in some cases quicker—to transform the polygon vertices separately using the diffusion equation, but the grid-based method described here is simpler and more general and gives results essentially as good; the grid is fine enough that the errors introduced by the interpolation are small.

program runs to completion within $3\frac{1}{2}$ minutes on a standard desktop computer with a 2.8GHz Intel Pentium IV processor. Total memory used is 22MB.

Another important measure is the accuracy of the algorithm. How precisely are the final areas proportional to population? We measure the fractional error in each region's area by calculating the quantity

$$\text{relative error} = \frac{\text{area of state on cartogram} \times \text{total population of all states}}{\text{total area of all states on map} \times \text{population of state}} - 1. \tag{2.15}$$

The results are shown as the red bars in Fig. 2.17. Most states are within $\pm 10\%$ of their target value, with the exception of Washington, DC, and Rhode Island, which are too small, and Vermont and West Virginia, which are too big. The inaccuracies are caused partly by sampling the density on a finite grid and partly by approximations made by the integrator routine. For most applications these errors are probably acceptable. On the input map the densities vary by more than a factor of 1600 between the densest region, Washington, DC, and the sparsest, Wyoming. On the cartogram the extreme densities—now Washington, DC, and West Virginia—only differ by a factor of about 2, or almost three orders of magnitude less.

However, if the errors are a major concern there is a simple way to improve the result, namely running the algorithm again starting from the cartogram produced by the first run. The time taken is the same as before, but the area errors are now reduced to less than 3.5% in all cases (the green bars in Fig. 2.17) which is certainly much less than can be detected by eye. If even higher accuracy is needed one can of course run the algorithm as many times as needed. Alternatively, one could make the initial square grid finer from the start at the expense of longer run-times. For an $L_x \times L_y$ grid the time needed for a single run scales as $L_x L_y \log(L_x L_y)$, the bottleneck being the fast Fourier transform. The memory use grows as $L_x L_y$. Memory is not likely to be a problem on current computers for maps of any reasonable size. We have without difficulty carried out computations on lattices of size up to 4096×2048.

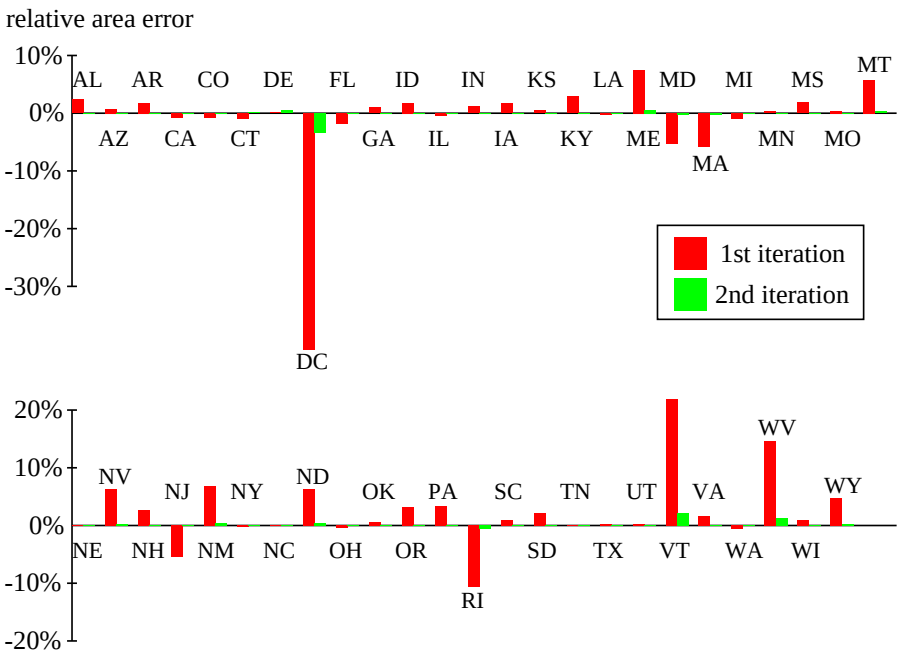

Figure 2.17: Relative area errors on a population cartogram of the lower 48 states and Washington, DC, after applying a C implementation of the algorithm outlined in App. B.

Off-shore islands and donut shaped regions are treated correctly by our method; their topology will be strictly preserved. If, however, the map consists of several non-contiguous regions scattered over a wide area with a lot of empty space in between, covering all of the map with one big grid can be wasteful. For example, on an accurate map of all fifty states in the USA, Alaska extends so far to the north and Hawaii so far to the west of the American continent that calculating the cartogram on a grid wide enough to reach the remotest parts of the country would be very wasteful. We would in particular spend a large fraction of our effort on points in the middle if the Pacific Ocean between California and Hawaii, which are irrelevant for most purposes.

Conventionally, map makers move Alaska and Hawaii closer to the contiguous USA to fit them on one map, but this is a rather poor choice as input to our cartogram program because moving them closer to the lower 48 states can cause their diffusion patterns to interact with those of the lower 48, influencing the final shape of the

cartogram.[10] (The areas of the states would still be correct, but their shapes would be distorted.) A better solution is to construct separate cartograms for Alaska and Hawaii, and then afterward place them next to the cartogram for the continental USA in the traditional fashion. Some care must be taken to make sure the population scale of the different cartograms matches correctly. The cartograms in Fig. 2.15 and Fig. 2.16 were constructed in this manner.

2.9 Conclusion

In this chapter we have described a new general method for constructing density-equalizing projections or cartograms, which provide an invaluable tool for the presentation and analysis of geographic data. Our method is simpler than many earlier methods, allowing for rapid calculations, while generating accurate and readable maps. The method allows its users to choose their own balance between good density equalization and low distortion of map regions, making it flexible enough for a wide variety of applications. We have presented a number of examples of the use of our cartograms in the representation of human data.

We have implemented our method as a C program which is available from our web site.[11] Directly embedding the algorithm as a function in GIS software packages should also be straightforward. For even the most complex examples, the program achieves good density-equalization on the time scale of a few minutes with current computing resources.

One interesting direction for future research is the creation of cartograms in spaces

[10] Islands just off the coast, like Long Island, will of course influence diffusion on the mainland too, but this is correct because these islands really are in the "neighborhood" and their influence should be felt. The high population of Brooklyn and Queens on Long Island for example should expand and repel the mainland as in Fig. 2.7, thereby preserving the topology of map.

[11] `http://www.santafe.edu/~mgastner`

that are not flat. Here we have assumed that our mapped space is flat, but this is never strictly true since the surface of the Earth is curved. Even for an area as large as the contiguous United States it is usually legitimate to neglect curvature when a suitable projection is used for the input map (such as the Albers conic projection used here). For a larger area—a cartogram of the entire world, for instance—this would no longer be possible. In that case we would have to solve the diffusion equation (2.7) on the surface of the sphere in spherical coordinates:

$$\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\rho}{\partial\theta}\right)+\frac{1}{\sin^2\theta}\frac{\partial^2\rho}{\partial\phi^2}=\frac{\partial\rho}{\partial t}. \tag{2.16}$$

The solution can be expressed in terms of spherical harmonics $Y_{lm}(\theta,\phi)$ thus:

$$\rho(\theta,\phi,t)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\tilde{\rho}_{lm}Y_{lm}(\theta,\phi)\exp[-l(l+1)t] \tag{2.17}$$

where

$$\tilde{\rho}_{lm}=\int Y^*_{lm}(\theta,\phi)\rho(\theta,\phi,t=0)\,\mathrm{d}\Omega. \tag{2.18}$$

As in the Cartesian case we can now use Equation (2.8) to solve for the velocities

$$\begin{aligned}
v_\theta(\theta,\phi,t) &= -\frac{1}{\rho}\frac{\partial\rho}{\partial\theta}\\
&= \frac{\sum_{l=0}^{\infty}\sum_{m=-l+1}^{l}\sqrt{l(l+1)-m(m-1)}\tilde{\rho}_{lm}Y_{l,m-1}(\theta,\phi)e^{i\phi}e^{-l(l+1)t}}{2\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\tilde{\rho}_{lm}Y_{lm}(\theta,\phi)e^{-l(l+1)t}}\\
&\quad -\frac{\sum_{l=0}^{\infty}\sum_{m=l}^{l-1}\sqrt{l(l+1)-m(m+1)}\tilde{\rho}_{lm}Y_{l,m+1}(\theta,\phi)e^{-i\phi}e^{-l(l+1)t}}{2\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\tilde{\rho}_{lm}Y_{lm}(\theta,\phi)e^{-l(l+1)t}},\\
v_\phi(\theta,\phi,t) &= -\frac{1}{\rho\sin\theta}\frac{\partial\rho}{\partial\phi}\\
&= -i\frac{\sum_{l=0}^{\infty}\sum_{m=-l}^{l}m\tilde{\rho}_{lm}Y_{lm}(\theta,\phi)}{\sin\theta\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\tilde{\rho}_{lm}Y_{lm}(\theta,\phi)e^{-l(l+1)t}}.
\end{aligned} \tag{2.19}$$

To solve Equations (2.17), (2.18), and (2.19) efficiently we need the equivalent of the fast Fourier forward and backward transforms for spherical harmonics. Although such transforms exist, implementations are far from trivial and a current field of research [37].

CHAPTER III

FACILITY LOCATION - THE CONTINUOUS P-MEDIAN PROBLEM FOR A NON-UNIFORM POPULATION

3.1 Introduction

Suppose we are given the population density $\rho(\mathbf{r})$ of a country or province, i.e. the number of people per unit area, as a function of geographical position $\mathbf{r}$. And suppose we are charged with choosing the sites of p hospitals, post offices, supermarkets, gas stations, elementary schools or similar facilities. This problem belongs to a class of optimization problems known collectively as "facility location": Given a set of demand points, e.g. the home addresses of patients, customers, or schoolchildren, find a set of facilities which minimizes some objective function. This function might depend on travel times, land prices, utility costs, local taxes, proximity to customers and raw materials, and many other factors.

Here we simply wish to minimize the mean distance to the nearest facility for a member of the population. If the facilities are at positions $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_p$, the objective function can be written as

$$f(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_p) = \int_A \rho(\mathbf{r}) \min_{i \in \{1,2,...,p\}} |\mathbf{r} - \mathbf{r}_i| \, \mathrm{d}^2 r \tag{3.1}$$

which is proportional to this mean distance. The facilities are allowed to lie anywhere in continuous two-dimensional space. Finding exact solutions, a task known as the p-median problem, turns out to be very challenging. In Sec. 3.2 we review previous

work on the p-median problem before presenting a new heuristic solution method based on simulated annealing in Sec. 3.3.

Even though determining the optimum positions $\mathbf{r}_1, ...\mathbf{r}_p$ is difficult, an approximate analytic solution for the density of facilities can be obtained. In Sec. 3.4 we prove that the density of facilities increases as the two-thirds power of the population density ρ, which will allow us to draw a connection between the density-equalizing map projections from Chap. II and the p-median problem. In Sec. 3.5 we investigate whether real facilities follow the two-thirds power law. We end this chapter with concluding remarks in Sec. 3.6.

3.2 Literature review

Facility location problems have a long history dating back to the work of Weber in 1909 [38]. Named after him, the "Weber problem" consists of locating one facility in two-dimensional Euclidean space for a finite number of demand points such that the sum of the distances between facility and demand points is minimal. (For three demand points, the solution is the well-known Fermat point of the triangle spanned by the three points, see Fig. 3.1.) The Weber problem is a special case of Eq. (3.1) with $p = 1$ and a finite sum replacing the integral. An efficient and exact solution to the Weber problem was presented by Weiszfeld already in 1937 [39]. But for $p > 1$, the

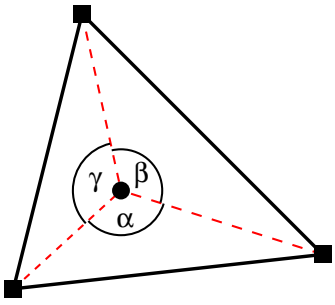

Figure 3.1: A simple facility location problem. The point whose summed distance to three demand points (squares) is minimal is the Fermat point (circle) in the triangle spanned by the demand points. It is characterized by $\alpha = \beta = \gamma = 120°$.

problem becomes considerably more difficult. Exact solutions for this "multisource Weber problem" were first obtained by Kuenne and Soland in 1972 [40], but, although more refined geometric techniques were put forward subsequently, even the largest exactly solvable problem did not exceed $p = 5$ and 30 demand points [41]. Since the p-median problem for a discrete set of demand points is NP-hard [42], it is in fact very unlikely that an algorithm obtaining exact solutions in a polynomially bounded time will ever be found. (A decision problem belongs to the class NP if a "yes" instance can be verified in a time that does not increase faster than a polynomial in the size of the input. A problem Π is *NP-hard* if all problems in NP can be solved in polynomial time after solving Π. NP-hard problems are hence, roughly speaking, at least as difficult as all problems in NP. Many optimization problems are NP-hard, e.g. the traveling salesman problem and the knapsack problem. For details see [43].)

The same is incidentally true for the p-median problem on a network [44]. There we are given a finite set of points V and connections between them, and we are charged with finding a set $W \subseteq V$, $|W| = p$, of facilities such that the mean distance between points in V and their closest facility along the network's connections is minimal. Most work on the p-median problem has in fact concentrated on this network perspective, with numerous papers describing approximate solutions, e.g. [45, 46, 47, 48, 49]. These results, however, cannot easily be generalized to the case where the facilities are allowed to be located not only at discrete sites, but everywhere in the two-dimensional plane.

A number of authors have investigated the latter case, but for discrete sets of demand points [50, 51, 52, 53, 54, 55, 56, 57, 58, 59]. If the demand is given in the form of a continuous density $\rho(\mathbf{r})$, the p-median problem remains NP-hard [60], and, to our knowledge, the only heuristic method presented so far is by a group of Japanese authors [61, 62]. They use an alternative, but equivalent formulation of Eq. (3.1) based on the observation that p facilities naturally divide the area into p

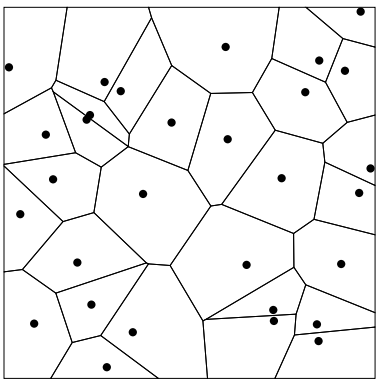

Figure 3.2: An example of a Voronoi tessellation. The polygon around each facility (circle) contains the points closer to this than any of the other facilities.

regions or "cells" such that every region contains the points closest to one of these facilities (see Fig. 3.2). This construction is known as the "Voronoi tessellation" [63] and the cells are called "Voronoi polygons" or "Voronoi cells". Since only the distance to the nearest facility appears in f, we can decompose the integral as a sum over the Voronoi cells $V_1, V_2, ..., V_p$ thus

$$f(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_p) = \sum_{i=1}^{p} \int_{V_i} \rho(\mathbf{r}) |\mathbf{r} - \mathbf{r}_i| \, \mathrm{d}^2 r. \tag{3.2}$$

From this expression it can be proved that the partial derivatives are

$$\frac{\partial f}{\partial x_i} = \int_{V_i} \rho(\mathbf{r}) \frac{x_i - x}{|\mathbf{r} - \mathbf{r}_i|} \, \mathrm{d}^2 r, \quad \frac{\partial f}{\partial y_i} = \int_{V_i} \rho(\mathbf{r}) \frac{y_i - y}{|\mathbf{r} - \mathbf{r}_i|} \, \mathrm{d}^2 r \tag{3.3}$$

with $\mathbf{r}_i = (x_i, y_i)$. Given an initial set of points $\mathbf{r}_1^{(0)}, ..., \mathbf{r}_p^{(0)}$, Suzuki and Okabe [62] proceed by moving each point in the direction of $-\partial f / \partial \mathbf{r}_i$ towards smaller values of f. Then the derivatives for the new set of points are calculated and the points shifted again in the opposite direction of the gradient. This procedure is iterated until a local minimum of f is reached and no further improvements can be made. Repeating the calculation for different initial conditions, the smallest observed local minimum is often a good approximation of the global minimum.

Implementing this algorithm obviously requires some work: First the Voronoi

tessellation must be constructed before each iteration, then the integrals of Eq. (3.3) have to be calculated, and finally we need to decide on a strategy for how far to move along the direction of the gradient at each step. Several authors have therefore wondered if such complications could be bypassed by more graphic and intuitive methods, for example using a suitable map projection. Bunge [64] conjectured that one solution is to put equally spaced points, e.g. a hexagonal lattice, on a density-equalizing map projection and transform the positions back to an equal-area map. Getis [65] and Rushton [66] have worked in this direction, but, as Gusein-Zade [67] points out, this approach does not even approximately solve the p-median problem. The crux is that a deformation that makes the population density constant cannot simultaneously conserve distances (see e.g. [68]). Nevertheless, as we will demonstrate in Sec. 3.4, there is a connection between the p-median problem and density-equalizing projections, but in a less direct way.

3.3 A heuristic based on simulated annealing

Summarizing the previous section, the only effective method of minimizing Eq. (3.1) proposed so far appears to be the steepest-descent algorithm by Suzuki and Okabe. Their algorithm finds the nearest local minimum, but, since f is neither convex nor concave, this is usually not globally the best [69]. It seems likely that performance could be improved by an algorithm that does not always get trapped at the bottom of the nearest "valley". In this section we present a new heuristic based on simulated annealing which not only has this property, but also generally finds better solutions than the steepest-descent method.

The basic idea of simulated annealing (SA) is to allow the algorithm to escape from a local minimum by occasionally permitting the objective function to increase, hoping that the solution will subsequently converge to a lower-lying minimum. To be concrete, we repeatedly generate new positions for the p facilities. If this leads to

an increase $\Delta > 0$ in f, we accept the new configuration with probability $\exp(-\beta\Delta)$, where $\beta \geq 0$, or discard it with probability $1 - \exp(-\beta\Delta)$. If on the other hand $\Delta \leq 0$, we always accept the new positions. (This acceptance probability is known as the Metropolis criterion. Other probabilities are possible, but less practical [70].)

Initially, β is set to a small value so that practically every arrangement of the p facilities is accepted. After each iteration, we then raise β, imposing an increasingly tougher penalty on "uphill" steps. The purpose is to first explore a large portion of the configuration space and then focus on the regions closer to the minima. After a certain number of iterations, the current facility positions are returned as the end result.

An efficient implementation depends crucially on how we generate new positions. In our work we have used two basic "moves" which change the position of one randomly chosen facility while leaving all others where they were before. In the first type of move, the facility jumps to a new entirely random position, while in the second type it is only permitted to move locally by a small amount. Since we only change one facility at a time, we could in principle avoid the bottleneck of the steepest-descent algorithm, namely the repeated construction of all Voronoi cells, by dynamically updating the tessellation [71], but we have not yet implemented this option. We also need to decide on the "cooling schedule," i.e. the values of β for each iteration. Here we chose an exponential cooling schedule with a starting value $\beta = 1/f$, where f is the function value for the initial random facility configuration. After each iteration, β is increased by a factor $1 + 1.5 \cdot 10^{-6}$, and we stopped after $1.5 \cdot 10^6$ steps when usually no further significant improvements of f could be made. Neither reducing the starting value of β nor slowing down its subsequent increase led to better numerical results. Faster annealing schedules, however, should be possible if time is a concern.

To compare this new heuristic with the steepest-descent method, we calculated near-optimal facility locations for four different density distributions and $p = 100$

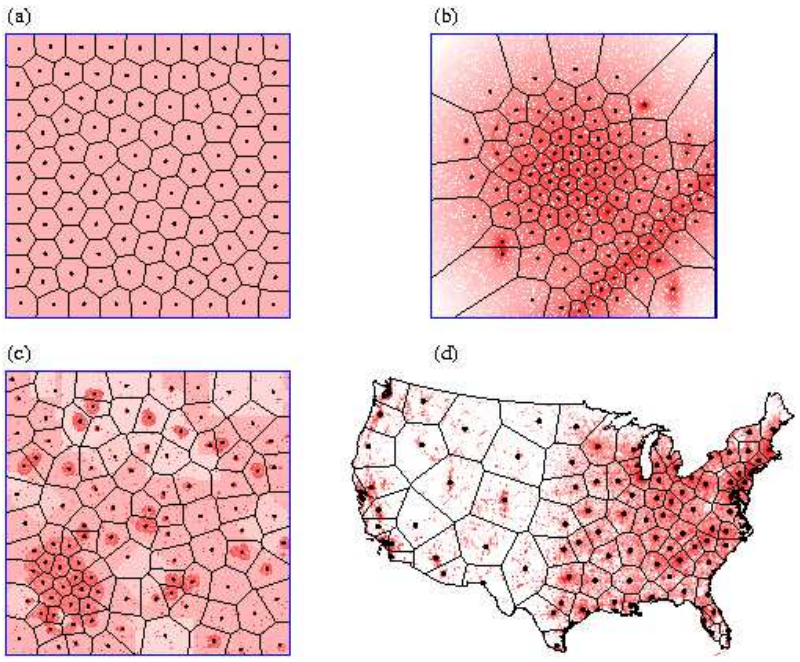

Figure 3.3: Near-optimal facility locations for 100 facilities. Stronger shades of red indicate a higher population density.

facilities. The minimum of the steepest-descent method, obtained in 20 runs with different random initial conditions, comes close to the SA result for the uniform density of Fig. 3.3a, as well as for the smoothly varying density of Fig. 3.3b. The difference is only a fraction of a percent, 0.07% in the first and 0.18% in the latter case, with SA performing slightly better in both cases. The difference, however, becomes more pronounced for the strongly fluctuating density of Fig. 3.3c and the real population density of the United States, Fig. 3.3d. Here the best minima found by SA are 3.44% and 7.54% less. In most real facility location problems, the density looks more like the one in Fig. 3.3d than that in a or b, and, hence, SA will be our method of choice for the following numerical computations.

3.4 The relationship between population density and optimal facility density

It is obvious from Fig. 3.3 that the largest Voronoi cells fall in regions of low population density. This is not surprising since it gains us little to build many facilities in sparsely populated areas. A more sensible choice is to distribute facilities in proportion to population density, so that a region with twice as many customers has twice as many facilities. But this too turns out to be suboptimal, because we also gain little by having closely spaced facilities in the highly populated areas—when facilities are closely spaced, the typical person is not much further from their second-closest facility than from their closest, so the closest could be removed with little penalty and substantial savings. The optimum of Eq. (3.1) has to lie somewhere between these two extremes.

Let us define $s(\mathbf{r})$ to be the area of the Voronoi cell in which a person at position $\mathbf{r}$ lives. In two-dimensions such a person will on average be a distance $g[s(\mathbf{r})]^{1/2}$ from the nearest facility, where g is a geometric factor of order 1, whose exact value depends on the shape of the Voronoi cell, but which will in any case drop out of our final result. Now the distance to the nearest facility averaged over all members of the population is proportional to $f = g\int_A \rho(\mathbf{r})[s(\mathbf{r})]^{1/2}\,\mathrm{d}^2r$, where we are making an approximation by neglecting variation of the geometric factor g. Since there are p facilities in total, $s(\mathbf{r})$ also satisfies the constraint

$$\int_A [s(\mathbf{r})]^{-1}\,\mathrm{d}^2r = p. \tag{3.4}$$

Optimizing the mean distance subject to this constraint gives

$$\frac{\delta}{\delta s(\mathbf{r})}\left[g\int_A \rho(\mathbf{r})[s(\mathbf{r})]^{1/2}\,\mathrm{d}^2r - \alpha\left(p - \int_A [s(\mathbf{r})]^{-1}\,\mathrm{d}^2r\right)\right] = 0, \tag{3.5}$$

where α is a Lagrange multiplier. Performing the functional derivatives and rearranging for $s(\mathbf{r})$, we find $s(\mathbf{r}) = [2\alpha/g\rho(\mathbf{r})]^{2/3}$. The Lagrange multiplier can be evaluated

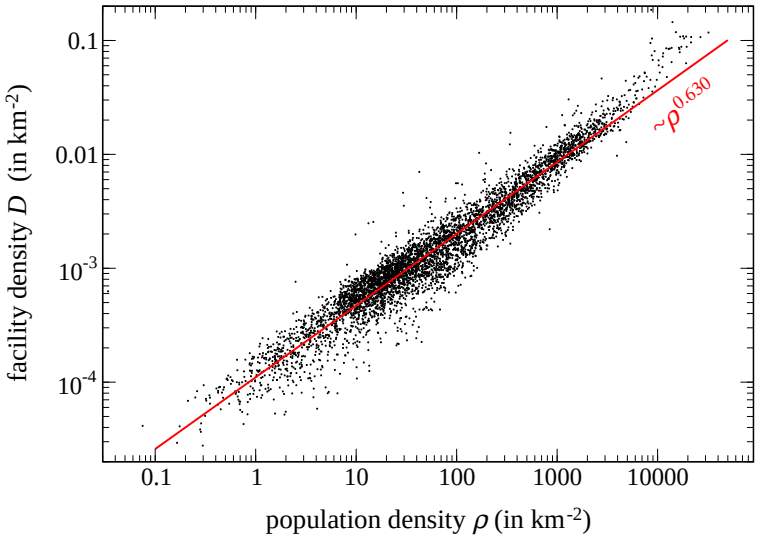

Figure 3.4: Facility density D versus population density ρ on a double-logarithmic plot. A least-squares linear fit to the data is a line of slope 0.630.

by substituting into Eq. (3.4), and we arrive at the result

$$D(\mathbf{r}) = \frac{1}{s(\mathbf{r})} = p\frac{[\rho(\mathbf{r})]^{2/3}}{\int [\rho(\mathbf{r})]^{2/3}\, \mathrm{d}^2 r}, \tag{3.6}$$

where we have introduced the notation $D(\mathbf{r}) = [s(\mathbf{r})]^{-1}$ for the density of the facilities.

Thus, if facilities are distributed optimally for the given population distribution, their density increases with population density but does so slower than linearly, namely as a power law with an exponent $\frac{2}{3}$.[1] This places most facilities in the densely populated areas where most people live while still providing reasonable service to those in sparsely populated areas where a strictly population-proportional allocation might leave inhabitants with little or nothing. Several authors have proved the same result by different means [72, 73, 74, 75]; our calculation is most similar to that by Gusein-Zade [67].

We can use the SA heuristic of the previous section to confirm the validity of

[1]The analogous calculation can be carried out in any dimension d yielding an exponent $\frac{d}{d+1}$ in this general case.

our approximations. For the US population density, Fig. 3.3d, and the near-optimal spatial arrangement of $p = 5000$ facilities, we determine the Voronoi cell around each facility, calculate D as the inverse of its area and ρ as the number of people living in the cell divided by its area. In a double-logarithmic plot with ρ along the x- and D along the y-axis, we expect the data to follow a line of slope $\frac{2}{3}$. A least-squares fit to the numerical data yields a slope 0.630, in good agreement with the theoretical prediction.

Some statistical concerns might be raised about this method. First, we used the Voronoi cell area to calculate both D and ρ, so the measurements of x- and y-values in the plot are not independent, and one might suspect that a positive slope is due to a positive correlation between the measurements rather than the actual densities [76]. Second, estimating the exponent of a power law such as Eq. (3.6) from a double-logarithmic plot introduces systematic biases [77] and should not be solely relied upon.

We therefore suggest a different method to test Eq. (3.6) based on the density-equalizing map projection of Chap. II. As pointed out in Sec. 3.2, facilities on a population-density cartogram are generally not regularly distributed. But if the proportionality $s(\mathbf{r}) \propto [\rho(\mathbf{r})]^{2/3}$ holds true, then the size of the Voronoi cells s should appear approximately constant on a cartogram based on the expected *facility* density $\rho^{2/3}$. Therefore, one way to determine the exponent x from our numerical data is to find the cartogram based on the density ρ^x that minimizes the variation of the Voronoi cell sizes on the cartogram. This approach does not suffer from the shortcomings of the previous method, since we neither use the Voronoi cells to calculate the population density nor take logarithms. One might argue that the Voronoi cells on the cartogram are not equal to the projections of the Voronoi cells in actual space. This is true; the cells generally will not even remain polygons under the cartogram transformation. The difference, however, is small if the density does not vary much

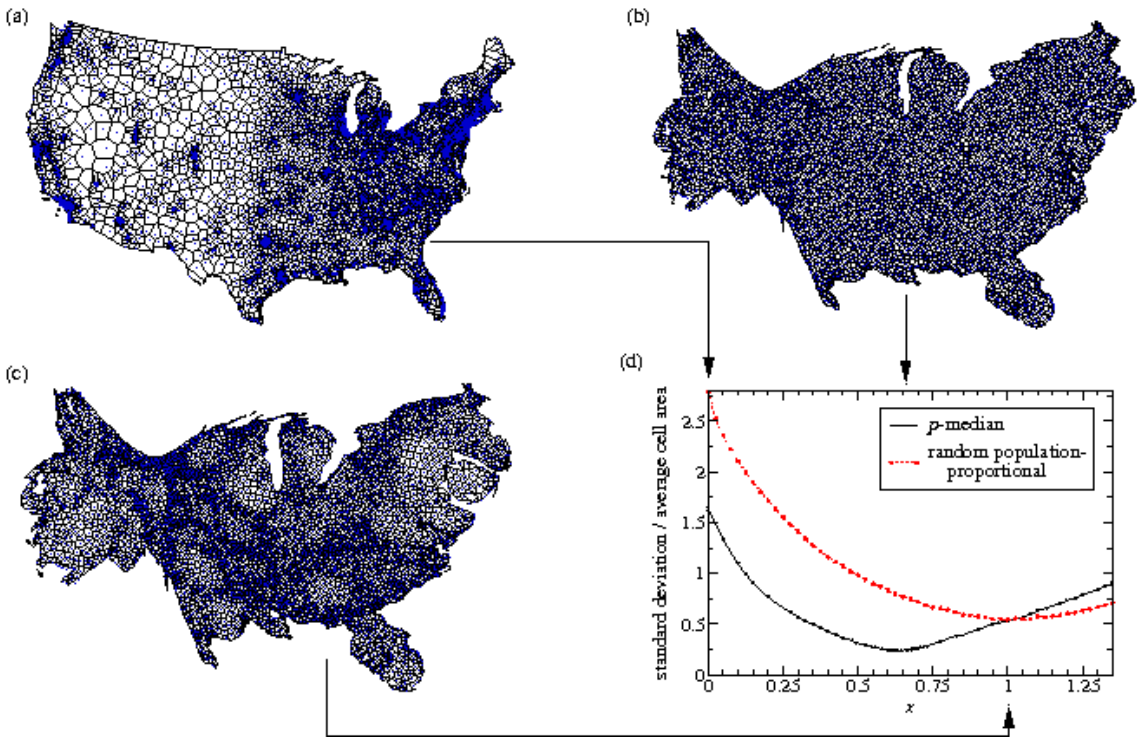

Figure 3.5: Near-optimal facility location and Voronoi tessellation on (a) an equal-area density, (b) a cartogram based on $\rho^{2/3}$, (c) a cartogram based on ρ. (d) The standard deviation in the distribution of Voronoi cell areas as they appear on a cartogram against the exponent x of the underlying density ρ^x.

between neighboring facilities. (This is, in fact, an assumption we already used in deriving Eq. (3.6).)

We pointed out in Sec. 2.4 that, to create a continuous population density, some level of coarse graining or blurring is necessary. Here we apply the Gaussian blur of Eq. (2.14) with a width $\sigma \approx 20$ km to the US population. This width is smaller than the average distance between two neighboring facilities for $p = 5000$. But the blur gets rid of the small-scale density fluctuations around big cities, where most facilities are, so that we can neglect the difference between projected and actual Voronoi cells.

The nearly-optimally located facilities are shown in Fig. 3.5a-c on an equal-area map and two cartograms based on this mildly blurred density for the exponents $x = \frac{2}{3}$ and $x = 1$. Not surprisingly, the points are not homogeneously scattered on the

equal-area map, 3.5a, since the population density is not uniform and more densely populated regions have more facilities than less populated ones. But the distribution is clearly not homogeneous on the equal-population-density cartogram 3.5c either; here we have overshot the mark since the points appear concentrated where the population is low. Fig. 3.5b, based on the expected facility density $\rho^{2/3}$, by contrast, appears to find the correct compromise between regions of high and low population density.

In Fig. 3.5d, we plot the standard deviation of the Voronoi cell sizes as they appear on the cartograms against the exponent x. The solid curve, obtained for the p-median facility distribution, indeed has a minimum near the predicted value $x = \frac{2}{3}$. For a comparison and as a further test of whether this technique determines the correct exponent, we have made the same measurement for 5000 points randomly distributed in proportion to population. Since the density of these points is by definition equal to ρ, we expect the minimum standard deviation in the cell areas to be at $x = 1$. The numerical result, the dotted curve in Fig. 3.5d, agrees with this prediction. Comparing the solid and the dotted curve, we find that not only the x-values of the minima differ, but also the minimal y-values. The lower standard deviation of the p-median distribution indicates that optimally located facilities are not *randomly* distributed with a density $\propto \rho^{2/3}$. Instead facilities occupy space in a rather regular fashion reminiscent of the hexagonal lattices in Central Place Theory [78].

3.5 Real facilities

Since we have now established that the optimal facility density for the p-median problem satisfies $D \propto \rho^{2/3}$, one might wonder if real facilities obey this rule. We take three examples: zip codes, hospitals and a national electronics store chain (RadioShack) in the lower 48 states. For the zip codes, we have looked up area and population in GIS databases; here we assume that zip code areas are roughly equal to the Voronoi cells around mail sorting centers. For the hospitals and stores, we looked up their

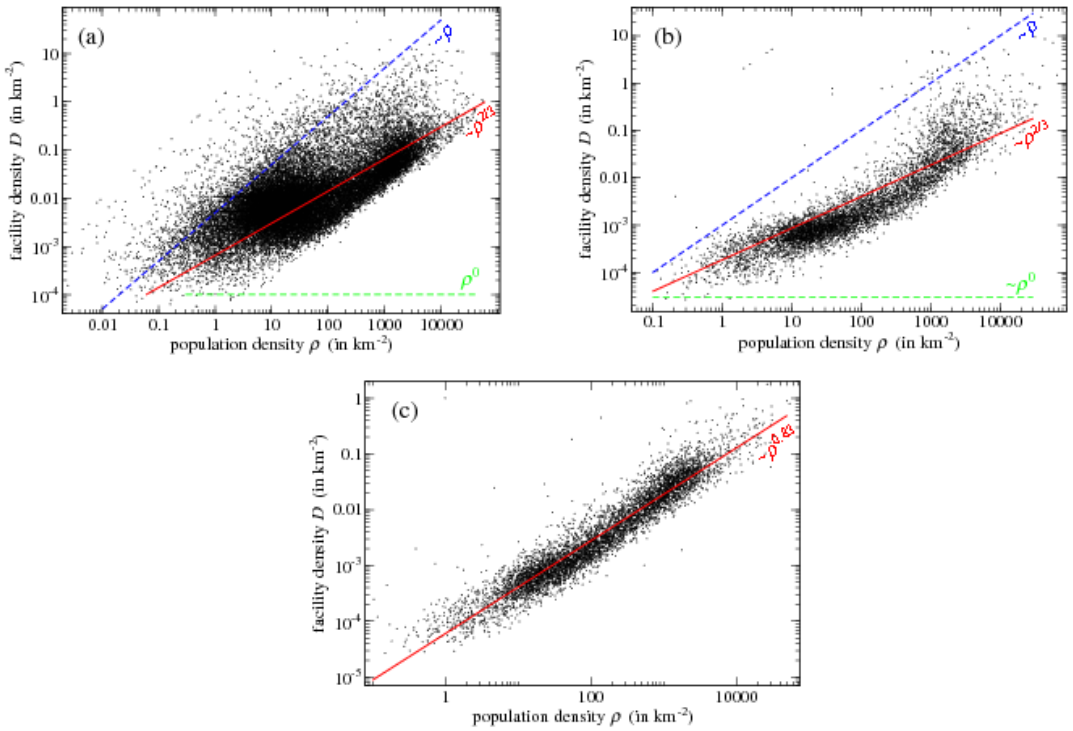

Figure 3.6: Density of real facilities versus population density. (a) Zip codes (approximately mail sorting centers). (b) Hospitals. (c) Electronics stores.

addresses in online directories, determined their geographic position in terms of longitude and latitude, and constructed the Voronoi tessellation. For data from the year 2005 there were 29 605 zip codes, 5734 hospitals, and 6064 stores.

In Fig. 3.6, we plot facility against population density on a double-logarithmic scale for our three examples. The zip code data (Fig. 3.6a), are so widely scattered that a line of slope $\frac{2}{3}$ is within the range of possibilities, but there is room for many other interpretations as well. Nevertheless, the alternative postulates of uniform distribution or population-proportional distribution (the two dashed lines) can be firmly ruled out. One explanation why the data is inconclusive might be the age of the zip code system. Since its introduction in 1963, the population distribution has changed and potentially randomized an earlier power law relation.

The hospital data (Fig. 3.6b) is less scattered, but is not well approximated by a simple power law. For low population density, the exponent appears to be less than $\frac{2}{3}$, whereas for high density it is closer to 1. This behavior might be caused by saturation effects. If the distance to the nearest hospital is too long, it is not of any use to a patient, and hence the hospital density can for practical reasons not go to zero even in nearly unpopulated areas. In highly populated areas, on the other hand, hospitals might be limited by the number of patients they can accept; no hospital can treat more than a few thousand at the most. If hospitals operate near saturation, the only way to serve all patients is to build more hospitals so that their density, in this limit, grows almost in proportion to the population. Hence, hospital locations are not optimal in terms of Eq. (3.1), but it is possible that such an optimization plays a role for medium densities. Stephan, who compiled a list of exponents for various facilities in 1987, in fact reports an exponent 0.68 for hospitals [79].

The data for the electronics stores (Fig. 3.6c) are less ambiguous since the points fall pretty much along a line. A least-squares fit yields a slope 0.83, almost exactly halfway between the expected exponents for a p-median distribution and an equal-population distribution, $\frac{2}{3}$ and 1. This finding confirms Stephan's observation that exponents near $\frac{2}{3}$ are observed for non-profit facilities such as post offices, hospitals, or county seats, but for business facilities the exponents are usually closer to 1.

Stephan and later Gusein-Zade [80] have proposed a modification of Eq. (3.1) that might explain exponents different from $\frac{2}{3}$, namely replacing $|\mathbf{r} - \mathbf{r}_i|$ by $|\mathbf{r} - \mathbf{r}_i|^\beta$. Repeating the calculation of Sec. 3.4, we are then led to $D \propto \rho^{2/(2+\beta)}$. Exponents $> \frac{2}{3}$ hence require $\beta < 1$, meaning that large distances are less penalized than in the p-median problem. For business facilities competing for customers, such as a store chain, this situation is not entirely plausible. The farther away the nearest store location is from a customer, the more likely is the competition to run a store closer to him, which should put larger distances at a rather over-proportional disadvantage.

Instead of tweaking Eq. (3.1), it would be more interesting to investigate the effects of competition between different facility types, but this would go beyond the scope of this dissertation.

3.6 Conclusion

We have in this chapter studied the p-median problem which consists of finding the positions of p facilities in geographic space such that the mean distance between customers and their nearest facility is minimized. Although obtaining exact solutions is practically not feasible, we have shown that a simple heuristic based on simulated annealing achieves good results improving the performance of a simple steepest descent algorithm.

Analytic arguments indicate that the density of facilities should be proportional to the population density to the two-thirds power. We have tested this relation by numerically solving the p-median problem for the population density of the conterminous United States using the simulated annealing heuristic. The thus calculated facility distribution confirms the algebraic relation between the two densities. Although real data provide a more complex picture, the p-median problem is the natural starting point for future investigations of facility location problems.

CHAPTER IV

SHAPE AND EFFICIENCY IN GROWING SPATIAL DISTRIBUTION NETWORKS

4.1 Introduction

In the previous chapter we studied the distribution of point-like facilities, such as post offices, hospitals, or electronics stores. In many cases, these facilities are not isolated points; instead they are interconnected as networks to transport goods and services between different facilities. Post offices, for example, deliver mail over a network of truck, rail and air links from one location to another. Power stations distribute electricity over a network of transmission lines and relay stations to households. Department stores receive their products from warehouses across the country via a network of truck deliveries. Libraries are networked to participate in interlibrary loan services for sharing rarely requested books. And the list goes on. The next logical step in our analysis of spatial distribution phenomena, therefore, will be to consider networked systems.

A network (or graph) is a set of points or *vertices* joined together in pairs by lines or *edges* (Fig. 4.1). Networks provide a useful framework for the representation and modeling of many technological, biological, and social systems, and have received a substantial amount of attention in the past few years. Reviews of recent developments can be found in [81, 82, 83]. The remainder of this dissertation deals with the special case of networks in which the vertices occupy particular positions in geometric space. Not all networks have this property—web pages on the world wide web, for example,

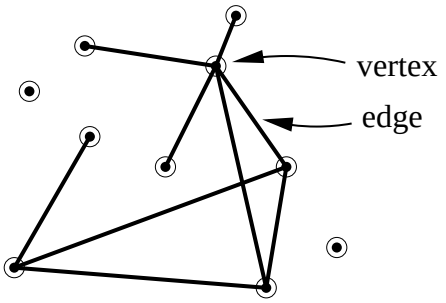

Figure 4.1: A small example network with ten vertices and nine edges.

do not live in any particular geometric space—but many others do. Examples include transportation networks, communication networks, and power grids. Although the study of spatial networks has a history of several decades [84, 85, 86], it has only come back into the limelight during the latest surge of network research. The past and current literature on spatial networks will be reviewed in Sec. 4.2.

In this chapter we study and model the spatial layout of man-made distribution or collection networks, such as oil and gas pipelines, sewage systems, and train or air routes. The vertices in these networks represent, for instance, households, businesses, or train stations and the edges represent pipes or tracks. The networks considered here also have a "root node," a vertex that acts as a source or sink of the commodity distributed—a sewage treatment plant, for example, or a central train station.

Geography clearly affects the efficiency of these networks. There are various possible definitions of efficiency and optimality [87, 88, 89]; in this paper we follow an idea put forward by Stevens [90]. A "good" distribution network, as we will consider it, has two definitive properties. First, the network should be efficient in the sense that the paths from each vertex to the root vertex are relatively short. That is, the sum of the lengths of the edges along the shortest path through the network should not be much longer than the "crow flies" distance between the same two vertices: if a subway track runs all around the city before getting you to the central train station, the train is probably not of much use to you. Second, the sum of the lengths of

all edges in the network should be low so that the network is economical to build and maintain. These two criteria are generally conflicting goals, but real networks nonetheless manage to find solutions to the distribution problem that come remarkably close to being optimal in both senses. We suggest possible explanations for this observation in the form of two growth models for geographic networks that generate networks of comparable efficiency to our real-world examples.

4.2 Literature review

There has in the last few years been considerable interest, mostly from the statistical physics community, in the analysis and modeling of networked systems. Some of the networks studied, such as scientific citation networks [91], protein interaction networks [92] and the world-wide web [93, 94], exist only in an abstract "network space" where the precise positions of the vertices have no particular meaning. But many others exist in literal physical space with vertices having well-defined positions. The edges in these networks are often real physical constructs, such as roads or railway lines in transportation networks [95], optical fiber or other connections in the Internet[1] [96], cables in a power grid [97], or pipes in a water distribution network [98]. In other cases the edges may be more ephemeral, such as flights between airports [99], business relationships between companies [100], or wireless communications [101].

Interest in the spatial structure of networks dates back to the quantitative geography movement of the 1960s and 70s [84, 102, 86, 103, 104, 105, 106] and particularly the work of Kansky [85]. Early work was hampered however by limited data and computing resources, and geographers' attention moved on after a while to other

[1]Colloquially, the terms "Internet" and "World Wide Web" are often used synonymously. From a technological perspective, however, they are completely different objects. While the Internet is a physical network of computers, routers and fiber cables, the World Wide Web is a network of web sites connected by hyperlinks. The Internet is hence a spatial network, whereas the World Wide Web is not.

topics. Networks are now again on the scientific forefront, but spatial aspects have received comparatively little attention. Empirical studies of networks, even networks in which geography plays a pivotal role, have, with some exceptions, focused mostly on topological, i.e. non-geographic, features [96, 107, 95]. Similarly, the best-known theoretical models of networks either make no reference to space at all [108, 109], or they place vertices on simple regular lattices whose structure is quite different from that of real systems [110, 111].

Recently, several models have been proposed that incorporate more sophisticated geographic features, but empirical geographic data against which one might verify such models is still mostly lacking. These models fall roughly speaking in two categories. In the first category, all vertices and edges are present from the beginning, whereas in the second category the network grows as time progresses.

4.2.1 Non-growing spatial network models

Many of the commonly studied graphs in computational geometry belong to this first category, such as the minimum spanning tree, random geometric (or disk) graph, Delaunay graph, k-nearest-neighbor graph, and sphere-of-influence graph (Fig. 4.2). The definitions are as follows.

- The *minimum spanning tree* (MST) on a set of n given vertices is the set of $n-1$ edges joining them such that all vertices belong to one single component and the sum of the lengths of all edges is minimized. In other words, the MST is the shortest of all connected networks on the given set of vertices.
- The *random geometric graph* contains all possible edges shorter than a certain distance d_{lim}, but no edges larger than d_{lim} [112, 113, 114]. Random geometric graphs are often used as a model for wireless communication networks [115, 116]. Depending on d_{lim}, the graph can consist of several disconnected components. The size of the components as a function of d_{lim} is of interest for continuum

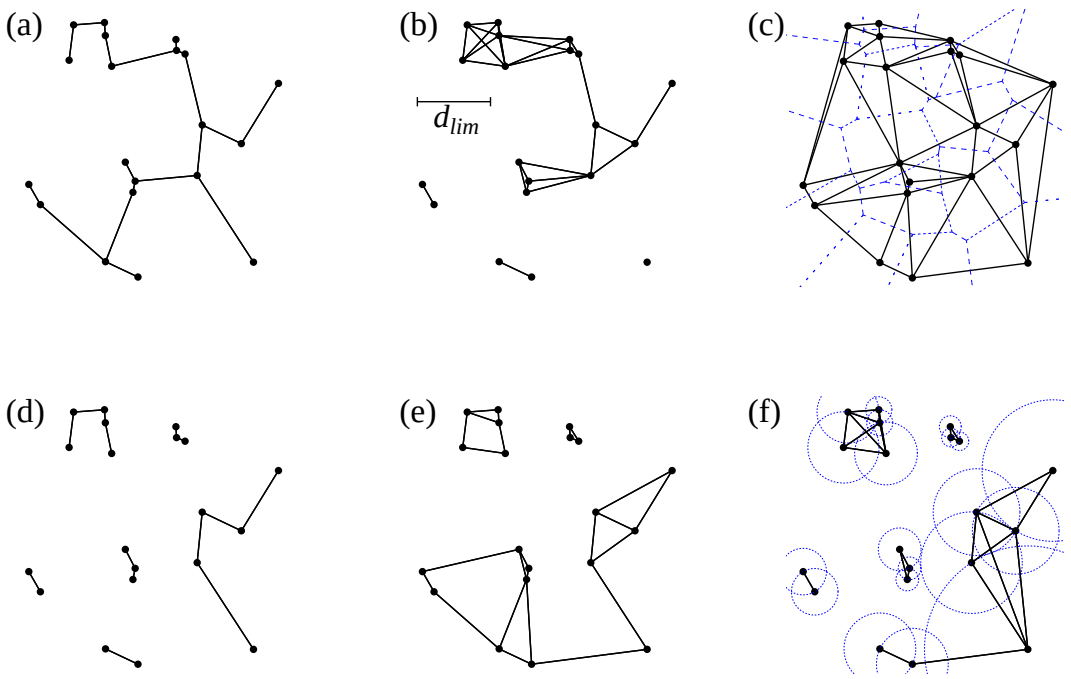

Figure 4.2: Some commonly studied geometric graphs. (a) Minimum spanning tree. (b) Random geometric graph. (c) Delaunay graph. The dashed lines indicate the Voronoi tessellation. (d) Nearest neighbor graph. (e) 2-nearest neighbor graph. (f) Sphere-of-influence graph. The circles represent the boundaries of the spheres of influence.

percolation models [117, 118, 119].

- The *Delaunay graph* is based on the Voronoi tessellation of the vertex set. All vertices belonging to adjacent cells are connected [120]. Delaunay graphs always contain one single component, and none of the edges intersect. The MST is a subgraph of the Delaunay graph [121].

- In the *k-nearest neighbor graph*, the edge between two vertices i and j exists only if i is among the k vertices closest to j or if j is among the k vertices closest to i. The special case $k = 1$, also simply known as nearest neighbor graph, is a subgraph of the MST.

- The *sphere-of-influence graph* is constructed in the following manner. If the vertices have positions $\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n$, the ball around each vertex x with radius $\min_{y\in\{1,2,...,n\}\setminus x} d_{xy}$, where d_{xy} is the Euclidean distance between the vertices x

and y, is called x's sphere of influence. This is in other words the smallest ball around x that contains (at least) one vertex other than x. Now connect all vertices whose spheres of influence overlap to obtain the sphere-of-influence graph.

These graphs, although of considerable interest for the mathematics and computer algorithms community, are only rarely suitable models of real networks. Recent network research has, for example, shown the existence of highly connected vertices in many different networks. The number of edges k_i attached to a vertex i is called the *degree* of i. For many real networks, the probability distribution for a vertex to have degree k is very broad, often approximately following a power law $p(k) = k^{-\theta}$. The models above, however, have narrow degree distributions, unless the vertex positions are rather pathologically distributed [122, 123]. Rozenfeld *et al.* [124] and independently Warren *et al.* [125] therefore proposed a generalization of the k-nearest neighbor graph, where the degree of every vertex is randomly drawn from a power law distribution.

Besides heavy-tailed degree distributions, a commonly found feature of real networks is the "small-world" property: No pair of vertices is separated by more than a small number of edges. The minimum number of edges along a path between two vertices is called their "graph distance". (It should not be confused with the geometric distance, see Fig. 4.3.) On the other hand, most connections are made on a local level in the sense that the probability for a connection between i and j is high if both i and j are connected to one common neighbor k, a property known as "clustering". Watts and Strogatz explained this phenomenon with a model where vertices first make connections to their nearest neighbors, and subsequently a small number of longer edges are added to the network [110]. These additional edges create "shortcuts" that can reduce the average graph distance significantly. In their original model, shortcuts were chosen uniformly at random. In spatial networks, on the

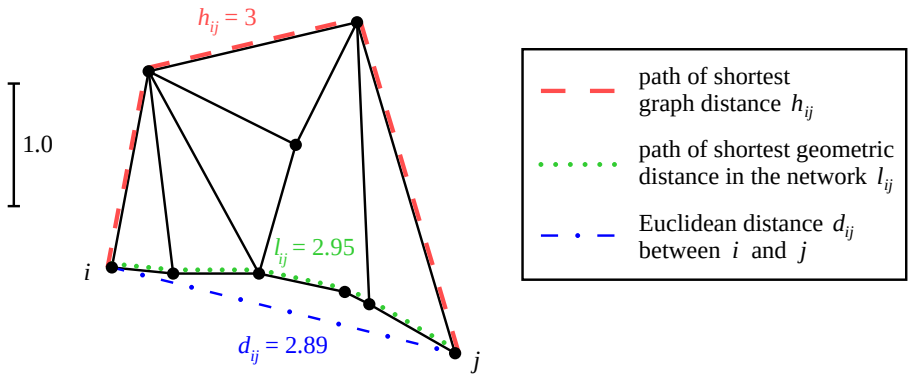

Figure 4.3: The distance between two vertices i and j in spatial networks can be measured in different ways. The Euclidean or "crow flies" distance d_{ij} is measured along a straight line between the vertices. If there is no edge between i and j, the effective geometric distance l_{ij} in the network will generally be longer. The shortest geometric path does not need to be the shortest path in terms of the number of edges. In the network above the geometrically shortest route (dotted) traverses five edges, whereas the dashed path only consists of three. The smallest number of edges along any path between i and j is called "graph distance". Here we denote it h_{ij}.

other hand, a shortcut between two vertices i and j is less likely if their geometric distance d_{ij} is large. Several authors have investigated shortcut probabilities that decay as $\propto {d_{ij}}^{-\mu}$. Kleinberg [111] studied navigation strategies on such networks and found that, in two dimensions, for $\mu = 2$ efficient navigation from vertex to vertex is possible even without global knowledge of all edges in the network. Sen *et al.* [126] analyzed the average path length and clustering coefficient in one dimension, whereas Petermann and De Los Rios [127] focused on wiring costs and robustness as functions of μ.

Many of the models analyzed by operations researchers also fall under the category of non-growing spatial networks. There, the objective is the design of a network with optimum functionality, for example measured by cost or travel time. A survey of typical problems can be found in [128]. One particular example is the "Optimal Network Problem" posed by Scott [12] and later modified by Billheimer and Gray [129]. This model is central to the next chapter and will be introduced there.

4.2.2 Growing spatial network models

In many networks, neither the positions of vertices nor the connections between them are initially known. Instead, which vertices and edges are to be added to the network might be decided at a later stage. Examples are subway networks that slowly expand over decades, or power lines that are extended to newly developed neighborhoods. Barabási and Albert have identified growth as an important ingredient in modeling non-spatial networks [109]. They have shown that preferentially attaching new vertices to already existing highly connected vertices reproduces the power law degree distributions frequently seen in real networks.

In spatial networks, the distance between vertices will also play a role; nearby vertices are more likely to be connected than vertices far apart. A generalization of the Barabási-Albert model that takes this distance into account works as follows. We begin with a small network of m vertices containing all $m(m-1)/2$ possible edges between them. (Such a network is called a *complete graph*.) Then, we repeatedly add one new vertex i at a random position and connect it with m edges to previously placed vertices. The probability $P(i,j)$ that one of these edges connects i to j is assumed to be proportional to

$$P(i,j) \propto {k_j}^{\lambda} f(d_{ij}) \tag{4.1}$$

where k_j is the degree of j, d_{ij} the distance between i and j, λ a non-negative real number, and f a monotonically decreasing function.

Several recent papers investigated this model (or slight variations) for specific values of λ and specific functions f. Yook *et al.* [130], for example, chose a polynomial $f(d_{ij}) = {d_{ij}}^{-\mu}$ claiming that the model for $\lambda = 1$ and $\mu = 1$ reproduces essential features of the Internet. However, Lakhina *et al.* [131] convincingly showed that a better agreement with actual geographic data on the Internet is achieved by an exponential $f(d_{ij}) = \exp(-\nu d_{ij})$ with $\nu \approx 200$ km. This latter case is also known as

the Waxman model [132], and is used by the Internet topology generator BRITE [133]. (Internet topology generators are programs simulating the structure of the Internet. They are important for optimizing Internet software and protocols.)

Further studies differ mostly by choosing f either as a polynomial or exponential. Sen and Manna [134, 135] studied the first case for various values of λ and μ, and identified a phase transition in the degree distribution. On one side of the transition, the distribution is a stretched exponential, on the other side, one vertex is connected to all other vertices, a phenomenon called "gelation". Along the phase boundary the degree distribution is a power law. Xulvi-Brunet and Sokolov, although placing vertices on a lattice and not in continuous space, report similar results for $\lambda = 1$ [136].

Barthelemy [137], by contrast, investigated the Waxman model and the dependence of network properties, such as degree distribution and clustering coefficient, on the parameter ν. In order to match empirical data of airport networks, Barrat *et al.* [138] modified the model by introducing an additional factor in Eq. (4.1) for each vertex, which the authors refer to as "strength". Kaiser and Hilgetag [139], on the other hand, simplified the model by setting $\lambda = 0$, so that there is no dependence on the degree k_j. They also drop the constraint that exactly m edges are added at each step; instead all possible connections are investigated, and, if the new vertex could not establish any connections, it is removed.

A fundamentally different model was put forward by Fabrikant *et al.* [140]. Vertices are again added one after another at random positions. Let us call the first placed vertex 0. The i-th vertex in this model no longer preferentially attaches to high degrees, but to the previous vertex j that minimizes a weighted sum of the Euclidean distance between i and j, d_{ij}, and graph distance from j to 0, h_{j0}:

$$\min_{j<i} \xi d_{ij} + h_{j0}, \quad \xi \geq 0 \tag{4.2}$$

The model is motivated by the Internet, where d_{ij} represents the "last-mile" cost

to connect a new user to the network, and h_{j0} captures the operation cost due to communication delays. Originally, the model was believed to produce a power law degree distribution similar to the one observed for the Internet for certain values of ξ. However, Berger *et al.* [141] proved that this is not true because the majority of the vertices have degree one, and another substantial fraction has very high degree so that the distribution is bimodal and only follows a power-law between the two peaks. The model is, nevertheless, fascinating in its own right and has inspired Berger *et al.* [142], and Alvarez-Hamelin and Schabanel [143] to similar ideas.

4.2.3 Other growth models

Physicists' interest in growth models predates their work on networks by several decades. Here we mention three well-studied models—invasion percolation, diffusion-limited aggregation and the Eden model—which, although not directly related to networks, are relevant in the present context since they also deal with growing spatial objects.

In *invasion percolation*, bonds in a square lattice are assigned random weights from a uniform distribution. Starting from a set of initial seeds, for example sites on the left-hand side of the lattice, a cluster is built by repeatedly adding the edge on the cluster's surface that has the smallest weight [144, 145]. The process ends when the cluster first reaches the opposite end of the lattice. The model simulates the displacement of one fluid by another, for example oil by water, in a porous medium. The random weights in the model represent pore diameters, and the dynamics of the growing cluster mimics the fact that the invading fluid advances first along the narrowest pore necks where the capillary force is strongest. The final cluster, Fig. 4.4(a), is geometrically self-similar, i.e. zooming in on a smaller portion of the cluster we find features typical of the cluster as a whole. In particular, there are "holes" ranging from micro- to macroscopic length scales. Because of these gaps, it would be inappropriate

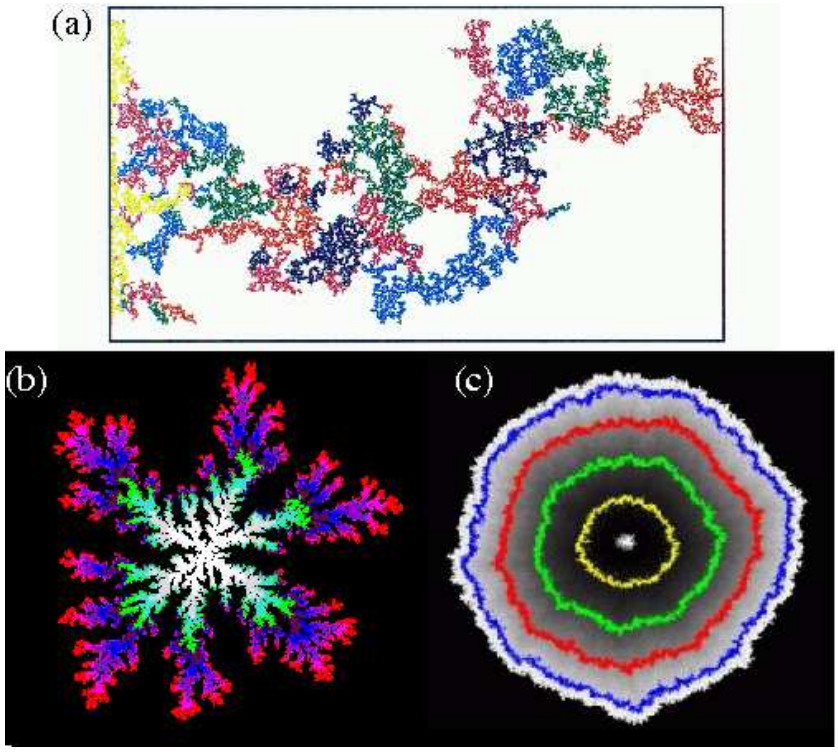

Figure 4.4: Clusters obtained by different growth models. (a) Invasion percolation. Reprinted with permission from [9], ©1988 The American Physical Society. (b) Diffusion-limited aggregation. Reprinted with the author's permission from [10]. (c) Eden model [11]. Colors in (a) and (b) represent when particles joined the cluster, in (c) they indicate the shape of the cluster at different points in time.

to assign a dimension of 2 to these clusters because the available two-dimensional space is not effectively filled during the growth process. Instead, it turns out that invasion percolation produces clusters with a *fractal dimension* between one and two. We will define this concept more precisely in Sec. 4.3. Here it shall suffice to state the literature value, 1.89, which is incidentally also the fractal dimension of the spanning cluster of ordinary percolation at the critical point [146].[2]

A very different class of fractals is obtained by *diffusion-limited aggregation*, first introduced by Witten and Sander [147, 148]. After a seed particle is placed at the

[2]The picture presented here becomes considerably more complicated if the defending fluid is incompressible, see [146].

central site of a square lattice, another particle is launched far from the center performing a random walk on the lattice until it reaches a site adjacent to the seed. There the particle stops and a new particle is launched which randomly walks until it touches one of the two previous particles, and so forth. Big clusters grown in this fashion, like the one in Fig. 4.4(b), have a characteristic "dendritic", i.e. tree-like branching structure. This shape is due to a "screening" effect: A diffusing particle is more likely to stick to the tip of a branch than to walk into one of the "fjords". Originally intended as a model for growth in colloids or coagulated aerosols, it has been used in a variety of other contexts from bacterial colonies [149] to urban sprawl [150].

Similar, but leading to yet again completely different aggregates, is the *Eden model* [151]. Starting from a central seed site on a square lattice, one of the unoccupied sites adjacent to the current cluster is chosen randomly and added to the cluster. Since there is no screening effect as before, the resulting clusters are compact and nearly spherical (Fig. 4.4(c)). The surface however has non-trivial properties; careful analyses have revealed self-affine scaling of its roughness [152]. Like diffusion-limited aggregation, the Eden model has found applications in many areas such as solid state physics [153], tumor growth [154] and human settlement patterns [155]. More details on fractal growth models can be found in [156].

4.3 A network growth model with minimum total length

In order to develop a model of a spatial distribution network, we will assume that the only cost in building a network is the total length of its edges. The cheapest of all possible networks then is the MST.[3] The construction of the MST requires that we

[3]If we are not restricted to the specified vertex set, but are allowed to add vertices freely, then the optimal solution is the Steiner tree. The construction of a Steiner tree is an NP-hard problem [157], and hence not feasible for all but very small networks. The total length of the MST can be proved to be in the worst case only a factor $2/\sqrt{3} \approx 1.15$ more than that of the Steiner tree [158], and all

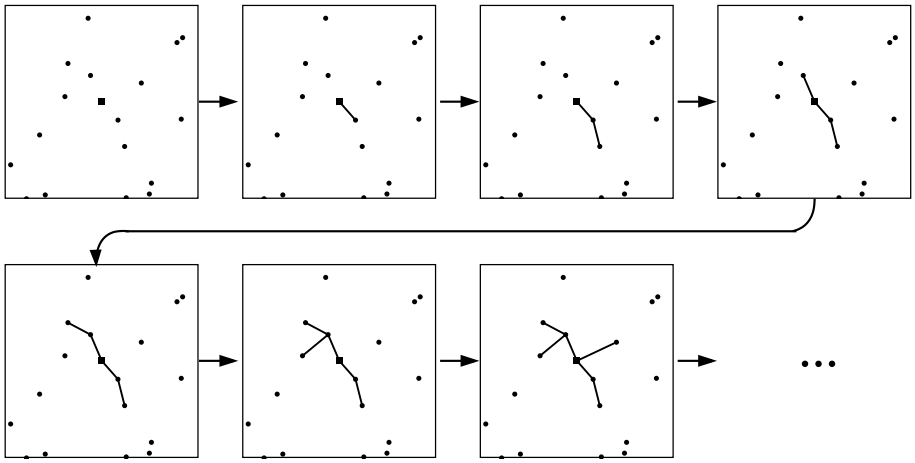

Figure 4.5: The first few steps during the construction of a growing minimum spanning tree. At the beginning, only the root vertex (square) is part of the tree. Then we repeatedly add the shortest edge between one connected and one unconnected vertex.

know which vertices will be included in the network. However, if the network forms by growing outward from a root vertex as the population swells and infrastructure is extended and improved, the number of vertices in the final network cannot be predicted from the start. Instead we study a variation of the static MST which might be called a "growing" or "invading" minimum spanning tree [159].

We assume that at the beginning we are given the positions of the root vertex, e.g. an oil well or a central train station, and of several other points, e.g. houses or towns, that are candidates to join the network. A cluster connected to the root is built up by repeatedly adding the shortest edge that joins one unconnected vertex to another that is part of the cluster. In Fig. 4.5, the first few steps of this growth process are shown. This algorithm is "greedy" in the sense that at each step the added edge is the one that increases the network length by the smallest possible amount. Prim's algorithm for constructing static MSTs uses in fact the same strategy. Hence, if we keep adding the shortest connection until all vertices are connected, there is no difference between growing and static MST. Here, however, we assume that the number of candidate

results in table 4.1 of Sec. 4.4 remain practically unchanged if Steiner trees were used instead of MSTs.

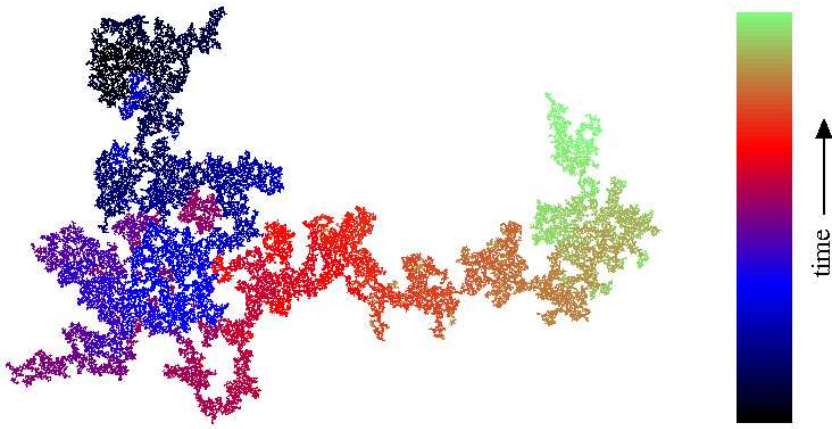

Figure 4.6: Growing minimum spanning tree with 100 000 vertices.

vertices is infinite so that the growth continues indefinitely.

Fig. 4.6 shows a large growing MST on a random point distribution. (We have normalized the length scale by setting the average density equal to one.) Colors indicate when the edges were added. The similarity with the invasion percolation cluster of Fig. 4.4(a) is noteworthy. The construction of the network is in fact closely related to invasion percolation. Although the vertices in our model do not lie on a regular lattice, we can think of the Delaunay graph as some kind of distorted lattice [160]. Since every possible edge in the MST belongs to the Delaunay graph, the growing MST can be mapped onto an invading percolation cluster on this distorted lattice. However, unlike invasion percolation, the weights are neither uniformly nor independently distributed. For a random point pattern, distances follow a Poisson distribution and, furthermore, the weight of each edge is equal to its Euclidean length. This last fact introduces quite complicated dependences between the different weights making a rigorous mathematical treatment difficult, but there are good reasons to conjecture that the growing Euclidean MST falls in the same universality class as ordinary percolation.

To support this conjecture, we have calculated the fractal dimension of the set of vertices in the tree as well as the fractal dimension of its external perimeter. The

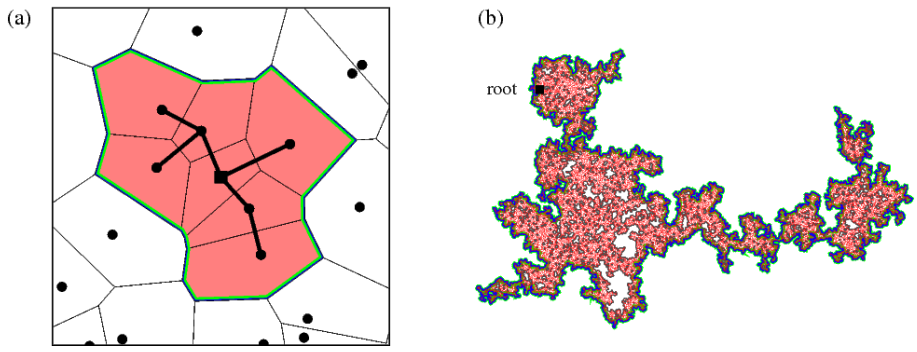

Figure 4.7: (a) The Voronoi cells of the tree in the last panel of Fig. 4.5. All cells containing a connected vertex are marked red and the perimeter is highlighted in green. (b) The interior and perimeter of the tree of Fig. 4.6.

perimeter is defined by constructing the Voronoi cells for all vertices, including those that are unconnected. Every cell containing a connected vertex is considered as filled, all others as empty. The external perimeter divides the filled Voronoi cells from the unbounded empty exterior, see Fig. 4.7(a). A typical arrangement of filled Voronoi cells for a large network on a random point pattern is shown in Fig. 4.7(b). The interior contains numerous holes of various sizes and is bounded by a complex curve with "fjords" reaching deep into the cluster.

One of many (more or less equivalent) ways to define a fractal dimension is based on density correlations. First, we fix a number of vertices $v_1, v_2, ..., v_k$ belonging to the tree. Then we count the number of connected vertices in the tree inside a ball of radius ϵ about v_i, $N_i(\epsilon)$. The sum of these numbers is expected to grow as ϵ^d in d dimensions. Hence, we can calculate d as the derivative

$$d = \frac{\mathrm{d}\log\left(\sum_{i=0}^{k} N_i\right)}{\mathrm{d}\log(\epsilon)}. \tag{4.3}$$

The value of d obtained in this manner is the *correlation dimension* [161]. For points on an n-dimensional lattice we retrieve the expected result $d = n$. However, generally d does not need to be an integer; in this latter case, the network is a *fractal* and d is called the *fractal dimension*.

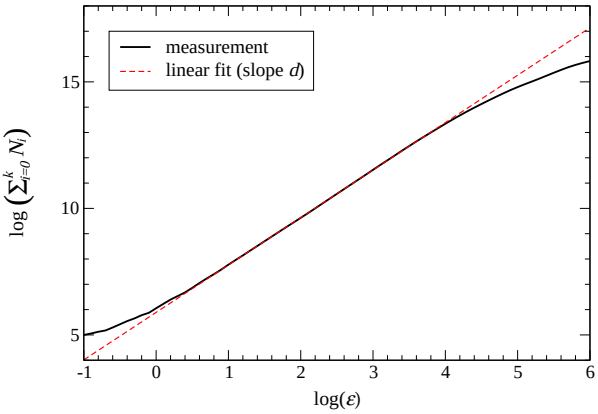

Figure 4.8: Typical measurement of $\log\left(\sum_{i=0}^{k} N_i\right)$ versus $\log(\epsilon)$ for the interior of a growing MST with $k = 50$ and 100 000 vertices. In the scaling regime ($1 \leq \epsilon \leq 4$), the slope of a linear fit is approximately equal to the network's fractal dimension d.

Strictly speaking, the above derivative depends on ϵ, as a typical example in Fig. 4.8 shows. For very small ϵ, the only point in each ϵ-ball is the center itself, hence the slope becomes zero. The same happens for large ϵ because each ϵ-ball encloses the whole tree. Only for intermediate values of ϵ can we expect the slope to be a good estimate of the fractal dimension. We have therefore calculated d as the slope in the region of the steepest descent. We tested this approach on point patterns of known dimensions such as lines or areas and found this method to work very well if an average over several measurements is taken. Based on networks with 100 000 vertices each, we find a fractal dimension of $D_{netw} = 1.89 \pm 0.02$ which is the same as for invasion percolation clusters.

Applying the same technique to the external perimeter, we determined its fractal dimension to be $D_{per} = 1.36 \pm 0.03$. Hence, the perimeter is not a simple one-dimensional line because it extends into all the fjords along the boundary. The fractal dimension is again consistent with that of the external accessible perimeter of invasion percolation clusters, 4/3, which supports our conjecture that growing MSTs

belong to the same universality class.[4] In Sec. 4.5 and 4.6 we will analyze how simple modifications of the presented growth model lead to quite different fractal geometries.

4.4 The efficiency of real networks

Let us now compare the properties of MSTs with some real-world distribution networks. We consider four examples as follows. The first network is the sewer system of the City of Bellingham, Washington. From GIS data for the city, we extracted the shapes and positions of the parcels of land (roughly households) into which the city is divided and the lines along which sewers run. We constructed a network by assigning one vertex to each parcel whose centroid was less than 100 meters from a sewer. The vertex was placed on the sewer at the point closest to the corresponding centroid and adjacent vertices along the sewers were connected by edges. The city's sewage treatment plant was used as the root vertex, for a total of 23 922 vertices including the root.

Our next two examples are networks of natural gas pipelines, the first in Western Australia (WA) and the second in the southeastern part of the US state of Illinois (IL) [5]. We assigned one vertex to each city, town, or power station within 10km (WA) or 10,000 feet (IL) of a pipeline. The vertex was placed on the pipeline at the point closest to each such place, and adjacent vertices were joined by edges. The root for WA was chosen to be the shore point of the pipeline leading to the Barrow Island oil fields and for IL to be the confluence of two major trunk lines near the town of Hammond, IL. The resulting networks have 226 (WA) and 490 (IL) vertices including

[4]However, the fractal dimension of an invasion percolation cluster's hull is bigger, namely 7/4 [146]. There is no obvious geometric construction for growing MSTs which has this fractal dimension.

[5]South of 41.00°N and east of 89.85°W. We consider only the largest component within this region.

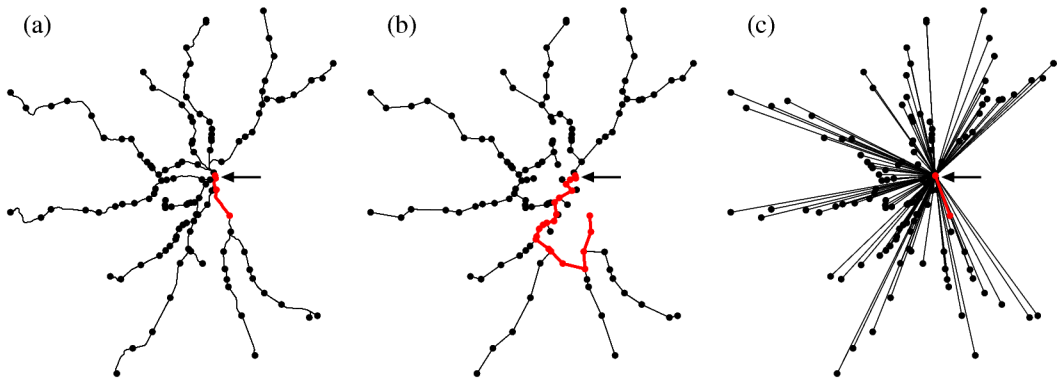

Figure 4.9: (a) Commuter rail network in the Boston area. (b) Minimum spanning tree. (c) Star graph. The arrow marks the assumed root of the network. Paths in the real network, like the one highlighted in red, are more direct than in the minimum spanning tree. In fact they are almost as short as the straight line connections of the star graph.

the roots.

For our last example we take the commuter rail system operated by the Massachusetts Bay Transportation Authority in the city of Boston, MA (Fig. 4.9a). In this network, the 125 stations form the vertices and the tracks form the edges. In principle, there are two components to this network, one connected to Boston's North Station and the other to South Station, with no connection between the two. Since these two stations are only about one mile apart, however, we have, to simplify calculations, added an extra edge between the North and South Stations, joining the two halves of the network into a single component. The root node was placed halfway between the two stations for a total of 126 vertices in all.

In Table 4.1, we show the total edge lengths for each of our networks, along with the edge lengths for the MST on the same set of vertices. We find that the real-world networks, although not strictly optimal, are quite competitive with MSTs. The combined edge lengths of the real networks range from 1.12 to 1.63 times those of the corresponding MSTs. But even though the MST on the set of stations in the Boston commuter rail network, Fig. 4.9(b), is merely 11% shorter, it does not resemble the real network much. Looking at some of the paths in Fig. 4.9(a) and (b), we find that

network	n	edge length (km) actual	MST	star	route factor actual	MST	star
sewer system	23 922	498	421	102 998	1.59	2.93	1.00
gas (WA)	226	5 578	4 374	245 034	1.13	1.82	1.00
gas (IL)	490	6 547	4 009	59 595	1.48	2.42	1.00
rail	126	559	499	3 272	1.14	1.61	1.00

Table 4.1: Number of vertices n, total edge length, and route factor q for each of the networks described in the text, along with the equivalent results for the star graphs and minimum spanning trees on the same vertices.

the paths to the root tend to be more circuitous in the MST than in reality. Hence, passengers would have to take quite long journeys in the MST, even from stations whose "crow flies" distance to the root is rather short. These circuitous routes would of course be annoying, and a good reason why one would not build the MST, but some different network.

To make a comparison between MSTs and real networks, we consider the distance from each non-root vertex to the root first along the edges of the network and second along a simple Euclidean straight line, and calculate the mean ratio of these two distances over all such vertices. Following Ref. [162], we refer to this quantity as the network's *route factor*, and denote it q:

$$q = \frac{1}{n} \sum_{i=1}^{n} \frac{l_{i0}}{d_{i0}}, \tag{4.4}$$

where l_{i0} is the distance along the edges of the network from vertex i to the root (which has label 0), and d_{i0} is the direct Euclidean distance. (A similar measure "tortuosity" was used in [159].) If there is more than one path through the network to the root, we take the shortest one, but, since our examples are all nearly trees, most paths are unique. Thus, for example, $q = 2$ would imply that on average the shortest path from a vertex to the root through the network is twice as long as a direct straight-line connection.

The smallest possible value of the route factor is 1, which is achieved by the "star

graph", Fig. 4.9(c), in which every vertex is connected directly to the root by a single straight edge. The route factors for the four real networks are shown in Table 4.1. As we can see, paths in the networks, although not perfectly straight, are in most cases not far away from simple straight lines, with route factors quite close to 1. Actual values range from $q = 1.13$ for the Western Australian gas pipelines to $q = 1.59$ for the sewer system. Furthermore, the route factors in all real networks are consistently better than those in the MSTs.

But now consider the column in the table, which gives the the total edge lengths for the star graphs. These figures are for all networks much larger than the optimal case and, more importantly, much poorer than the real-world networks too. Thus, although the MST is optimal in terms of total edge length it is very poor in terms of route factor and the reverse is true for the star graph. Neither of these models would be a good general solution to the problem of building an efficient and economical distribution network. Real-world networks, on the other hand, appear to find a remarkably good compromise between the two extremes, possessing simultaneously the benefits of both the star graph and the minimum spanning tree, without any of the flaws. In the remainder of this chapter we consider two mechanisms by which this might occur.

4.5 A network growth model with low route factor

Our first attempt to explain these observations is a modification of the growing MST of Sec. 4.3. Like the growing MST, this modified model will build a growing network based on a "greedy" optimization criterion that always adds the current best candidate edge. However, now the best candidate is not simply the shortest edge; instead the route factor will become part of the decision about which vertex to add next.

This is simply done by specifying a weight for each edge (i, j) thus:

$$w_{ij} = d_{ij} + \alpha \frac{d_{ij} + l_{j0}}{d_{i0}}, \tag{4.5}$$

where α is a non-negative independent parameter. As before, d_{ij} is the direct Euclidean distance between vertices i and j and l_{ij} the distance along the shortest path in the network. The first term in (4.5) is the length of the prospective edge, which represents the cost of building the corresponding pipe or track, and the second term is the contribution to the route factor from vertex i. At every step we now add to the network the edge with the global minimum value of w_{ij}. The single parameter α controls the extent to which our choice of edge depends on the route factor. For $\alpha = 0$, the network is a growing MST, which we found to give unrealistically high route factors. As α is increased from zero, however, the model becomes more and more biased in favor of making connections that give good values for the route factor.

For simplicity we will, as we already did in Sec. 4.3, assume that the vertices are randomly distributed in two-dimensional space with unit mean density and with one vertex randomly designated as the root of the network. The inset of Fig. 4.10 shows a network grown in this manner for $\alpha = 12$. The network has a dendritic appearance, with relatively straight trunk lines and short branches, bearing a superficial resemblance to diffusion-limited aggregation clusters, although they are based on entirely different mechanisms.

In Fig. 4.10 we also plot the route factor q of the network and the average length of an edge $\bar{l}$ against α. As α is increased, the route factor does indeed go down in this model, just as we expect. Furthermore, it decreases initially very sharply with α, while at the same time $\bar{l}$, which is proportional to the cost of building the network, increases only slowly. Thus, it appears to be possible to grow networks that cost only a little more than the optimal ($\alpha = 0$) network, but which have far less circuitous routes. This finding fits well with our observations of real distribution networks.

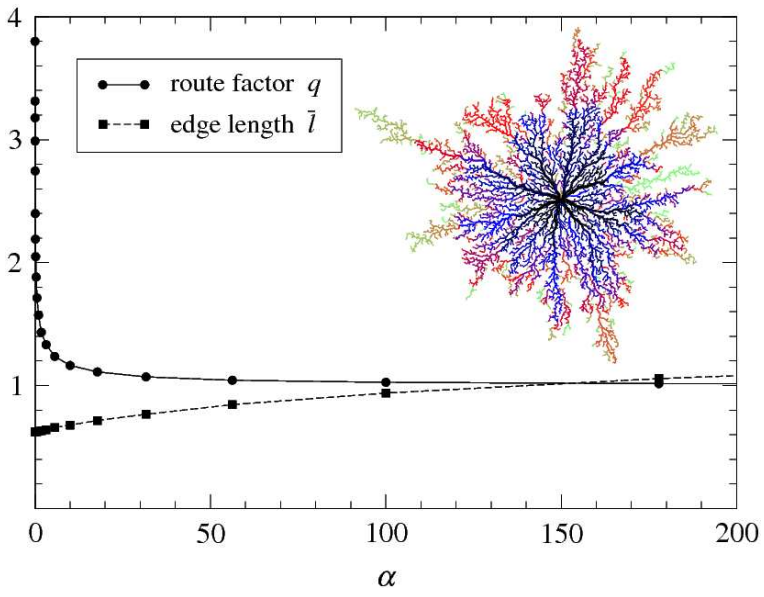

Figure 4.10: Simulation results for the route factor q and average edge length $\bar{l}$ as a function of α for our first model with 10 000 vertices. The length scale is normalized by setting the mean density equal to one. Inset: an example model network with $\alpha = 12.0$. Colors indicate the order in which edges were added to the network.

The transition from the rather amorphous shapes of a growing MST to a dendritic pattern is best studied by looking at the arrangement of filled Voronoi cells. Fig. 4.11 shows the patterns for different values of α. Clearly, the shape becomes more centered about the root (marked by the black square) and the holes become smaller and are pushed towards the periphery of the network as α increases. Assuming that only the filled Voronoi cells will be close enough to the network to attract human settlement, our findings for $\alpha = 2.4$ and $\alpha = 24$ are consistent with Benguigui's observation that cities are more compact around their centers than around their periphery [155]. One way to quantify how much better the network fills the surrounding space is by looking at the fractal dimension. Since the holes tend to move to the periphery, the fractal dimension is not necessarily uniform everywhere, but we can obtain meaningful results by measuring the fractal dimension only for the core region. We define the core by first determining the maximum distance between the root and any point on the perimeter, and then eliminating everything farther away from the root than half that

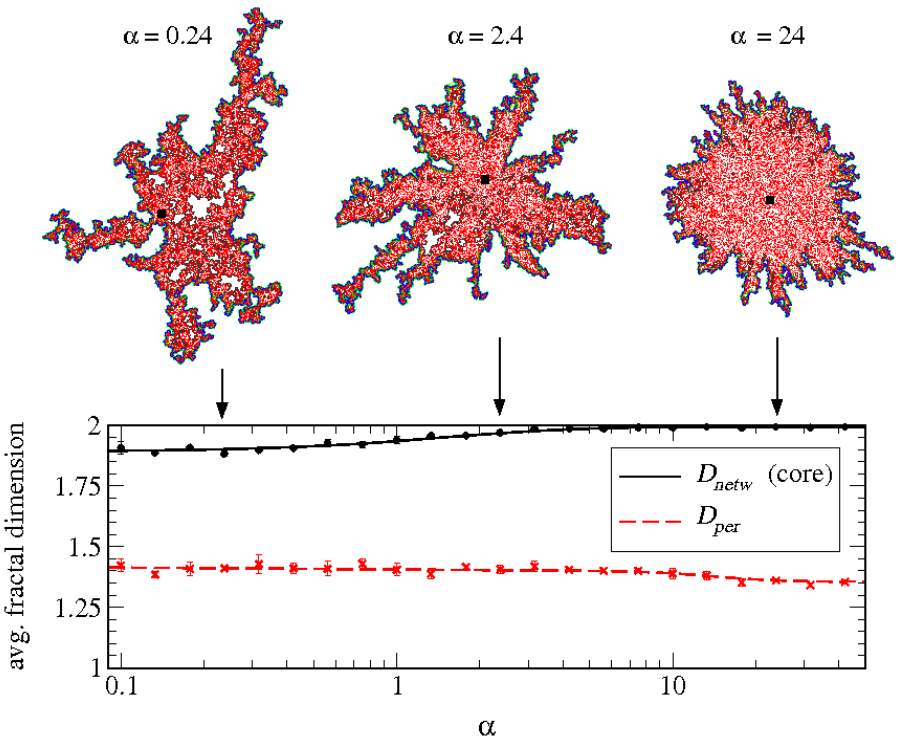

Figure 4.11: Top: The filled Voronoi cells of networks for three different values of α. The black square marks the position of the root. Bottom: The fractal dimension of the network D_{netw} and its perimeter D_{per} as a function of α measured in networks with $50,000$ vertices each.

distance. The plot in Fig. 4.11 shows a smooth transition from a fractal dimension of 1.89 for the growing MST to the "trivial" value 2, indicating that the holes near the root gradually shrink and disappear. This is different from the dendritic patterns of diffusion-limited aggregation whose fractal dimension is only 1.71 [148].

The boundaries of the networks, on the other hand, remain complex geometric objects even for large α. This is reflected by the fractal dimension of the external perimeter whose value, although slightly decreasing, remains bigger than 1. Empirical measurements of city borders from satellite images yield fractal dimensions between 1.25 and 1.38 [163] similar to what we find in our model.

While this is a pleasing result, another aspect of the model is quite unrealistic. Some vertices, even ones lying quite close to the root, are joined to the network very

late, because connecting them is costly in terms of the route factor. Arguably, the real world does not work this way: one does not decide to leave parts of a city without sewer service just because there is no convenient straight line for the sewer to take. Instead, connections are presumably made to those vertices that can be connected to the root by a reasonably short path, regardless of whether that path is straight. In the case of trains, for instance, people will use a train service, and thereby justify its construction, if their train journey is short in absolute terms, and are less likely to take a longer journey even if the longer one is along a straight line. As we now show, we can, by incorporating these considerations, produce a different model that still generates highly efficient networks.

4.6 A network growth model with short connections to the root

Let us modify Eq. (4.5) to give preference to short paths regardless of their shape. To do this, we write the weight of a new edge (i, j) as simply

$$w'_{ij} = d_{ij} + \beta l_{j0}. \tag{4.6}$$

This weight function is similar to the one studied by Fabrikant *et al.* [140], Eq. (4.2), with the geometric distance l_{j0} replacing the graph distance h_{j0}. However, unlike Fabraikant *et al.*, we assume here that the vertex positions are specified from the outset, rather than being added to the network one by one. This corresponds to a situation where sites available for settlement are determined from the beginning, which we feel is appropriate for a model of urban networks since city ordinances typically determine potential sites decades before they are finally developed.

Note that, unlike in Eq. (4.5), there is now no explicit term in Eq. (4.6) that guarantees low route factors. Nonetheless, the model self-organizes to a state whose route factor is small. Figure 4.12 shows results from simulations of this second model.

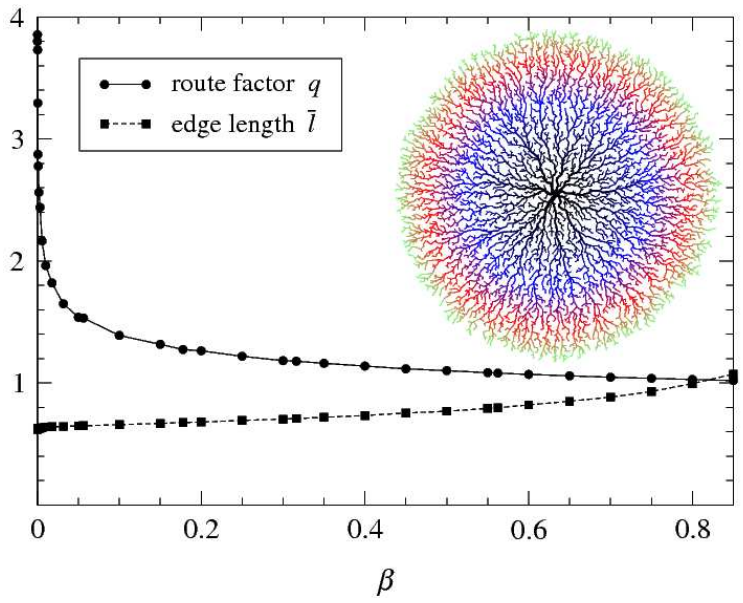

Figure 4.12: Route factor q and average edge length $\bar{l}$ as a function of β for our second model and 10 000 vertices. Inset: an example model network with $\beta = 0.4$.

As the plot shows, the results are qualitatively quite similar to our first model: the high value of q seen for $\beta = 0$ drops off quickly as β is increased, while the mean edge length increases only slowly. Thus we can again choose a value for β that gives behavior comparable with our real-world networks, having simultaneously low route factor and low total cost of building the network. Values of q in the range 1.1 to 1.6 observed in the real-world networks are easily achieved.

When we look at the shape of the network itself, however, we get quite a different impression (see Fig. 4.13). This model produces more symmetric networks that fill space out to some approximately constant radius from the root, not unlike the clusters produced by the Eden model. The second term in Eq. (4.6) makes it economically disadvantageous to build connections to outlying areas before closer areas have been connected. Thus, all vertices within a given distance of the root are served by the network, without gaps, increasing the fractal dimension of the network quickly from 1.89 to 2 already for small β. However, unlike the dendritic shapes of Fig. 4.10, the perimeter now becomes more circular and, hence, in the limit of large β, a one-

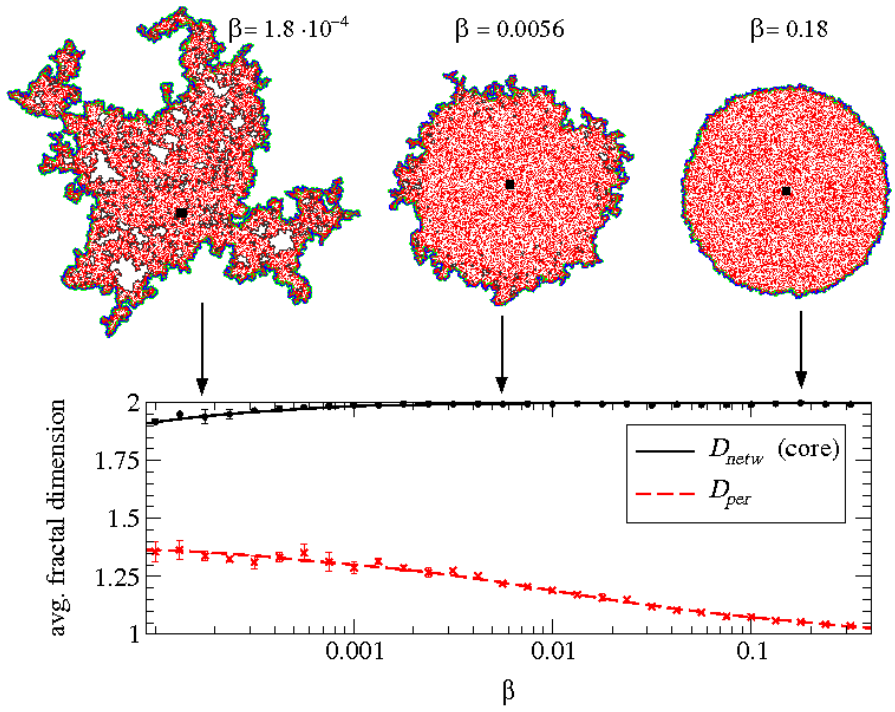

Figure 4.13: Top: The filled Voronoi cells of networks for three different values of β and $50,000$ vertices. Bottom: Fractal dimensions.

dimensional object.

Radial growth in fact may be the secret of how low route factors are achieved in reality. Our second model, unlike our first, does not explicitly aim to optimize the route factor, but it does a creditable job nonetheless, precisely because it fills space radially. The main trunk lines in the network are forced to be approximately straight simply because the space to either side of them has already been filled and there is nowhere else to go but outwards.

How can this be reconciled with the observation that real networks, and the towns they serve, *are* dendritic in form? One might argue that it is primarily a consequence of other factors, such as ribbon development along rivers or highways. In other words, the initial distribution of vertices in real networks is usually non-uniform, unlike our model. It is interesting to see therefore what happens if we apply our model to a realistic scatter of points, and in Fig. 4.9b we have done this for the stations of the

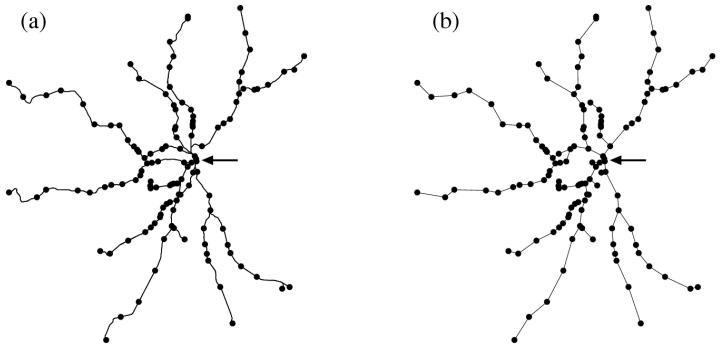

Figure 4.14: (a) Commuter rail network in the Boston area. (b) The model of Eq. (4.6) applied to the same set of stations. The arrow marks the assumed root of the network.

Boston rail system. The figure shows the network generated by our second model with $\beta = 0.4$ acting on the real-world positions of the stations. The result is, with only a couple of exceptions, identical to the true rail network, with a comparable route factor of 1.11 and total edge length 511km. This is a nontrivial result: our first model, for example, does not reproduce the true network nearly so well for any value of α.

4.7 Conclusion

In this chapter, we have studied models of growing spatial distribution networks and compared our results with empirical data. After a review of previous models of networks and fractal growth, we presented a growing version of the minimum spanning tree which is closely related to invasion percolation. It keeps the cost in terms of edge length at a minimum, which appears to be a plausible strategy at first sight. However, in terms of the network distance between vertices, as measured by the so-called route factor, this model is rather poor. Generally, short network length and small route factor are at odds with one another, the first normally being decreased only at the expense of an increase in the second. However, analyzing several real spatial distribution and collection networks (pipelines, sewers and urban railroads),

we found that the real world finds good compromise solutions giving nearly optimal values of both network length and route factor. We have presented two models of growing networks based on greedy optimization strategies that show how this might occur. The first model produces networks of dendritic shape, the second model leads to radial growth. A comparison between fractal dimensions in model and reality gives some credibility to the first model. The second model, however, appears more plausible from first principle; it also is more successful at reproducing actual network structures on real point distributions.

Our focus was primarily on man-made networks, but it is plausible that our arguments are applicable to biological networks as well, such as the circulatory system [164, 165] or fungal mycels [166]. A more careful investigation, however, will be left for future research.

Finally, a word of clarification directed towards readers skeptical of a physicist's tendency to model real networks with a few simple equations as done here. We are well aware that the construction of a real network such as a commuter rail system is based on many complicated decisions that cannot be captured by such a simple approach. Certainly, the Massachusetts Bay Transportation Authority hired dozens of civil engineers before even one mile of railroad track was put in place. We also do not claim that the train network historically evolved exactly as our model would predict, but, nevertheless, the final network is almost the same. Therefore, MBTA could have saved much time and money and just followed our equations; the effect would have been the same. This, indeed, might be of interest to bureaucrats in city halls spending taxpayers' money.

CHAPTER V

OPTIMAL SPATIAL NETWORKS WITH MULTIPLE SOURCES

5.1 Introduction

The previous chapter focused on spatial networks containing one special "root" node which acted as the only sink or source in the network. Utility networks and urban transit systems are often of this type, but many other networks are more complex. Here we will investigate networks with multiple sinks and sources. The efficiency of such networks can be measured in terms similar to those in Chap. IV. On one hand, the smaller the sum of the lengths of all edges, the cheaper the network is to construct and maintain. On the other hand, the shorter the effective distances between vertices in the network, the faster the commodity can travel from point to point. These two objectives generally oppose each other: A network with few and short connections will not provide many direct and short links, while a network with all possible links is usually prohibitively expensive to build. Hence, the problem we are dealing with here is the following: Given n vertices and their positions in two-dimensional space, find the network connecting all vertices that achieves the best compromise between construction cost and convenience of traveling.

Previous literature on the optimum design of geographic networks presents slightly different definitions of what the "best compromise" is. We illustrate our definition in Sec. 5.2 where we also review earlier work on this issue. Regardless of the precise definition, finding the best network is a difficult optimization problem that can be solved

exactly only for very small networks. In practice, one must rely on heuristic methods. We present two possibilities, a simple greedy heuristic in Sec. 5.3.1, and a simulated annealing algorithm in Sec. 5.3.2. Although both methods return similar results if the user cost is proportional to the geometric distance traveled, simulated annealing performs considerably better if the cost is measured in terms of graph distance. (See Fig. 4.3 for the distinction between geometric and graph distance). Networks for these two cases are analyzed in detail in Sec. 5.4 and 5.5. Intermediate cases, where both geometric and graph distance matter, are also possible—the Internet is one important example—and are the subject of Sec. 5.7. In Sec. 5.8 we demonstrate how solutions to the network design problem can be combined with solutions to the p-median problem of Chap. III to create networks optimized for realistic population distributions. Sec. 5.9 concludes this chapter with a brief summary.

5.2 The optimal network design problem

An early attempt at a spatial network design problem can be found in Kansky's 1963 dissertation on transportation networks [85]. Taking the Sicilian railroad as an example, he tries to reconstruct the network from first principles given the positions of the railroad stations. In his solution, towns are ranked by wealth and edges are inserted into the network one by one, preferentially connecting to wealthy towns and preferentially establishing short links. The method, however, includes a number of exceptions based on rather subjective decisions to prevent edges that are "unlikely" or "not meaningful". That the end result bears similarity with the real Sicilian railroad network is hence not very surprising. (Growing networks with preferential attachment have resurfaced in the literature recently, see Sec. 4.2, but not as solutions to any specific optimization problem.)

Scott [12], labeling Kansky's model as "rather diffident," pursues a more rigorous approach. In his model, efficiency depends on two quantities: the total length of all

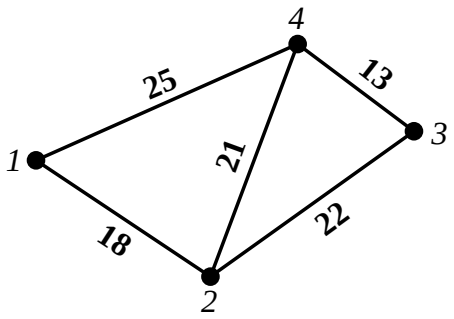

Figure 5.1: A simple illustrative network. Italic numbers refer to vertices, bold numbers to Euclidean edge lengths. Since there is no edge between *1* and *3*, the path of shortest geometric distance between these two vertices is $1 \leftrightarrow 4 \leftrightarrow 3$ which is slightly shorter than $1 \leftrightarrow 2 \leftrightarrow 3$.

edges, T, and the sum of the shortest effective geometric distances in the network between all pairs of vertices, Z. In Fig. 5.1, to take a simple example, the values for T and Z are

$$T = l_{12} + l_{14} + l_{23} + l_{24} + l_{34} = 18 + 25 + 22 + 21 + 13 = 99, \tag{5.1a}$$

$$\begin{aligned} Z &= l_{12} + l_{13} + l_{14} + l_{23} + l_{24} + l_{34} \\ &= 18 + 38 + 25 + 22 + 21 + 13 = 137 \end{aligned} \tag{5.1b}$$

where l_{ij} is the geometric distance between vertices i and j. Introducing the $n \times n$ adjacency matrix A whose elements are $A_{ij} = 1$ if there is an edge between i and j, and $A_{ij} = 0$ otherwise, we can write more generally

$$T = \sum_{i=1}^{n} \sum_{j=i+1}^{n} A_{ij} l_{ij}, \tag{5.2a}$$

$$Z = \sum_{i=1}^{n} \sum_{j=i+1}^{n} l_{ij}, \tag{5.2b}$$

where it is assumed that all distances satisfy the triangle inequality which is the case here as well as throughout this chapter. If there is no path between i and j, we formally set $l_{ij} = \infty$.

In Scott's formulation of the optimal network design problem, it is assumed that the cost of the network is equal to T and that there is a fixed limit on the network

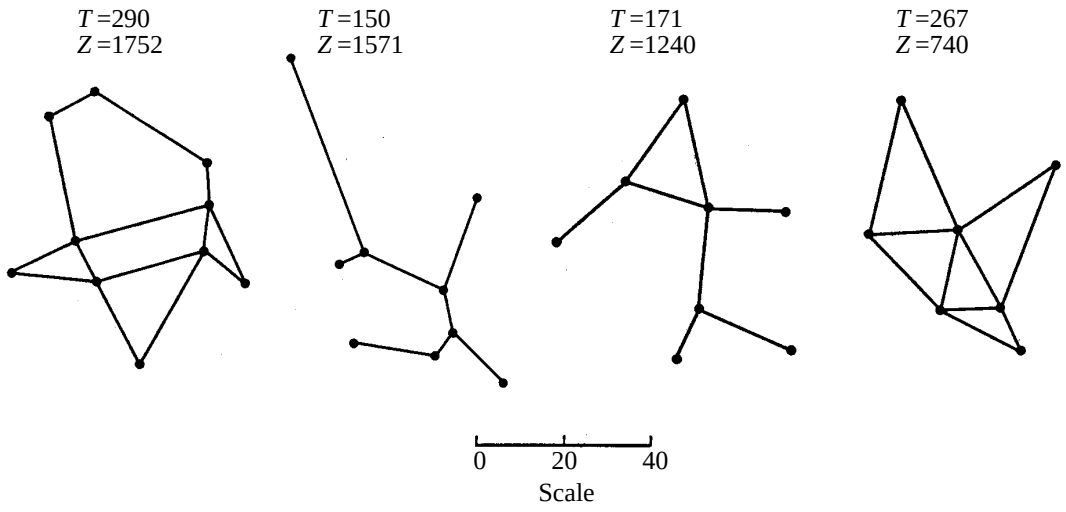

Figure 5.2: Optimal networks according to Scott [12].

cost T_{lim}. We then search for the network structure that can be built within budget, i.e. $T < T_{lim}$, and minimizes the mean vertex–vertex distance between all vertex pairs $\propto Z$. Some small optimum networks constructed this way are shown in Fig. 5.2.

This definition is certainly reasonable, but Billheimer and Gray [129] argue that the cost of a network is not only due to its construction, but also its use. Assuming that there will be p_{ij} passengers traveling between vertices i and j, and that travel costs are proportional to the distances traveled in the network, the total cost C will be the construction cost T plus the travel cost thus

$$C = T + \gamma \sum_{i=1}^{n} \sum_{j=i+1}^{n} l_{ij} p_{ij}. \tag{5.3}$$

where γ is a non-negative parameter determining the relative weight of the second term with respect to the first. If we can for simplicity assume that there are equally many passengers traveling between all pairs i and j, we can, by redefining γ, simplify the last equation to

$$C = T + \gamma Z \tag{5.4}$$

The optimum network from this point of view is the one minimizing this last equation. Unlike before, there is no fixed upper limit on T. Instead we aim at the best

compromise between T and Z which depends on the value of γ. There is no direct equivalence between the two definitions of optimal networks, but increasing T_{lim} in the first definition has a similar effect as increasing γ in the second one. The larger γ, the less weight is given to T in relative terms, and hence the larger T will become. The main advantage of the definition by Billheimer and Gray is that we search the optimum among all networks without any constraints, while in Scott's definition the constraint $T < T_{lim}$ introduces some additional difficulty for numerical solutions.[1]

Several modifications of Eq. (5.3) are possible. The geometric distances l_{ij} in the definition of Z, Eq. (5.2b), could for example be replaced by the graph distance h_{ij} (see Fig. 4.3). We will study the implications of this change in Sec. 5.5. Other authors have considered variable construction costs for different edges and variable user costs per kilometer [129, 168, 167], finite capacities and congestion effects [128], and limiting the possible networks to trees only [169]. However, these are considerations we will not address in this work.

5.3 Generating near-optimal networks

5.3.1 A greedy algorithm

In principle, one way to find the optimum network is to first calculate Eq. (5.4) for all possible networks containing the n given vertices and then determine the minimum among all results. Unfortunately, since the number of networks we would have to investigate is $2^{n(n-1)/2}$ and, hence, grows rapidly as a function of n, going through every single network is impossible even for relatively small n. Some researchers have proposed exact solution algorithms using clever, but quite intricate search strategies that can reduce the number of explicit calculations [170, 171, 172]. However, their

[1] In fact, γ plays the role of a Lagrange multiplier for solving the original constrained problem, see [167].

performance is still not fast enough in practice.

The challenge appears in some sense similar to the p-median problem of Chap. III where there was presumably no exact polynomial-time solution algorithm. Whether the spatial network design problem is, like the p-median problem, NP-hard is not known. Johnson *et al.* [173] proved NP-completeness for non-spatial networks with arbitrary distances. (A problem is *NP-complete* if it belongs to NP and is NP-hard.) Euclidean distances might simplify the problem, but in similar cases where the non-geometric problem is NP-complete (e.g. the traveling salesman problem) the constraints imposed by Euclidean space are too weak to reduce the complexity. Since an efficient exact algorithm is at least far from obvious, it is reasonable to develop a heuristic method for practical work.

One such method proposed by Billheimer and Gray [129], and later improved by Los and Lardinois [168] is a simple greedy optimization strategy where at each step one edge is deleted or added such that the network's cost is reduced by as much as currently possible. More precisely, the algorithm performs the following steps.

(1) Initialization: Begin with a full network, i.e. the network contains all possible $n(n-1)/2$ edges.

(2) Edge elimination phase: Calculate for each edge in turn how much the total cost would change if that edge is eliminated from the current network. Remove the edge that leads to the greatest reduction. Repeat this procedure for the new network. If no improvement can be made, go to (3).

(3) Edge insertion phase: Go through each pair of vertices in turn that are not currently connected, and calculate how much the total cost could be reduced by inserting an edge between them. Add the edge to the network that leads to the greatest decrease and return to (2). If the cost cannot

be improved, the algorithm terminates returning the current network as the best solution.

This is a polynomial-time algorithm whose speed could in principle be improved by some minor modifications (see [168]), but the major concern is whether it actually finds near-optimal solutions. Although it is guaranteed that the solution cannot be improved by one single edge insertion or deletion, the algorithm might get stuck in a local minimum that is still considerably more expensive than the global minimum. In Chap. III we raised the same objection against the steepest-descent method for solving the p-median problem, and, like there, we deal with this concern by developing a different heuristic based on simulated annealing which we will now present.

5.3.2 Simulated annealing

The implementation of a simulated annealing (SA) method for the optimal network design problem is similar to the procedure laid out in Sec. 3.3. We repeatedly generate new network configurations and accept them according to the Metropolis criterion

$$\text{acceptance probability} = \begin{cases} \exp[-\beta(C_{new} - C_{old})] & \text{if } C_{new} > C_{old}, \\ 1 & \text{otherwise,} \end{cases} \tag{5.5}$$

where C_{old} and C_{new} are the network costs before and after the reconfiguration, and β is a positive parameter which increases after every iteration.

We use two different methods to randomly create a new network configuration from the previous one; the first one changes the number of edges, whereas the second one keeps it constant. The first method is the random insertion or deletion of one edge (see Fig. 5.3a). We randomly choose two vertices and, if there is an edge between these two, we remove this edge from the network, otherwise we add an edge between them. The rest of the network remains unchanged. This move set is ergodic, i.e. we can in principle reach every possible network, in particular the optimal one, from any

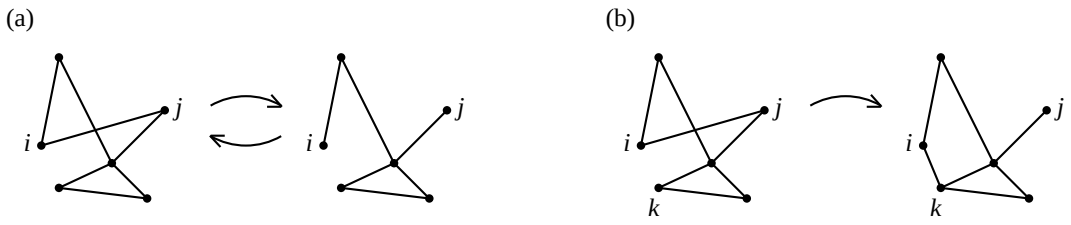

Figure 5.3: The random network reconfigurations used in the simulated annealing Markov chain. (a) Random edge insertion and deletion: Two vertices i and j are randomly chosen. If there is an edge between i and j, this edge is removed. Otherwise the edge $i \leftrightarrow j$ is added. (b) Random rewiring of one edge: The random edge $i \leftrightarrow j$ is deleted and instead the edge $i \leftrightarrow k$ added.

initial condition with a finite number of edge insertions or deletions. However, we found that the time to reach the optimum, is much reduced if we also allow a second type of elementary move (Fig. 5.3b). There we first randomly choose one of the edges, $i \leftrightarrow j$, and one of its endpoints, for example i. Then, one vertex k not adjacent to i is randomly chosen, and the edge $i \leftrightarrow j$ replaced by $i \leftrightarrow k$.

Most time is spent, here as well as in the greedy algorithm, on the calculation of the new network cost C_{new}. The shortest path lengths l_{ij} in the network, needed in Eq. (5.2), are most quickly calculated using Dijkstra's algorithm which needs $O(mn \log n)$ time where m is the number of edges and n the number of vertices in the network [43]. In an efficient implementation, however, one should avoid applying Dijkstra's algorithm over and over again after each reconfiguration. In fact, it only needs to be applied once to the initial network, and, because we only change one or two edges at a time, we can instead use the dynamic graph algorithm by Ramalingam and Reps [174] for all following iterations which avoids the complete recalculation of all distances. The Ramalingam-Reps algorithm is typically much faster than Dijkstra's algorithm, even though it has the same worst-case complexity.

We found that a good initial value of β in Eq. (5.5) is $0.1/C_{MST}$, where C_{MST} is the cost of the minimum spanning tree. After each iteration we increase β by a factor $1 + 3 \cdot 10^{-6}$ to slowly lower the probability of a cost increase. If 10^6 consecutive

iterations do not alter the network, we consider it as sufficiently optimized. We have chosen these parameters since neither lowering the initial value of β nor slowing down its subsequent increase had a measurable effect on the quality of the returned results.

To compare SA with the greedy algorithm, we have calculated a number of examples for randomly generated vertex positions and various values of γ in Eq. (5.4). For most of this chapter we place n vertices randomly in a square of side length $2\sqrt{n}$ and impose periodic boundary conditions, i.e. a line leaving the square at the top enters the square again at the bottom, and similarly a line at the left end reappears on the right. With this particular choice of the side length, the average Euclidean "crow flies" distance between a vertex and its nearest neighbor is equal to one, independently of n, and the periodic boundary conditions reduce finite-size effects.

The results of our comparison are summarized in Tab. 5.1. For $n = 7$, both algorithms are about equally reliable: SA found the global optimum in all cases, the greedy algorithm in all but one. For $n = 50$, the global optimum cannot be explicitly calculated—the number of vertices is simply too big—but we can still say that SA is always at least as good as the greedy algorithm, typically beating it by a few tenths of a percent. This may not seem very impressive, but we will see in Sec. 5.5 that SA outperforms the greedy algorithm by wider margins if the geometric distance l_{ij} in Eq. (5.2b) is replaced by the graph distance h_{ij}. Although we will focus on the geometric distance in the following section, we will nevertheless use the slightly better SA algorithm for numerical calculations.

$n = 7$	$\gamma = 0.4$	$\gamma = 1.0$	$\gamma = 2.0$	$\gamma = 4.0$	$\gamma = 10.0$
greedy alg.finds global minimum	9/10	10/10	10/10	10/10	10/10
SA finds global minimum	10/10	10/10	10/10	10/10	10/10
$n = 50$	$\gamma = 0.002$	$\gamma = 0.006$	$\gamma = 0.02$	$\gamma = 0.06$	$\gamma = 0.2$
SA better than greedy alg.	9	10	10	9	5
SA equal to greedy alg.	1	0	0	1	5
SA worse than greedy alg.	0	0	0	0	0
greedy alg. bigger on avg. (%)	0.9	0.6	0.6	0.2	0.0
Maximum Difference (%)	2.0	2.0	0.9	0.4	0.1

Table 5.1: Comparison of SA and the greedy algorithm for $n = 7$ and $n = 50$ vertices. For each value of gamma, 10 experiments were carried out.

5.4 Varying the cost per kilometer

The shape of an optimally designed network depends on the parameter γ in Eq. (5.4) which measures the weight given to the user's travel cost Z in comparison to the construction cost T. For two limiting cases the solutions are immediately clear. If $\gamma \to 0$, the optimal network is the minimum spanning tree since this is the cheapest network to build. If $\gamma \to \infty$, construction costs play no role, so that the optimum is a full network with all possible $n(n-1)/2$ edges. Between these limits, the optimum is less obvious, because, as we saw in Chap. IV, the MST is a poor solution as far as the traveled distance Z is concerned, and the full network is poor in terms of network length T.

To investigate what compromise between these extremes is best, we have constructed near-optimal networks for various values of γ with $n = 200$ vertices. A series of networks generated in this fashion is shown in Fig. 5.4. For $\gamma = 0.0002$, the optimum network is almost identical to the MST. As γ increases, the number of edges goes up, but for $\gamma = 0.002$ and 0.02, vertices still predominantly have links to vertices in the immediate neighborhood. Only for $\gamma = 0.2$ we find edges spanning longer distances.

We measured the efficiency of the networks in terms of T and Z as functions

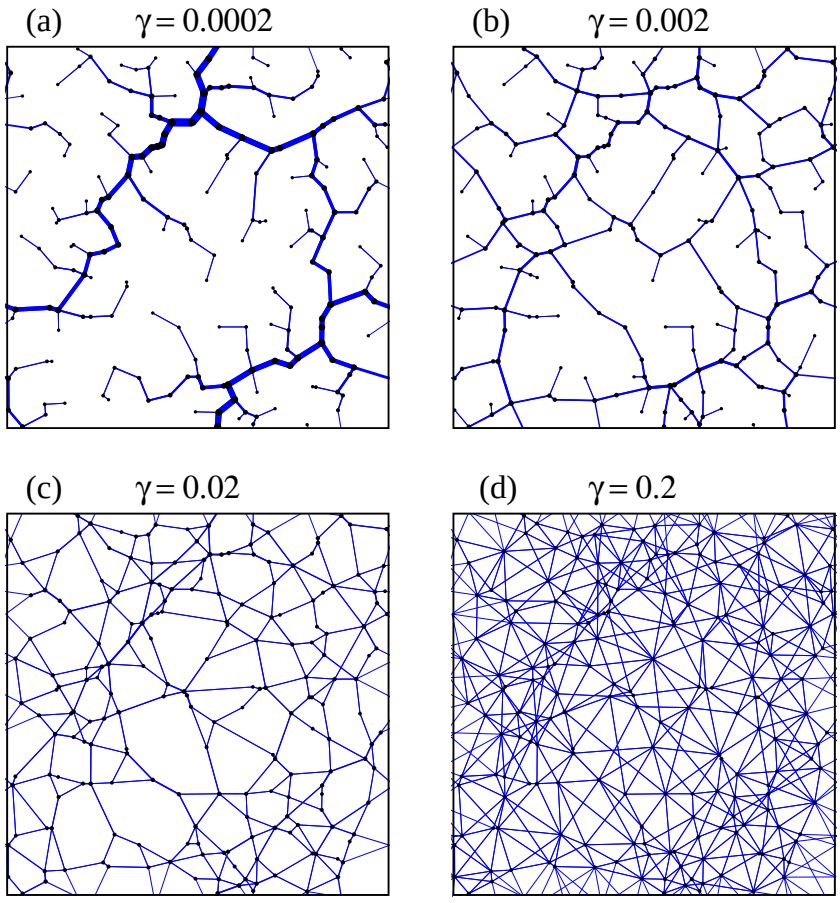

Figure 5.4: Near-optimal networks in terms of the total cost C with $n = 200$ vertices for different values of γ.

of γ. The result, plotted in Fig. 5.5, shows that T is a monotonically increasing and Z a monotonically decreasing function. Roughly speaking, two regimes can be distinguished. For small γ, T stays almost constant near its minimum value, whereas Z decreases rapidly. For large γ, the roles are reversed: Z does no longer decrease much, but T grows quickly.

The network length T is the product of two factors: the number of edges m and the average edge length $\bar{l}$. It is instructive to analyze how these two factors contribute separately to the increase in T in Fig. 5.5. In the upper panel of Fig. 5.6 we plot the average degree $\bar{k} = 2m/n \propto m$ as well as the maximum degree k_{max} against γ. Both increase only very slowly for small γ, but more quickly for larger γ. The lower

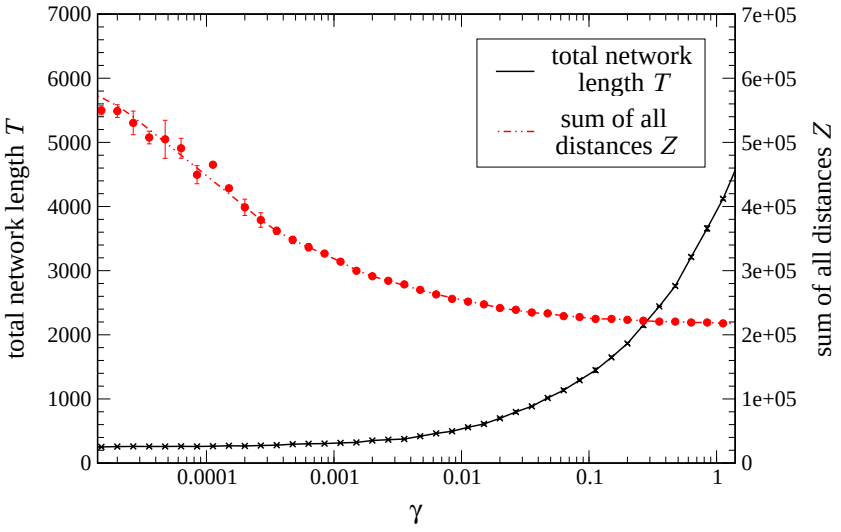

Figure 5.5: Construction cost T and user cost Z as functions of γ for $n = 200$ vertices.

panel shows the average and maximum edge length $\bar{l}$ and l_{max}. Here the growth in both is also more rapid for large γ, but l_{max} begins to grow noticeably sooner. Hence, the increase in T is caused both by an increase in the number of edges and their length. The average degree increases more strongly than the average length, but the maximum length grows more quickly than the maximum degree.

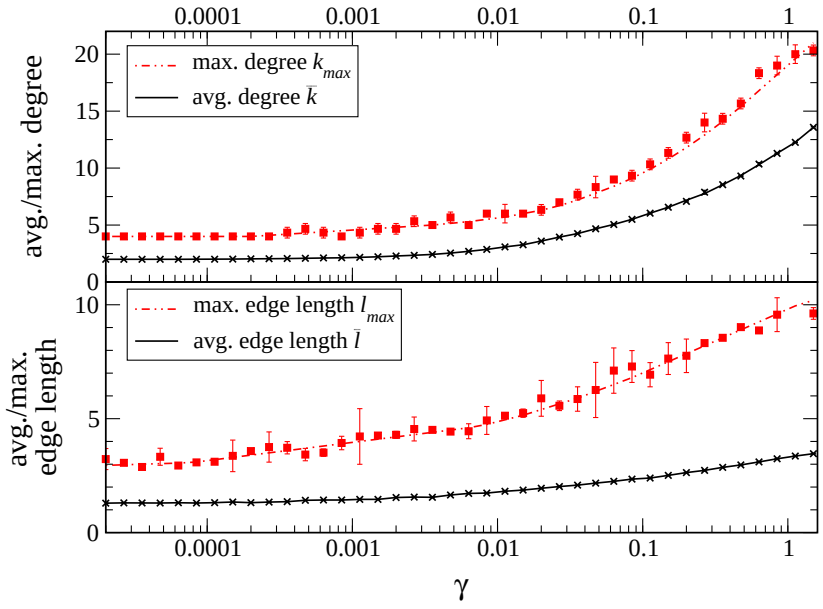

Figure 5.6: Contributions to the construction cost T in Fig. 5.5. Upper panel: Maximum degree k_{max} and average degree $\bar{k}$. Lower panel: Maximum edge length l_{max} and average edge length $\bar{l}$.

5.5 Networks with fixed cost per traversed edge

So far, we have assumed that user costs are proportional to geometric distances between vertices which is a plausible starting point. In a road network, for example, the quickest and cheapest route is usually not very different from the shortest route measured in kilometers. However, in some networks, users might have different preferences. In an airline network, for example, passengers often spend much time waiting for connecting flights, so that they care more about the number of stopovers than about the kilometers traveled. The appropriate modification of our model for a situation, where the delays occur at the vertices rather than between them, is to replace the geometric distance l_{ij} in Eq. (5.2b) by the number of legs in a journey, which is the graph distance h_{ij}, thus

$$Z' = \sum_{i=1}^{n} \sum_{j=i+1}^{n} h_{ij}. \tag{5.6}$$

$n = 7$		$\gamma = 0.1$	$\gamma = 0.2$	$\gamma = 0.5$	$\gamma = 1.1$	$\gamma = 2.1$
greedy alg.	finds global minimum	6/10	3/10	3/10	1/10	6/10
	Avg. Difference (%)	0.6	2.1	2.4	1.8	0.1
	Max. Difference (%)	2.1	8.9	7.4	4.0	0.5
SA	finds global minimum	9/10	9/10	8/10	7/10	8/10
	Avg. Difference (%)	0.1	0.1	0.1	0.3	0.1
	Max. Difference (%)	0.6	1.1	0.3	2.1	0.1

$n = 50$	$\gamma = 0.0008$	$\gamma = 0.003$	$\gamma = 0.008$	$\gamma = 0.03$	$\gamma = 0.08$
SA better than greedy alg.	7	9	10	10	10
SA equal to greedy alg.	0	0	0	0	0
SA worse than greedy alg.	3	1	0	0	0
greedy alg. bigger on avg. (%)	0.2	1.6	6.7	9.3	5.0
Maximum Difference (%)	0.7	4.3	9.1	13.9	6.9

Table 5.2: Comparison of SA and the greedy algorithm for $n = 7$ and $n = 50$ vertices where Z has been replaced by Z' of Eq. (5.6). For each value of γ, 10 experiments were carried out.

The total cost C' is then, as before, the sum of construction and travel cost, hence

$$C' = T + \gamma Z', \;\; \gamma > 0, \tag{5.7}$$

and the optimum network is the one minimizing C'.

Unfortunately, replacing the geometric distance by the graph distance does not seem to reduce the complexity of the optimization problem, but both the greedy and simulated annealing heuristics presented in Sec. 5.3 can be easily modified. We have tested both modified algorithms and have summarized our findings in Tab. 5.2. Both algorithms are less successful for $n = 7$ than their counterparts in Tab. 5.2, but SA still finds the global optimum in the majority of all cases unlike the greedy algorithm, and even where it does not find the optimum, its results are still better, both on average and in the worst case. For larger networks with $n = 50$, exact solutions cannot be computed on a human time scale, but comparing the results of both heuristics, SA outperforms the greedy algorithm by sometimes more than 10%. We will therefore rely on SA in the following calculations, although improvements are certainly still possible.

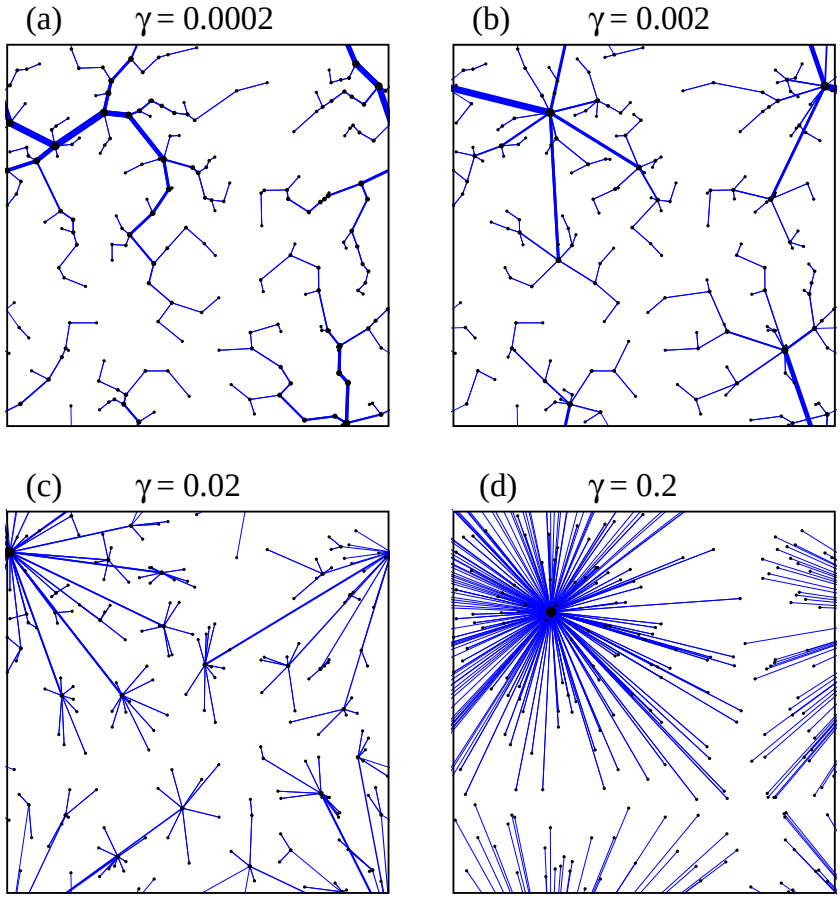

Figure 5.7: Near-optimal networks in terms of C' with $n = 200$ vertices for different values of γ.

In Fig. 5.7 we show a series of near-optimal networks obtained with SA for four different values of γ. For $\gamma = 0.0002$, the network is similar to the MST. Even for larger values, the networks remain trees, so the number of edges, unlike in Fig. 5.4, hardly increases. Instead, a small number of highly connected vertices appears which, like hubs in an airline network, collect most of the traffic from vertices in the vicinity. For $\gamma = 0.2$ the network finally turns into a "star graph" where one vertex connects to all other vertices.

The measures of efficiency for this model, T and Z', are plotted against γ in Fig. 5.8. Z' is a well-behaved monotonically decreasing function, similar to Z in Fig. 5.5, and asymptotically approaches the limiting value $n(n-1)$. T, by contrast,

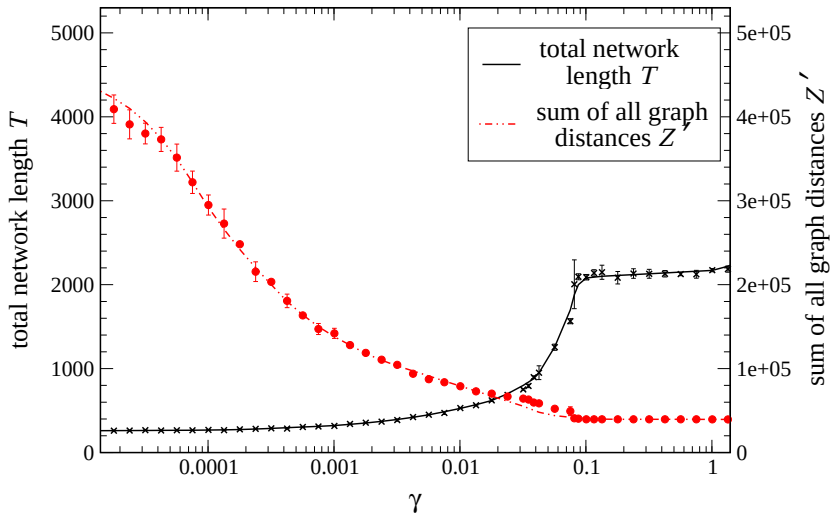

Figure 5.8: Construction cost T and user cost Z' as functions of γ for $n = 200$ vertices

is a more complicated function. It increases monotonically, but the slope changes suddenly from a high value to almost zero near $\gamma \approx 0.08$.

What causes this sharp bend? We can gain more insight by splitting T, as we did before, into two factors: the number of edges m and the average edge length $\bar{l}$. There is nothing directly suspicious about m as a function of γ (see upper panel of Fig. 5.9), but the maximum degree k_{max} shows a peculiar behavior near $\gamma \approx 0.08$. At this point k_{max} reaches the highest possible degree in a network with n vertices, $n - 1$. Since m, on the other hand, is still at its minimum value $n - 1$, we can conclude that the sudden change in the slope of T is related to the emergence of star graphs as optimal networks. (Some authors have referred to similar network phenomena as "gelation" in analogy with condensed matter physics [175, 176].)

Since m is well-behaved throughout, the bend in T must be caused by $\bar{l}$ as the lower panel of Fig. 5.9 confirms: $\bar{l}$ grows from approximately 1 for $\gamma \to 0$ to $\frac{1}{3}(\sqrt{2} + \log(\sqrt{2} + 1))\sqrt{n} \approx 0.765\sqrt{n}$, the average edge length in a star graph, at $\gamma \approx 0.08$ with an increasingly steep slope. At this point the slope suddenly drops to zero, and $\bar{l}$ even starts decreasing. This happens because every edge that is added to a star

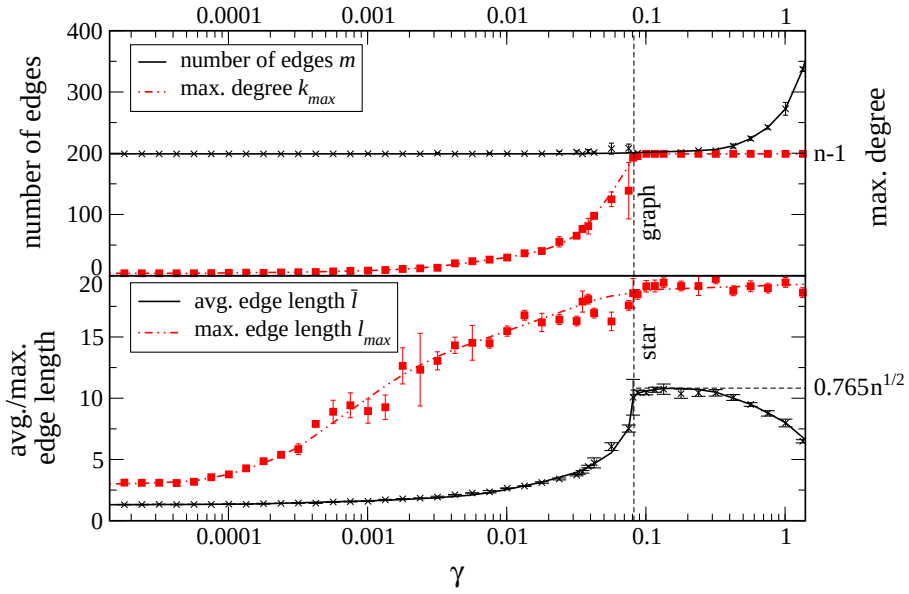

Figure 5.9: Contributions to the construction cost T in Fig. 5.8. Upper panel: Maximum degree k_{max} and average degree $\bar{k}$. Lower panel: Maximum edge length l_{max} and average edge length $\bar{l}$.

graph has the same effect on Z'—it only reduces the graph distance between the two endpoints, and it only reduces it by one—so that there is no point in inserting any long edges. If we add more edges at all, they would have to be short, thus decreasing the average edge length $\bar{l}$. This also explains why the maximum edge length l_{max}, which has grown steadily below $\gamma \approx 0.08$, stays almost constant above the transition.

5.6 The traffic in optimal networks

A quantity of practical interest in networks is the amount of traffic along the edges. The definitions of Z and Z', Eq. (5.2b) and (5.6), imply three assumptions. First, there is an equal demand for traveling between all origin-destination pairs. Second, all edges have infinite capacities, so that there are no delays due to congestion. Third, all the traffic is along the shortest paths in the network, either measured by geometric or graph distance; in other words, users do not take intentional detours. The situation in real networks is, of course, more complicated, but these assumptions keep the already

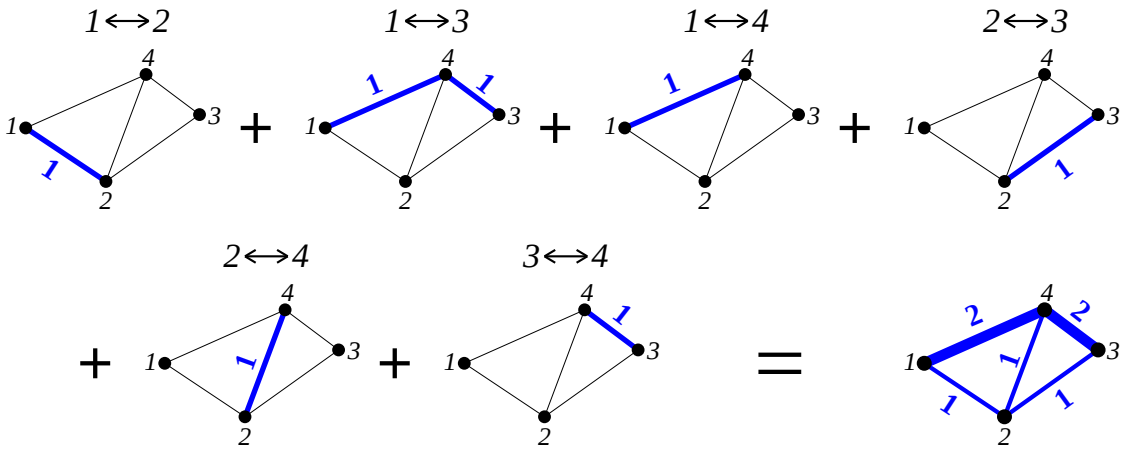

Figure 5.10: Edge betweenness for the simple network of Fig. 5.1. For every pair of vertices, we send one unit of flow along the shortest geometric path between them. The amount of flow is indicated by bold numbers on the edges. The distances are given in Fig. 5.1. Note that the shortest geometric path from *1* to *3* is via *4*. Adding the flow along each edge yields the "edge betweenness". In this example, the edges $1 \leftrightarrow 4$ and $3 \leftrightarrow 4$ have a betweenness of 2, all other edges a betweenness of 1. We could have also used graph distance instead of geometric distance, which amounts to setting all distances equal to one, but this generally gives different results. In terms of graph distance, there are, for example, two shortest paths between vertices *1* and *3*, namely via *2* and via *4*. Both paths would contribute one half unit of flow to the edge betweenness.

difficult network optimization problem more tractable.[2]

An appropriate way to measure traffic flow under these assumptions is a generalization of the "edge betweenness" [178]: We send one unit of flow between every possible origin-destination pair along the shortest path and count the number of units that have passed through one particular edge. Equivalently, the edge betweenness is the number of shortest paths in the network running along that edge. A sample calculation is shown in Fig. 5.10. In the networks of Fig. 5.4 and 5.7, we have indicated a higher edge betweenness with a larger line width.

For the models of Sec. 5.4 and 5.5, we have measured the betweennesses of all edges for several values of γ. In Fig. 5.11, we plot the cumulative distributions, i.e.

[2]Traffic assignments in networks with limited capacities can be counter-intuitive. Braess' paradox, for example, implies that under certain conditions the flow in the network is improved by removing(!) edges [177].

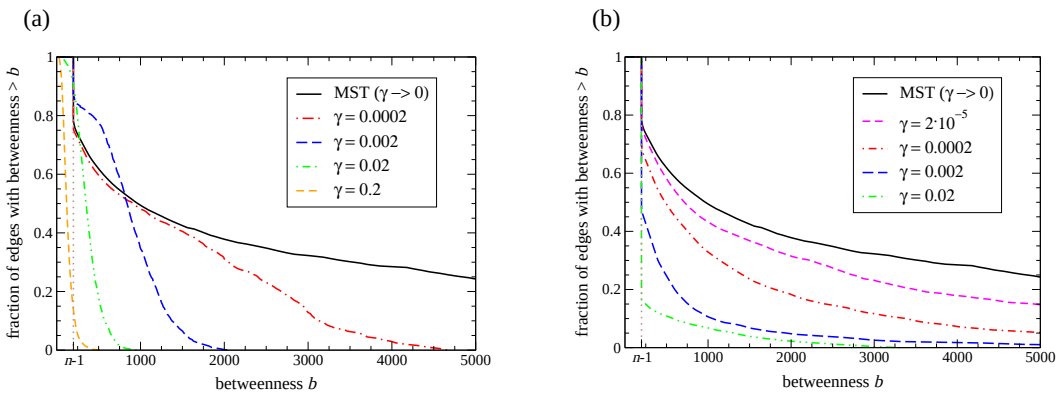

Figure 5.11: Cumulative edge betweenness distributions. (a) Distributions for networks minimizing the total cost C of Eq. (5.4), where the user costs depend on geometric distances. (b) Distributions for networks minimizing the total cost C' of Eq. (5.7), where the user costs depend on graph distances.

the fraction of the edges in the network whose betweenness is larger than a certain value b. Fig. 5.11a shows the result for the first model where the user cost Z depends on the geometric distances; Fig. 5.11b is for the second model whose user cost Z' depends on graph distances. For $\gamma \to 0$, where the optimal networks are MSTs, both models possess a long-stretched tail indicating that in this case some edges, like main arteries, have to support a large portion of the flow in the network. If such an edge fails, for example because of construction or because it cannot handle that much traffic, many routes in the network will be affected.

The distributions for both models become narrower as γ increases. The effect, however, is much stronger in the first than in the second model. For $\gamma = 0.02$, for example, no edge in the first model has a betweenness larger than 1200, whereas in the second model the maximum is around 3300. The difference is closely related to the different network structures. The second model, unlike the first, possesses a small number of highly connected hubs (see Fig. 5.7). These hubs collect most of the traffic and, since the networks are trees, the traffic must inevitably pass through the few edges between the hubs which explains their high betweenness. Networks generated by the first model, on the other hand, have no hubs but more edges (see Fig. 5.4) so

that the maximum betweenness is smaller.

Towards the left-hand side most curves in Fig. 5.11 have jumps at $b = n-1$. These jumps are present if a large fraction of the vertices have a degree of one because these vertices can only be reached along one edge, so that traffic from all other $n-1$ vertices must go through that edge. Since the second model leads to increasingly many such "dead ends" as γ grows, the jumps in Fig. 5.11b become bigger. However, a smaller betweenness than $n-1$ is possible as the orange curve in Fig. 5.11a proves: For $\gamma \to \infty$ the optimal networks contain all $n(n-1)/2$ possible edges, hence all betweennesses in this limit are equal to one.

5.7 Balancing geometric and graph distance

Generally, a network will be the better the shorter the paths between points are, but the way we measure path length can vary. In Sec. 5.4, we used geometric, in Sec. 5.5 graph distance. However, in some networks, a compromise between these two is desired. One example is the Internet, by which we mean the physical network of computers and routers connected mostly by optical fiber. The total time required for an Internet packet to reach its destination depends on two limiting factors: the propagation delay, i.e. the time needed to traverse the physical distance between vertices (computers and routers) at the speed of light in optical fiber, and store and forward delays, i.e. the time it takes the vertices along the path to receive, process, and resend the packets. The propagation delay is proportional to the geometric path length; store and forward delays, on the other hand, depend on the number of routers along the way. Empirically, both types of delays are more or less equal.[3]

[3]This can be easily verified with the UNIX ping utility. Pinging, for example, several Internet addresses in San Diego, CA, from Ann Arbor, MI, we found round trip times of around 80 ms. The propagation delay for the 6320 km long distance is easily calculated from the speed of light in optical fiber ($\approx 2 \cdot 10^8$ m/s) to be 32 ms (plus some additional delay due to the route factor), so it accounts for around one third of the total delay.

To account for such intermediate situations, we assign to each edge an effective length

$$\tilde{l}_{ij} = (1-\delta)l_{ij} + \delta \tag{5.8}$$

with $0 \leq \delta \leq 1$. The parameter δ determines the user's preference for measuring distance in terms of kilometers or legs. Now we define the effective distance between two (not necessarily adjacent) vertices to be the sum of the effective lengths of all edges along a path between them, minimized over all paths. The total user cost is then proportional to the sum of all effective path lengths

$$Z_\delta = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \tilde{l}_{ij} \tag{5.9}$$

and the optimum network for given γ and δ is the one that minimizes the total cost

$$C_\delta = T + \gamma Z_\delta. \tag{5.10}$$

Obviously, $Z_0 = Z$ and $Z_1 = Z'$, so that our previous results are limiting cases of this more general problem.

In Fig. 5.12 we show networks obtained by varying δ and keeping $\gamma = 0.06$ fixed. The structures of Fig. 5.12(a) and (d) are familiar from before (see Fig. 5.7 and 5.4)—structures with many loops but neither long edges nor hubs in the first case and tree-like structures with long edges and hubs in the second case. For intermediate values of δ, the networks find compromises between hub formation and local links.

We use three quantities to characterize the transition between the two limiting cases: the maximum degree, the average geometric distance along the shortest path, and the average number of edges along that path.[4] The results are plotted against

[4]Note that the shortest path between two vertices is the one whose *effective* length, given by Eq. 5.8, is minimal. Since that path generally neither minimizes the geometric nor the graph distance, the average of the geometric distances (or number of edges) along the shortest paths is *not* equal to the average of the shortest geometric (or graph) distance.

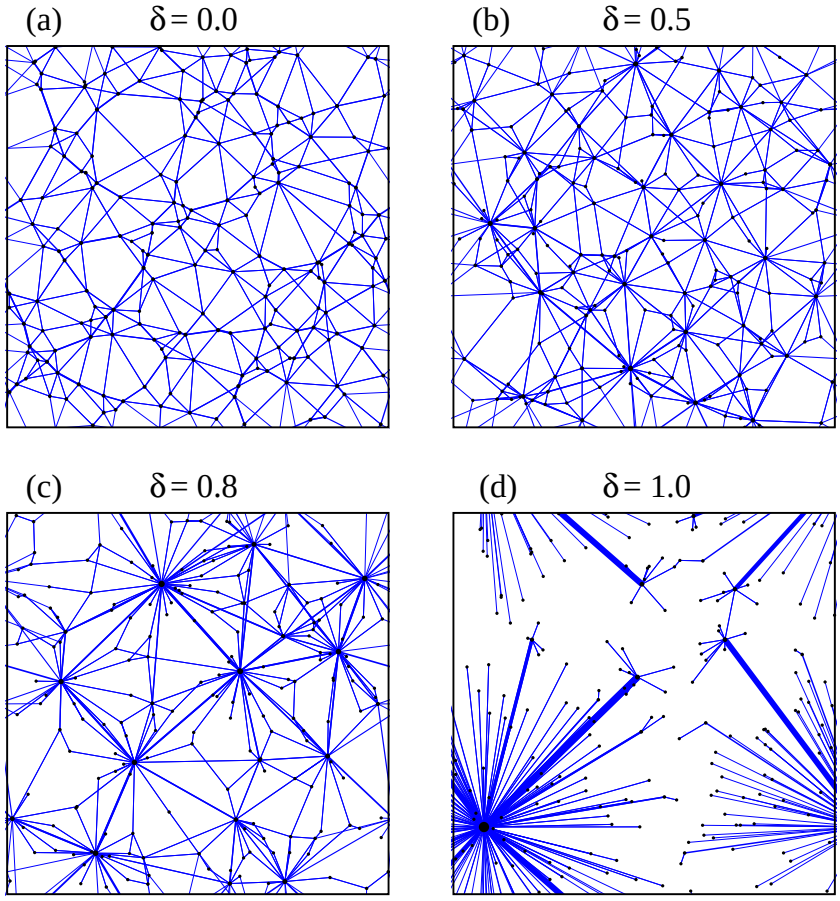

Figure 5.12: Near-optimal networks in terms of C_δ with $n = 200$ vertices. γ is kept constant at 0.06.

δ in Fig. 5.13. The maximum degree and average geometric distance increase only slowly near $\delta = 0$, but grow rapidly near $\delta = 1$. The number of edges along the shortest path, on the other hand, decreases steadily. Reducing the number of legs in a journey by accepting a higher mileage is hence not a linear trade-off. Optimizing the graph distance implies disproportionately long journeys in terms of geometric distance, since even paths between nearby vertices must take the (geometrically) long detour via at least one hub.

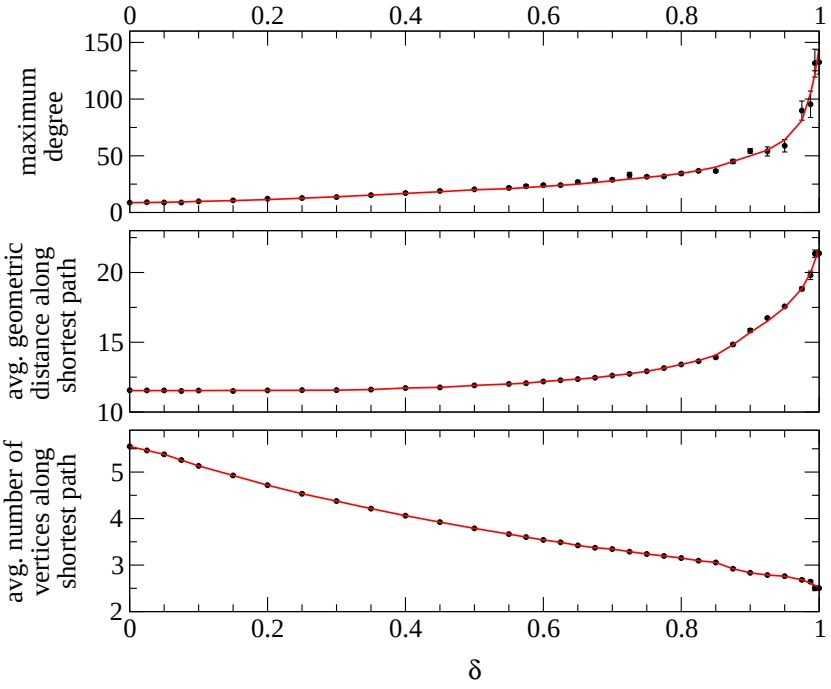

Figure 5.13: Network properties for $n = 200$ vertices and $\gamma = 0.06$ as functions of δ. Top: Maximum degree. Middle: Average geometric distance along the shortest path. Bottom: Average number of vertices along the shortest path.

5.8 Networks with optimally located facilities

Until now, all of our calculations were based on spatially random vertex distributions, but in many practical applications placing vertices randomly in space is a poor choice. Often the vertices are meant to serve the public at large like airports or train stations. In that case a more reasonable choice is to place the vertices such that the mean distance to the nearest vertex for members of the population is minimal. Such a vertex distribution is a solution to the p-median problem for which we have developed a heuristic solution method in Chap. III. It is therefore interesting to apply the network optimization algorithm of Sec. 5.3.2 to a thus optimized spatial distribution of facilities rather than the random distributions we have used so far.

Using the conterminous United States as a concrete example, we first determine a near-optimal placement of $n = 200$ facilities based on the population density in the

year 2000. We then wish to find networks connecting these facilities and optimizing

$$T + \gamma \sum_{i=1}^{n} \sum_{j=i+1}^{n} [(1-\delta) l_{ij} + \delta] p_{ij}. \tag{5.11}$$

Here we have made two changes compared to Eq. 5.4. First, we returned to the more general expression of Eq. 5.3, where we did not assume that the amount of traffic p_{ij} between the vertices i and j is the same for all pairs of vertices. Second, we replaced the geometric length l_{ij} by $\tilde{l}_{ij}$ of Eq. (5.8). The second change allows us to take different user preferences for measuring distance into account. The first change can be motivated as follows.

Since the population density ρ is highly non-uniform, and more densely populated regions generate more traffic, we should not assume that the demand for traveling is the same between all vertices. Vertex i serves the population in the surrounding Voronoi cell V_i, and therefore, the fraction of the population using that vertex is

$$P_i = \frac{\int_{V_i} \rho(\mathbf{r}) \, \mathrm{d}^2 r}{\sum_{k=1}^{n} \int_{V_k} \rho(\mathbf{r}) \, \mathrm{d}^2 r}. \tag{5.12}$$

Clearly, the demand for traffic between i and j should grow with P_i and P_j. The simplest functional relation between traffic and population that satisfies this condition is $p_{ij} = P_i P_j$. Other functional forms are of course possible and worth investigating, but here we have for simplicity exclusively used this assumption.

Optimized networks for four different values of δ are shown in Fig. 5.14. We have, as we did earlier, normalized the length scale such that the average nearest-neighbor "crow flies" distance between the vertices is equal to one. Using this length scale, we have set $\gamma = 800$ for this calculation. For $\delta = 0.0$, where the user only cares about the kilometers traveled, the network consists of mostly local connections between nearby vertices. The density of the vertices is higher and the edge lengths are shorter in regions of higher population density. Consequently, the network is particularly dense in the northeast, but sparse in the western half of the United States.

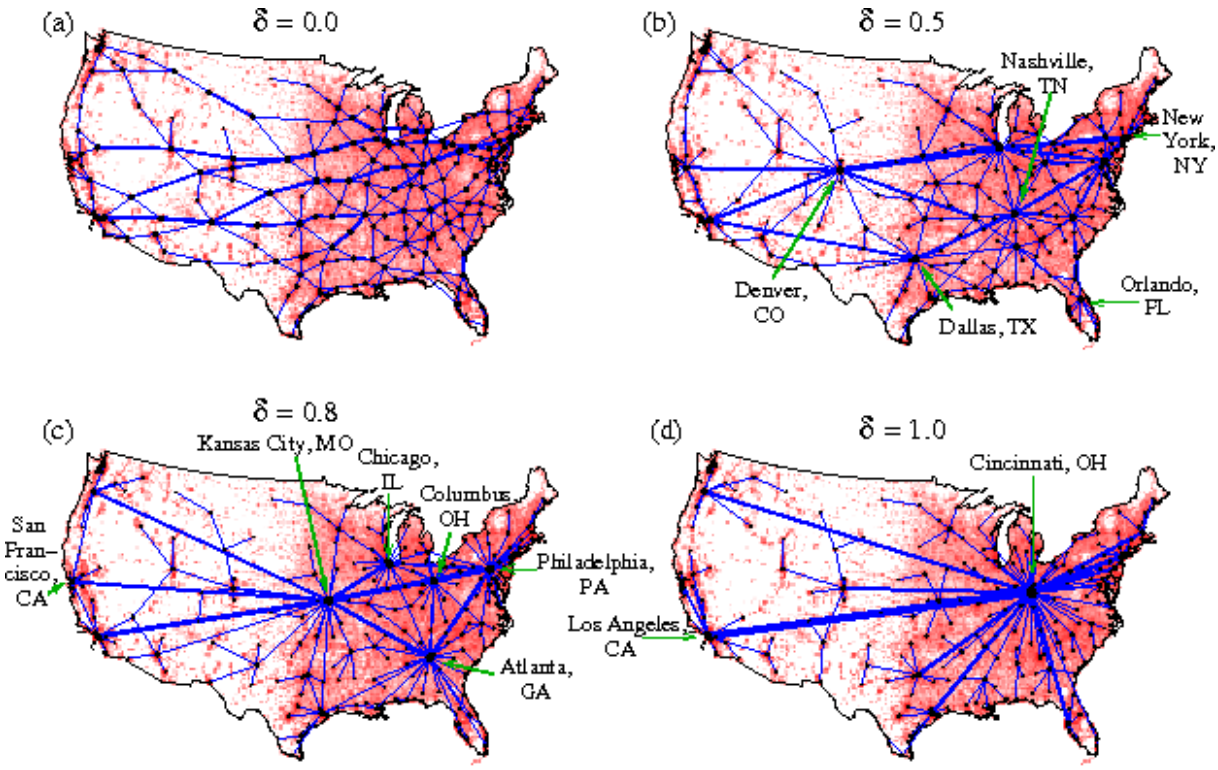

Figure 5.14: Optimal networks for the population distribution of the United States with $n = 200$ vertices for different values of δ.

By increasing δ, the number of legs starts playing a more important role, and the networks increasingly tend to form hubs. At $\delta = 0.5$, a compromise between short edges and several smaller hubs is reached, while at $\delta = 0.8$ hubs in Philadelphia, Columbus, Chicago, Kansas City, and Atlanta handle the bulk of the traffic. On the highly populated Californian coast, two smaller hubs near San Francisco and Los Angeles are visible.

In the extreme case $\delta = 1.0$, where the user does not care about mileage at all, but only about the number of legs, the hubs in the eastern half of the country condense to one central hub near Cincinnati. Similarly, California is now served by only one hub in Los Angeles. Since the edge between Los Angeles and Cincinnati is the only available route from California to the rest of the country, it carries by far the greatest amount of traffic in the network.

Combining solutions to the p-median problem with solutions to the network design problem in the manner presented here should be useful in many situations of practical interest such as the design of transportation networks[5], parcel delivery services, and the Internet backbone.

5.9 Conclusion

This chapter has treated a problem of optimal spatial network design if multiple sources are responsible for the flow in the network. The problem consists of finding the minimum of the sum of the construction cost and the cost to the user. We have assumed that the construction cost is equal to the total length of all edges in the network and that the user cost depends on the distances traveled between the vertices along the edges. Different kinds of distances might be relevant for the user: Geometric distance (in kilometers), graph distance (number of legs), or a combination of both.

Regardless what distance is considered, determining the optimal network is a difficult optimization problem for which only heuristic methods are available. We have compared a simple greedy algorithm with a simulated annealing method. For geometric distances, both algorithms give comparable results, but for graph distances, simulated annealing performs considerably better.

We have studied how the network structure changes if the user cost receives increasingly more weight than the construction cost. If shortest paths are measured in terms of geometric distance, the structure undergoes a smooth transition from minimum spanning tree to fully connected network by adding more and preferentially

[5]It is certainly no coincidence that many of the hubs in Fig. 5.14 coincide with hubs operated by major national airlines. In 2005, Atlanta and Cincinatti are Delta Airlines' main hubs. Chicago, Denver, Los Angeles, and San Francisco are hubs for United Airlines, whereas American Airlines uses hubs in Chicago, Dallas, and Los Angeles.

short edges. If graph distance is used, the network first becomes a star graph, i.e. all vertices connect to one central hub, and only then more edges are added. For mixed distances, intermediate solutions between short edges and hub formation are obtained. We ended with an application of the network design problem to the real geography of the United States by placing vertices such that the mean distance to the nearest vertex for members of the population is minimal. This method to calculate optimal geometric networks for actual population distributions is of potential use for real design problems.

CHAPTER VI

CONCLUSION

In this dissertation, we have studied several applications of statistical physics to geographic problems.

In Chap. II, we have presented an algorithm to create cartograms, i.e. maps in which the sizes of geographic regions appear in proportion to their population or some other analogous property. Such maps are invaluable for the representation of census results, election returns, disease incidence, and many other kinds of human data. Unfortunately, in order to scale regions and still have them fit together, one is normally forced to distort the regions' shapes, potentially resulting in maps that are difficult to read. Many methods for making cartograms have been proposed, some of them extremely complex, but all suffer either from this lack of readability or from other pathologies, like overlapping regions or strong dependence on the choice of coordinate axes. Here we have presented a new technique based on the linear diffusion equation that suffers none of these drawbacks. Our method is conceptually simple and produces useful, elegant, and easily readable maps. We have illustrated the method with applications to lung cancer cases in the State of New York, homicides in California, Autonomous Systems in the Internet, the results of the 2004 US presidential election, energy consumption and production, and the geographical distribution of stories appearing in the news.

In Chap. III, we have studied the p-median problem for a non-uniform population density which consists of finding the optimal positions of p service facilities in

geographic space such that the mean distance between a member of the population and the nearest facility is minimal. We have introduced a heuristic based on simulated annealing which improves the results of a simple steepest-descent optimization. Analytic calculations indicate that, if facilities are distributed optimally for the given population distribution, their density increases with population density, but does so slower than linearly, as the two-thirds power. Numerical calculations based on the density-equalizing projection of Chap. II confirm this result. Although real facilities do not appear to strictly follow a two-thirds power law, the p-median problem provides a starting point for many actual facility location problems.

While Chap. II and III have dealt with distributions of isolated points in geographic space, Chap. IV and V have focused on networks where the points are connected by lines. In Chap. IV, we have analyzed spatial distribution networks growing from an initial "root" node which is the only source or sink of the commodity distributed in the network. The efficiency depends on two properties. First, the network should have short paths from each point to the root, ideally not much longer than the "crow flies" distance between the same two points. Second, the sum of the length of all connections in the network should be low so that the network is economical to build and maintain. These two criteria cannot be optimized simultaneously, the first being improved only at the expense of the second and vice versa, but real networks nevertheless are nearly optimal in both respects. We have suggested two models of growing networks to explain how this situation might occur.

Chap. V looks at the problem of finding optimal networks if there are multiple sinks and sources. Although it is difficult to find globally optimal solutions, we can derive nearly optimal networks using simulated annealing. We have characterized networks optimized for different user preferences and distance metrics in terms of their efficiency and traffic distribution. If we know the population density to be served by the network, the p-median problem can be solved for the point locations

and an optimal network can be derived for these points. Results for the population of the United States have been presented.

We believe that the problems studied here are of practical relevance, and we hope that this dissertation contributes to a better understanding of the laws governing the distribution of points and networks in geographic space.

APPENDICES

APPENDIX A

HOPKINS STATISTIC - A METHOD TO TEST SPATIAL POINT PATTERNS FOR RANDOMNESS

In physics, geography, and biology, data is often in the form of points that are, more or less, randomly distributed within some region of two- or three-dimensional space. Examples include occurrences of a disease, locations of trees in a forest, the arrangement of cell nuclei in tissue, or the positions of galaxies in the universe. Three examples of spatial point patterns, reproduced from [13], are shown in Fig. A.1. Panel (a) shows the locations of 65 Japanese black pine saplings in a square of side 5.7 m [179], panel (b) the locations of 62 redwood seedlings in a square of side 23 m [180, 181], and panel (c) the pattern formed by the centers of mass of 42 cells from insect tissue [182, 181].

The mechanisms behind these three patterns are clearly distinct. In Fig. A.1(a) the points appear to be randomly scattered. Every sapling seems equally likely to grow everywhere in the square. In particular, there is no obvious dependence of the position of one sapling on the positions of the other saplings. Fig. A.1(b), on the other hand, shows strong "clustering" or "aggregation" in the sense that it is more likely to find a redwood seedling in the immediate neighborhood of another one. A possible explanation for this clustering is that the seedlings preferentially grow around redwood stumps whose positions are not recorded, but might be inferred from the data. The pattern in Fig. A.1(c), by contrast, shows no such clusters. The points

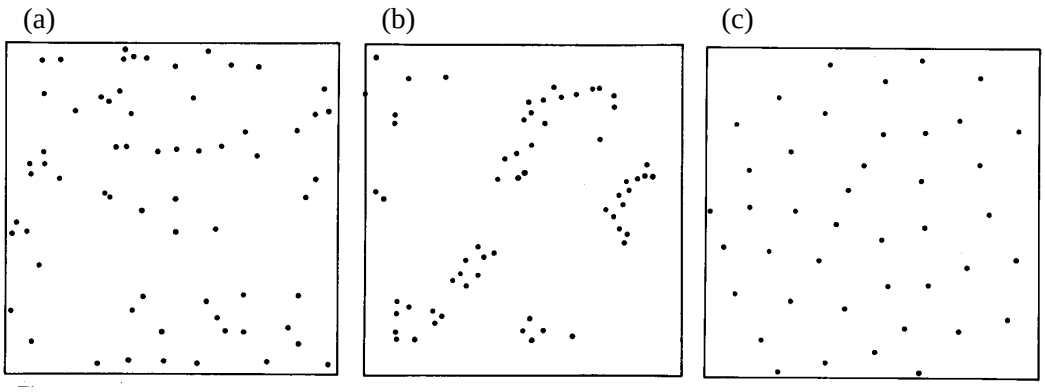

Figure A.1: Three different point patterns. (a) Locations of 65 Japanese black pine saplings in a square of side 5.7 m. (b) Locations of 62 redwood seedlings in a square of side 23 m. (c) Locations of 42 insect cell centers. Reproduced from [13].

appear more evenly distributed than in (a) and (b), with all cell centers relatively far away from their nearest neighbors. Such a "regular" distribution is likely to be the result of repulsion or competition between different points.

In Fig. A.1(b) and (c) the deviations from pure randomness are clearly visible to the eye. In many cases, however, the difference between a random, aggregated, or regular distribution will be more subtle, so we would wish to develop a simple method to characterize the type of distribution and, if possible, state in some way how confident we are with our classification. Several such methods exist, and some are quite sophisticated, see [183, 13]. However, here we describe a simple and elementary one due to Hopkins [184] which gives quite good results (see [185, 186] for a comparison with other methods), and we concentrate on the case where the sampling region is two-dimensional.

Following Diggle's nomenclature [13], we call the points that constitute the pattern we wish to analyze (e.g. saplings, seedlings and cell centers in Fig. A.1) *events* to distinguish them from other arbitrary points. A stochastic process generating events in an area A is called *completely spatially random* with density λ if

(i) the number of events $N(B)$ in any region $B \subset A$ with area $|B|$, is Poisson-distributed with mean $\lambda|B|$ and

(ii) the number of events in disjoint sets are independent.

Let us assume that A is very large and non-pathologically shaped so that we can neglect finite-size effects. The probability to find at least one other event within a circle of radius r around one randomly selected event in a completely spatially random distribution is then, according to property (i),

$$P(N(\pi r^2) > 0) = 1 - \exp(-\lambda \pi r^2). \tag{A.1}$$

The probability density function for the distance r to the nearest neighbor is the derivative,

$$p(r) = \frac{\mathrm{d}}{\mathrm{d}r} P(N(\pi r^2) > 0) = 2\lambda \pi r \exp(-\lambda \pi r^2), \tag{A.2}$$

or, if we work with the quantity $a = \lambda \pi r^2$ to simplify notation,

$$p(a) = p(r) \frac{\mathrm{d}r}{\mathrm{d}a} = \exp(-a). \tag{A.3}$$

Hence, a is exponentially distributed.

An important consequence of complete spatial randomness (CSR) is that, instead of choosing one *event* at random, we can choose *any point* in A, even one that is not an event, and, without any change in the arguments above, we obtain the same probability distribution $p(a)$ for the distance between that point and the nearest event. For clustered point patterns, on the other hand, an event will typically be closer to the nearest neighbor than a random point would be to the nearest event, since a randomly chosen event is more likely to be within a cluster of other events (see Fig. A.1(b)). The opposite is true for regular point patterns. There, the distance between one event and the nearest other event will typically be larger than the distance between a random point and its nearest event because events tend to "repel" each other (see Fig. A.1(c)).

This observation is the basis of Hopkins' test for CSR. We choose n events randomly, measure the distances r_1, r_2, ..., r_n to the nearest neighboring event for each

case, compute $a_i = \lambda\pi r_i^2$, $i = 1, 2, ..., n$, and add the results,

$$X_{event} = \sum_{i=1}^{n} a_i. \tag{A.4}$$

Then we choose n random points, measure again the distances ρ_1, ρ_2, ..., ρ_n to the nearest event, compute $\alpha_i = \lambda\pi\rho_i^2$, $i = 1, 2, ..., n$, and calculate their sum,

$$X_{point} = \sum_{i=1}^{n} \alpha_i. \tag{A.5}$$

If the events are at completely spatially random positions, then X_{event} and X_{point} are two independent and identically distributed random variables, so that for large enough n they are expected to give essentially the same result. However, if X_{event} is significantly smaller than X_{point}, the point pattern is likely to be aggregated; in the opposite case we would suspect that the pattern is regular.

Whether we accept the null hypothesis of CSR will depend on X_{event} and X_{point} as well as n. We will prove that, for CSR, X_{event} and X_{point} are gamma-distributed, and the so-called Hopkins statistic,

$$H = \frac{X_{point}}{X_{point} + X_{event}}, \tag{A.6}$$

is beta-distributed with parameters depending on n, allowing us to judge the validity of the null hypothesis. To derive these results, let us first review some important properties of the beta and gamma distributions.

The gamma distribution with parameters μ and ν has the probability density function

$$p_{\mu,\nu}(x) = \frac{1}{\Gamma(\mu)\nu^{\mu}} x^{\mu-1} \exp\left(-\frac{x}{\nu}\right), \tag{A.7}$$

and the beta distribution with parameter σ and τ is defined by

$$p_{\sigma,\tau}(x) = \frac{1}{B(\sigma,\tau)} x^{\sigma-1}(1-x)^{\tau-1}. \tag{A.8}$$

Let X_1 and X_2 be independent gamma-distributed random variables with parameters

(μ_1, ν) and (μ_2, ν) respectively. Then their sum $X_1 + X_2$ is gamma-distributed with parameters $(\mu_1+\mu_2, \nu)$, and $\frac{X_1}{X_1+X_2}$ is beta-distributed with parameters (μ_1, μ_2). This can be proved with a simple variable transformation

$$u = x_1 + x_2, \tag{A.9a}$$

$$v = \frac{x_1}{x_1 + x_2}, \tag{A.9b}$$

whose Jacobian is

$$\frac{\partial(x_1, x_2)}{\partial(u, v)} = -u. \tag{A.10}$$

The joint probability density of $X_1 + X_2$ and $\frac{X_1}{X_1+X_2}$ in terms of u and v is hence

$$p(u, v) = p(x_1, x_2) \left| \frac{\partial(x_1, x_2)}{\partial(u, v)} \right| = \frac{1}{\Gamma(\mu_1)\Gamma(\mu_2)\nu^{\mu_1+\mu_2}} u^{\mu_1+\mu_2-1} \exp\left(-\frac{u}{\nu}\right) v^{\mu_1-1}(1-v)^{\mu_2-1}. \tag{A.11}$$

The probability density for $X_1 + X_2$ alone follows by integrating over v,

$$p_{X_1+X_2}(u) = \frac{1}{\Gamma(\mu_1)\Gamma(\mu_2)\nu^{\mu_1+\mu_2}} u^{\mu_1+\mu_2-1} B(\mu_1, \mu_2) \exp\left(-\frac{u}{\nu}\right), \tag{A.12}$$

and that for $\frac{X_1}{X_1+X_2}$ by integrating over u,

$$p_{X_1/(X_1+X_2)} = \frac{1}{\Gamma(\mu_1)\Gamma(\mu_2)\nu^{\mu_1+\mu_2}} \Gamma(\mu_1 + \mu_2) v^{\mu_1-1}(1 - v)^{\mu_2-1}. \tag{A.13}$$

Since $B(\mu_1, \mu_2) = \frac{\Gamma(\mu_1)\Gamma(\mu_2)}{\Gamma(\mu_1+\mu_2)}$, $X_1 + X_2$ is gamma- and $\frac{X_1}{X_1+X_2}$ beta-distributed as claimed.

Eq. (A.3) is the special case of a gamma distribution with $\mu = \nu = 1$. With a simple induction on n, we can show that X_{event} and X_{point} (Eq. (A.4) and (A.5)) are gamma distributed with parameters $\mu = n$, $\nu = 1$. It is similarly easy to prove that the Hopkins statistic, defined in Eq. (A.6), is beta-distributed with parameters $\sigma = \tau = n$,

$$p_H(x) = \frac{x^{n-1}(1 - x)^{n-1}}{B(n, n)}. \tag{A.14}$$

With this last equation, it is a simple exercise to show that the expectation value of

H is $\frac{1}{2}$ and the variance $\frac{1}{4(2n+1)}$.

These results allow us to compute p-values for the null hypothesis of CSR. The p-value is the probability that the observed value of the Hopkins statistic would occur if point position were purely random. If the measured Hopkins statistic is h, then the p-value is equal to

$$p\text{-value} = 1 - \frac{1}{B(n,n)} \int_{1/2-|h-1/2|}^{1/2+|h-1/2|} x^{n-1}(1-x)^{n-1}\mathrm{d}x \tag{A.15}$$

The last result can be expressed in terms of the incomplete beta function

$$B_{inc}(\sigma,\tau,x) = \int_0^x x'^{\sigma-1}(1-x')^{\tau-1}\mathrm{d}x' \tag{A.16}$$

as

$$p\text{-value} = 1 + B_{inc}\left(n,n,\frac{1}{2}+\left|h-\frac{1}{2}\right|\right) - B_{inc}\left(n,n,\frac{1}{2}-\left|h-\frac{1}{2}\right|\right). \tag{A.17}$$

Values of the incomplete beta function have been tabulated. B_{inc} is also a built-in function in computer software such as Mathematica or Matlab.

To demonstrate that the Hopkins statistic agrees with our intuition we can calculate it for the three patterns in Fig. A.1 with $n = 10$. Repeating the calculation 10 times independently for each case, the results are $H = 0.5 \pm 0.1$ for A.1(a), $H = 0.84 \pm 0.05$ for A.1(b), and $H = 0.21 \pm 0.05$ for A.1(c). Hence, the Hopkins statistic could not detect a significant deviation from CSR in the first case, but a tendency towards aggregation in the second and regularity in the third case. The p-value for A.1(b) is 5.2×10^{-4} and for (c) 4.7×10^{-3}, so the deviation from CSR appears statistically significant which is indeed what we intuitively expect by looking at the figure.

APPENDIX B

ALGORITHM FOR CARTOGRAM DISPLACEMENT FIELD CONSTRUCTION

Constants and variables

▷ Length of map in x and y directions
integer LX,LY

▷ Array for densities
float rho_0$[LX+1][LY+1]$, rho$[LX+1][LY+1]$
▷ rho_0$[j][k]$ is the density at $t=0$, rho$[j][k]$ that at $t>0$.

▷ Arrays for the velocity field at integer-valued positions
float gridvx$[LX+1][LY+1]$, gridvy$[LX+1][LY+1]$
▷ gridvx$[j][k]$ is the velocity component in x direction at position (j,k). Similarly for y.

▷ Arrays for the position at $t>0$
float x$[LX+1][LY+1]$, y$[LX+1][LY+1]$
▷ x$[j][k]$ is the x-coordinate for the element that was at position (j,k) at time $t=0$.

▷ Arrays for the velocity field at position (x$[j][k]$, y$[j][k]$)
float vx$[LX+1][LY+1]$, vy$[LX+1][LY+1]$
▷ vx$[j][k]$ is the velocity component in x direction at the position (x$[j][k]$, y$[j][k]$),
vy$[j][k]$ the velocity component in y direction.

Program CARTOGRAM
Initialize rho_0.
If wanted, perform Gaussian blur by Fast Fourier Transform.
Replace rho_0 by cosine Fourier transform in both variables.
$t \leftarrow 0$ ▷ Initialize time.
$h \leftarrow HINITIAL$ ▷ Initialize time step size.
▷ Initialize x, y, vx and vy.
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do x$[j][k] \leftarrow j$
y$[j][k] \leftarrow k$

Call subroutine CALCV(time = 0.0).
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do vx[j][k] $\leftarrow$ gridvx[j][k]
vy[j][k] $\leftarrow$ gridvy[j][k]
while the position arrays x and y have not sufficiently converged
do Call subroutine CALCV(time = $t + h$).
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do ▷ Find the new positions in the following manner. First we take a naive integration step:
$vxminus \leftarrow$ vx[j][k]
$vyminus \leftarrow$ vy[j][k]
$vxplus \leftarrow \vec{v}_x(t + h$, x[$j$][$k$]+$h$*vx[$j$][$k$], y[$j$][$k$]+$h$*vy[$j$][$k$])
$vyplus \leftarrow \vec{v}_y(t + h$, x[$j$][$k$]+$h$*vx[$j$][$k$], y[$j$][$k$]+$h$*vy[$j$][$k$]),
▷ where the velocity $\vec{v}$ at time $t + h$ and position (x[j][k]+h*vx[j][k], y[j][k]+ h*vy[j][k]) can be interpolated from the arrays gridvx and gridvy. Then we expect the new position at time $t + h$ to be:
$xguess \leftarrow$ x[j][k] + 0.5*h*($vxminus$+$vxplus$)
$yguess \leftarrow$ y[j][k] + 0.5*h*($vyminus$+$vyplus$)
▷ Then we make a better approximation by solving the two nonlinear equations:
xappr[j][k] - 0.5*h*$\vec{v}_x(t + h$, xappr[j][k], yappr[j][k]) - x[j][k] - 0.5*h*vx[j][k] = 0,
yappr[j][k] - 0.5*h*$\vec{v}_y(t + h$, xappr[j][k], yappr[j][k]) - y[j][k] - 0.5*h*vy[j][k] = 0
▷ simultaneously for xappr[j][k], yappr[j][k], e.g., using the Newton-Raphson method with ($xguess$, $yguess$) as initial guess. The velocity $\vec{v}$ at time $t + h$ and position (xappr[j][k],yappr[j][k]) can again be interpolated from the arrays gridvx and gridvy. If ($xguess$, $yguess$) and (xappr[j][k], yappr[j][k]) differ by more than some predefined tolerance, reduce step size h, break, and try again.
$t \leftarrow t + h$

for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do x[j][k] $\leftarrow$ xappr[j][k]
y[j][k] $\leftarrow$ yappr[j][k]
vx[j][k] $\leftarrow \vec{v}_x(t + h$, xappr[$j$][$k$], yappr[$j$][$k$])
vy[j][k] $\leftarrow \vec{v}_y(t + h$, xappr[$j$][$k$], yappr[$j$][$k$])
Increase step size h.

Subroutine CALCV(time t)
▷ First calculate the density rho by filling the array with the Fourier coefficients.
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do rho[j][k] ←
exp(-(($\pi * j/LX$)*($\pi * j/LX$)+($\pi * k/LY$)*($\pi * k/LY$))*t)*rho_0[j][k]
▷ Calculate the Fourier coefficients for the partial derivative of rho. Store temporary result in arrays gridvx, gridvy.
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do gridvx[j][k] ← -($\pi * j/LX$)*rho[j][k]
gridvy[j][k] ← -($\pi * k/LY$)*rho[j][k]
Replace rho by cosine Fourier backtransform in both variables.
Replace gridvx by sine Fourier backtransform in the first and cosine Fourier backtransform in the second variable.
Replace gridvy by cosine Fourier backtransform in the first and sine Fourier backtransform in the second variable.
▷ Calculate the velocity field.
for $j \leftarrow 0$ **to** LX
do for $k \leftarrow 0$ **to** LY
do gridvx[j][k] ← -gridvx[j][k]/rho[j][k]
gridvy[j][k] ← -gridvy[j][k]/rho[j][k]

BIBLIOGRAPHY

[1] J. E. Dobson, "No, Einstein didn't say geography is harder than physics," *GIS World*, vol. 10, p. 30, 1997.

[2] E. Raisz, "The rectangular statistical cartogram," *Geographical Review*, vol. 24, pp. 292–296, 1934.

[3] D. Dorling, "Area cartograms: Their use and creation," *Concepts and Techniques in Modern Geography (CATMOG)*, vol. 59, 1996.

[4] C. J. Kocmoud, "Constructing continuous cartograms: A constraint-based approach," Master's thesis, Texas A&M University, College Station, Texas, 1997.

[5] D. Keim, S. North, and C. Panse, "Cartodraw: A fast algorithm for generating contiguous cartgrams," *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, pp. 95–110, 2004.

[6] W. R. Tobler, "A continuous transformation useful for districting," *Annals of the New York Academy of Sciences*, vol. 219, pp. 215–220, 1973.

[7] J. A. Dougenik, N. R. Chrisman, and D. R. Niemeyer, "An algorithm to construct continuous area cartograms," *The Professional Geographer*, vol. 37, pp. 75–81, 1985.

[8] S. M. Gusein-Zade and V. S. Tikunov, "A new technique for constructing continuous cartograms," *Cartography and Geographic Information Systems*, vol. 20, pp. 167–173, 1993.

[9] L. Furuberg, J. Feder, A. Aharony, , and T. Jossang, "Dynamics of invasion percolation," *Physical Review Letters*, vol. 61, p. 2117, 1988.

[10] http://www-personal.umich.edu/~lsander.

[11] P. Prusinkiewicz, "Visual models of morphogenisis," *Artificial Life*, vol. 1, pp. 61–74, 1994.

[12] A. J. Scott, "The optimal network problem: Some computational procedures," *Transportation Research*, vol. 3, pp. 201–210, 1969.

[13] P. J. Diggle, *Statistical analysis of spatial point patterns*. London: Academic Press, 1983.

[14] W. Bunge, *Theoretical geography*. Lund Studies in Geography, Gleerup, 1966.

[15] P. Holme, *Form and Function od complex networks*. PhD thesis, Umeå University, Umeå, 2004.

[16] D. Harvey, "Revolutionary and counter-revolutionary theory in geography and the problem of ghetto formation," *Antipode*, vol. 4, no. 1-13, 1972.

[17] L. E. Fernandez, D. G. Brown, R. W. Marans, and J. I. Nassauer, "Characterizing location preferences in an exurban population: Implications for agent based modeling," *Environment and Planning B*, in press.

[18] W. R. Tobler, "Geographic area and map projections," *Geographical Review*, vol. 53, pp. 59–78, 1963.

[19] S. M. Guseyn-Zade and V. S. Tikunov, "Analog methods in the compilation of areal transformed images," *Mapping Sciences and Remote Sensing*, vol. 31, pp. 49–65, 1994.

[20] B. D. Dent, *Cartography: Thematic Map Design*. Boston: WCB/McGraw-Hill, 5th ed., 1999.

[21] W. Tobler, "Thirty five years of computer cartograms," *Annals of the Association of American Geographers*, vol. 94, pp. 58–73, 2004.

[22] A. Appel, C. J. Evangelisti, and A. J. Stein, "Animating quantitative maps with cellular automata," *IBM Technical Disclosure Bulletin*, vol. 26, pp. 953–956, 1983.

[23] S. Selvin, D. Merrill, S. Sacks, L. Wong, L. Bedell, and J.Schulman, "Transformations of maps to investigate clusters of disease," tech. rep., Lawrence Berkeley Laboratory, University of California, No. LBL-18550, 1984.

[24] C. Cauvin, C. Schneider, and G. Cherrier, "Cartographic transformations and the piezopleth maps method," *The Cartographic Journal*, vol. 26, pp. 96–104, 1989.

[25] J. S. Torguson, "Cartogram: A microcomputer program for the interactive construction of value-by-area cartograms," Master's thesis, University of Georgia, Athens, Georgia, 1990.

[26] H. Edelsbrunner and R. Waupotitsch, "A combinatorial approach to cartograms," *Computational Geometry*, vol. 7, pp. 343–360, 1997.

[27] J. M. Hunter and J. C. Young, "A technique for the construction of quantitative cartograms by physical accretion models," *The Professional Geographer*, vol. 20, pp. 402–407, 1968.

[28] L. Skoda and J. C. Robertson, "Isodemographic map of canada." Geographical Paper No. 50, Department of Environment, Ottawa, 1972.

[29] N. A. Raspolozhenskiy, Y. V. Sventek, and V. S. Tikunov, "The use of electrical models in geography," *Soviet Geography*, vol. 5, pp. 315–321, 1973.

[30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1992.

[31] D. W. Merrill, "Use of a density equalizing map projection in analysing childhood cancer in four california counties," *Statistics in Medicine*, vol. 20, pp. 1499–1513, 2001.

[32] D. Dorling, "Bringing elections back to life," *Geographical Magazine*, vol. 66, pp. 20–21, 1994.

[33] P. A. Klinkner, "Red and blue scare: The continuing diversity of the American electoral landscape," *The forum*, vol. 2, no. 2, Art. 2, 2004.

[34] "State energy data report, 2000." Energy Information Administration, Office of Energy Markets and End Use, U.S. Department of Energy, Washington, DC.

[35] R. MacIntyre, "North American news text supplement." CD-ROM (catalog no. LDC98T30), 1998. Philadelphia: Linguistic Data Consortium.

[36] J. Mitchell, ed., *The Random House Encyclopedia*, p. 667. New York: Random House, 3rd ed., 1990.

[37] D. M. Healy, D. N. Rockmore, P. J. Kostelec, and S. Moore, "FFTs for the 2-sphere—improvements and variations," *Journal of Fourier Analysis and Applications*, vol. 9, pp. 341–385, 2003.

[38] A. Weber, *Über den Standort der Industrien*. Tübingen: JCB Mohr, 1909.

[39] E. Weiszfeld, "Sur le point pour lequel la somme des distances de n points donnés est minimum," *Tohoku Mathematical Journal*, vol. 43, pp. 355–386, 1937.

[40] R. E. Kuenne and R. M. Soland, "Exact and approximate solutions to the multisoure Weber problem," *Mathematical Programming*, vol. 3, pp. 193–209, 1972.

[41] K. E. Rosing, "An optimal method for solving the (generalized) multi-Weber problem," *European Journal of Operational Research*, vol. 58, pp. 414–426, 1992.

[42] N. Megiddo and K. Supowit, "On the complexity of some common geometric location problems," *SIAM Journal on Computing*, vol. 13, pp. 182–196, 1984.

[43] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Upper Saddle River: Prentice-Hall, 1993.

[44] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems. Part II: The p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, pp. 539–560, 1979.

[45] P. Hansen and N. Mladenović, "Variable neighborhood search for the p-median," *Location Science*, vol. 5, pp. 207–226, 1997.

[46] F. Chiyoshi and R. D. Galvão, "A statistical analysis of simulated annealing applied to the p-median problem," *Annals of Operations Research*, vol. 96, pp. 61–74, 2000.

[47] E. S. Correa, M. T. A. Steiner, A. A. Freitas, and C. Carnieri, "A genetic algorithm for the p-median problem," in *Proc. 2001 Genetic and Evolutionary Computation Conference*, pp. 1268–1275, 2001.

[48] K. E. Rosing and M. J. Hodgson, "Heuristic concentration for the p-median: an example demonstrating how and why it works," *Computers and Operations Research*, vol. 29, pp. 1317–1330, 2002.

[49] M. G. C. Resende and R. F. Werneck, "A hybrid heuristic for the p-median problem," *Journal of Heuristics*, vol. 10, pp. 59–88, 2004.

[50] L. Cooper, "Solutions of generalized locational equilibrium models," *Journal of regional science*, vol. 7, pp. 1–18, 1967.

[51] P. J. Sullivan and N. Peters, "A flexible user oriented location-allocation algorithm," *Journal of Environmental Management*, vol. 10, pp. 181–193, 1980.

[52] R. F. Love and H. Juel, "Properties and solution methods for large location-allocation problems," *Journal of the Operational Research Society*, vol. 33, pp. 443–452, 1982.

[53] B. A. Murtagh and S. R. Niwattisyawong, "An efficient method for the multi-depot location-allocation problem," *Journal of the Operational Research Society*, vol. 33, pp. 629–634, 1982.

[54] R. Chen, "Solution of minisum and minimax location-allocation problems with Euclidean distances," *Naval Research Logistics Quarterly*, vol. 30, pp. 449–459, 1983.

[55] I. Bongartz, P. H. Calamai, and A. R. Conn, "A projection method for l_p norm location-allocation problems," *Mathematical Programming*, vol. 66, pp. 283–312, 1994.

[56] J. Brimberg and N. Mladenovic, "A variable neighborhood algorithm for solving the continuous location-allocation problem," *Studies in Locational Analysis*, vol. 10, pp. 1–12, 1996.

[57] S. Arora, P. Raghavan, and S. Rao, "Approximation schemes for Euclidean k-medians and related problems," in *Proc. 30th Annual ACM Symposium on Theory of Computing*, pp. 106–113, 1998.

[58] P. Hansen, N. Mladenović, and E. Taillard, "Heuristic solution of the multi-source Weber problem as a p-median problem," *Operations Research Letters*, vol. 22, pp. 55–62, 1998.

[59] H. M. Ghaziri, "A neural heuristic for the multisource Weber problem," *INFOR*, vol. 37, pp. 226–235, 1999.

[60] S. P. Fekete, J. S. B. Mitchell, and K. Beurer, "On the continuous fermat-weber problem," *Operations Research*, vol. 53, pp. 61–76, 2005.

[61] M. Iri, K. Murota, and T. Ohya, "A fast Voronoi-diagram algorithm with applications to geographical optimization problems," in *Proceedings of the 11th IFIP Conference* (P. Throft-Christensen, ed.), Lecture Notes in Control and Information Sciences, (Berlin), pp. 273–288, Springer, 1984.

[62] A. Suzuki and A. Okabe, "Using Voronoi diagrams," in *Facility location* (Z. Drezner, ed.), pp. 103–118, New York: Springer, 1995.

[63] G. M. Voronoi, "Nouvelles applications des parametres continus a la théorie des formes quadratiques, deuxieme memoire, recherches sur les parallelloedres primitifs.," *Journal für die reine und angewandte Mathematik*, vol. 134, pp. 198–287, 1908.

[64] W. Bunge, "Patterns of location," 1964. Michigan Inter-University Community of Mathematical Geographers, Discussion Paper 3, University of Michigan, Ann Arbor.

[65] A. Getis, "The determination of the location of retail activities with the use of a map transformation," *Economic Geography*, vol. 39, pp. 14–22, 1963.

[66] G. Rushton, "Map transformations of point patterns: Central place patterns in areas of variable population density," *Papers of the Regional Science Association*, vol. 28, pp. 111–129, 1971.

[67] S. M. Gusein-Zade, "Bunge's problem in central place theory and its generalizations," *Geographical Analysis*, vol. 14, pp. 246–252, 1982.

[68] S. Angel and G. M. Hyman, *Urban Fields: A Geometry of Movement for Regional Science*. London: Pion Limited, 1976.

[69] L. Cooper, "Heuristic methods for location-allocation problems," *SIAM Review*, vol. 6, pp. 37–53, 1964.

[70] M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics*. Oxford: Oxford University Press, 1999.

[71] M. A. Mostafavi, C. M. Gold, and M. Dakowicz, "Dynamic Voronoi/Delaunay methods and applications," *Journal of Computers and Geosciences*, vol. 29, pp. 523–530, 2003.

[72] J. Viriakis, "Minimum effort as a determinant of the area, population and density of residential communities," *Ekistics*, vol. 27, pp. 362–371, 1969.

[73] D. S. Palmer, "The placing of service points to minimize travel," *Operational Research Quarterly*, vol. 24, pp. 121–123, 1973.

[74] G. E. Stephan, "Territorial division: The least-time constraint behind the formation of subnational boudaries," *Science*, vol. 196, pp. 523–524, 1977.

[75] C. Cameron, S. Low, and D. Wei, "High density model for server allocation and placement," in *Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, (Marina del Rey), pp. 152–159, 2002.

[76] D. R. Vining, C.-H. Yang, and S.-T. Yeh, "Political subdivision and population density," *Science*, vol. 205, p. 219, 1979.

[77] M. E. J. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary Physics*, 2005. In press.

[78] W. Christaller, *Die zentralen Orte in Süddeutschland.* Darmstadt: Wissenschaftliche Buchgesellschaft, 3rd ed., 1980.

[79] G. E. Stephan, "The distribution of service establishments," *Journal of Regional Science*, vol. 28, pp. 29–40, 1987.

[80] S. M. Gusein-Zade, "Alternative explanations of the dependence of the density of centers on the density of population," *Journal of Regional Science*, vol. 33, pp. 547–558, 1993.

[81] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, p. 47, 2002.

[82] S. N. Dorogovtsev and J. F. F. Mendes, "Evolution of networks," *Advances In Physics*, vol. 51, p. 1079, 2002.

[83] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, p. 167, 2003.

[84] W. L. Garrison, "Connectivity of the Interstate Highway systerm," *Papers and proceesings of the Regional Science Association*, vol. 6, pp. 121–137, 1960.

[85] K. J. Kansky, *Structure of transportation networks: Relationships between network geometry and regional characteristics.* University of Chicago: Department of Geography, 1963.

[86] P. Haggett and R. J. Chorley, *Network analysis in geography.* New York: St. Martin's Press, 1969.

[87] L. B. Leopold, "Trees and streams: The efficiency of branching patterns," *Journal of Theoretical Biology*, vol. 31, pp. 339–354, 1971.

[88] T. A. McMahon and R. E. Kronauer, "Tree structures: Deducing the principle of mechanical design," *Journal of theoretical biology*, vol. 59, pp. 443–466, 1976.

[89] M. Zamir, "Optimality principles in arterial branching," *Journal of Theoretical Biology*, vol. 62, pp. 227–251, 1976.

[90] P. S. Stevens, *Patterns in nature*. Boston: Little, Brown and Company, 1974.

[91] S. Redner, "How popular is your paper? An empirical study of the citation distribution," *European Physical Journal B*, vol. 4, pp. 131–134, 1998.

[92] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, p. 41, 2001.

[93] R. Albert, H. Jeong, and A.-L. Barabási, "The diameter of the world-wide web," *Nature*, vol. 401, p. 130, 1999.

[94] B. A. Huberman and L. A. Adamic, "Internet: Growth dynamics of the world-wide web," *Nature*, vol. 401, p. 131, 1999.

[95] P. Sen, S. Dasgupta, A. Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna, "Small-world properties of the Indian railway network," *Physical Review E*, vol. 67, p. 036106, 2003.

[96] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," *Computer Communications Review*, vol. 29, pp. 251–262, 1999.

[97] D. J. Watts, *Small Worlds*. Princeton University Press, 1999.

[98] S. Yang, N. Hsu, and W. W.-G. Louie, P. W. F.and Yeh, "Water distribution network reliability: Connectivity analysis," *Journal of Infrastructure Systems*, vol. 2, pp. 54–64, 1996.

[99] R. Guimerà, S. Mossa, A. Turtschi, and L. A. N. Amaral. Preprint cond-mat/0312535, 2003.

[100] M. S. Mizruchi, *The American Corporate Network, 1904-1974*. Beverley Hills: Sage, 1982.

[101] L. E. Miller, "Distribution of link distances in a wireless network," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, pp. 401–412, 2001.

[102] W. L. Garrison and D. F. Marble, "Factor-analytic study of the connectivity of a transportation network," *Regional Science Association Papers*, vol. 12, pp. 231–238, 1964.

[103] L. Sen, "The geometric structure of an optimal transport network in a limited city—Hinterland case," *Geographical Analysis*, vol. 3, pp. 1–14, 1971.

[104] E. J. Taaffe and H. L. Gauthier, *Geography of transportation*. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

[105] J. C. Lowe and S. Moryadas, *The Geography of Movement*. Boston: Houghton Mifflin, 1975.

[106] A. Cliff, P. Haggett, and K. Ord, "Graph theory and geography," in *Applications of graph theory* (R. J. Wilson and L. W. Beineke, eds.), pp. 293–326, London, New York: Academic Press, 1979.

[107] L. A. N. Amaral, A. Scala, M. Barthélémy, and H. E. Stanley, "Classes of small-world networks," *Proceedings of the National Academy of Sciences USA*, vol. 97, pp. 11149–11152, 2000.

[108] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, pp. 161–179, 1995.

[109] A.-L. Barabási and A. R., "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[110] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[111] J. M. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, p. 845, 2000.

[112] E. N. Gilbert, "Random plane networks," *Journal of the Society for Industrial and Applied Mathematics*, vol. 9, pp. 533–553, 1961.

[113] J. Dall and M. Christensen, "Random geometric graphs," *Physical Review E*, vol. 66, p. 016121, 2002.

[114] M. Penrose, *Random Geometric Graphs*. Oxford Studies in Probability, Oxford: Oxford University Press, 2003.

[115] I. Glauche, W. Krause, R. Sollacher, and M. Greiner, "Continuum percolation of wireless ad hoc communication networks," *Physica A*, vol. 325, pp. 577–600, 2003.

[116] R. M. D'Souza, S. Ramanathan, and D. Temple Lang, "Measuring performance of ad hoc networks using timescales for information flow," Proceedings of IEEE INFOCOM, 2003.

[117] G. E. Pike and C. H. Seager, "Percolation and conductivity: A computer study. I," *Physical Review B*, vol. 10, pp. 1421–1434, 1973.

[118] A. Drory, "Theory of continuum percolation. I. General formalism," *Physical Review E*, vol. 54, pp. 5992–6002, 1996.

[119] R. Meester and R. Roy, *Continuum Percolation*. Cambridge: Cambridge University Press, 1996.

[120] B. Delaunay, "Sur la sphére vide," *Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk*, pp. 793–800, 1934.

[121] M. de Berg, O. Schwarzkopf, M. van Kreveld, and M. Overmars, *Computational geometry : Algorithms and applications*. Berlin: Springer, 2000.

[122] M. D. Penrose and J. E. Yukich, "Central limit theorems for some graphs in computational geometry," *Annals of Applied Probability*, vol. 11, pp. 1005–1041, 2001.

[123] C. Hermann, M. Barthelemy, and P. Provero, "Connectivity distribution of spatial networks," *Physical Review E*, vol. 68, p. 026128, 2003.

[124] A. F. Rozenfeld, R. Cohen, D. ben Avraham, and S. Havlin, "Scale-free networks on lattices," *Physical Review Letters*, vol. 89, p. 218701, 2002.

[125] C. P. Warren, L. Sander, and I. M. Sokolov, "Geography in a scale-free network model," *Physical Review E*, vol. 66, 2002.

[126] P. Sen, K. Banerjee, and T. Biswas, "Phase transitions in a network with a range-dependent connection probability," *Physical Review E*, vol. 66, p. 037102, 2002.

[127] T. Petermann and P. De Los Rios, "Spatial small-world networks: A wiring-cost perspective." Preprint cond-mat/0501420, 2005.

[128] R. T. Wong, "A survey of network design problems," Working Paper OR 080-78, Massachusetts Institute of Technology, 1978.

[129] J. W. Billheimer and P. Gray, "Network design with fixed and variable cost elements," *Transportation Science*, vol. 7, pp. 49–74, 1973.

[130] S.-H. Yook, H. Jeong, and A.-L. Barabási, "Modeling the internet's large-scale topology," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, pp. 13382–13386, 2002.

[131] A. Lakhina, J. W. Byers, M. Crovella, and I. Matta, "On the geographic location of internet resources," *IEEE Journal on Selected Areas in Communications, Special Issue on Internet and WWW Measurement, Mapping, and Modeling*, vol. 21, pp. 934 – 948, 2003.

[132] B. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, 1988.

[133] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation," in *Proceedings of MASCOTS '01: The International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2001.

[134] S. S. Manna and P. Sen, "Modulated scale-free network in Euclidean space," *Physical Review E*, vol. 66, p. 066114, 2002.

[135] P. Sen and S. S. Manna, "Clustering properties of a generalized critical Euclidean network," *Physical Review E*, vol. 68, p. 026104, 2003.

[136] R. Xulvi-Brunet and I. M. Sokolov, "Evolving networks with disadvantaged long-range connections," *Physical Review E*, vol. 66, p. 026118, 2002.

[137] M. Barthélemy, "Crossover from scale-free to spatial networks," *Europhysics Letters*, vol. 63, pp. 915–921, 2003.

[138] A. Barrat, M. Barthélemy, and A. Vespignani, "The effects of spatial constraints on the evolution of weighted complex networks," *Journal of Statistical Mechanics*, p. P05003, 2005.

[139] M. Kaiser and C. C. Hilgetag, "Spatial growth of real-world networks," *Physical Review E*, vol. 69, p. 036103, 2004.

[140] A. Fabrikant, E. Koutsoupias, and C. H. Papadimitriou, "Heuristically optimized trade-offs: A new paradigm for power laws in the internet," in *ICALP*, vol. 2380 of *Lecture Notes in Computer Science*, pp. 110–112, Springer, 2002.

[141] N. Berger, B. Bollobás, C. Borgs, J. Chayes, and O. Riordan, "Degree distribution of the FKP network model," in *ICALP*, pp. 725–738, 2003.

[142] N. Berger, C. Borgs, R. D'Souza, and R. D. Kleinberg, "Competition-induced preferential attachment," in *ICALP*, pp. 208–221, 2004.

[143] J. I. Alvarez-Hamelin and N. Schabanel, "An internet graph model based on trade-off optimization," *European Physical Journal B*, vol. 38, pp. 231–237, 2004.

[144] R. Lenormand and S. Bories, "Description d'un mécanisme de connexion de liaisons destiné à l'étude du drainage avec piegeage en milieu poreaux," *C. R. Acad. Sci. Paris B*, vol. 291, p. 279, 1980.

[145] R. Chandler, J. Koplik, K. Lerman, and J. Willemsen, "Capillary displacement and percolation in porous media," *Journal of Fluid Mechanics*, vol. 119, p. 249, 1982.

[146] D. Stauffer and A. Aharony, *Introduction to percolation theory*. London: Taylor & Francis, 1994.

[147] T. A. Witten and L. M. Sander, "Diffusion-limited aggregation, a kinetic critical phenomenon," *Physical Review Letters*, vol. 47, pp. 1400–1403, 1981.

[148] T. A. Witten and L. M. Sander, "Diffusion-limited aggregation," *Physical Review B*, vol. 27, pp. 5686–5697, 1983.

[149] M. Matsushita and H. Fujikawa, "Diffusion-limited growth in bacterial colony formation," *Physica A*, vol. 168, pp. 498–506, 1990.

[150] M. Batty, P. Longley, and S. Fotheringham, "Urban growth and form: Scaling, fractal geometry, and diffusion-limited aggregation," *Environment and Planning A*, vol. 21, pp. 1447–1472, 1989.

[151] M. Eden, "A two-dimensional growth process," in *Proc. Fourth Berkeley Symposium Math. Statistics and Probability*, (Berkeley, CA), pp. 223–239, University of California Press, 1961.

[152] R. Jullien and R. Botet, "Scaling properties of the surface of the Eden model in d=2, 3, 4," *Journal of Pysics A*, vol. 18, pp. 2279–2287, 1985.

[153] N. Vandewalle and M. Ausloos, "Lacunarity, fractal, and magnetic transition behaviors in a generalized Eden growth process," *Physical Review E*, vol. 50, pp. R635–R638, 1994.

[154] A. Brú, S. Albertos, J. L. Subiza, J. L. Garca-Asenjo, and I. Brú, "The universal dynamics of tumor growth," *Biophysical Journal*, vol. 85, pp. 2948–2961, 2003.

[155] L. Benguigui, "A new aggregation model. application to town growth," *Physica A*, vol. 219, pp. 13–26, 1995.

[156] T. Vicsek, *Fractal growth phenomena*. Singapore: World Scientific, 2nd ed., 1992.

[157] M. R. Garey, R. L. Graham, and D. S. Johnson, "Some NP-complete geometric problems," in *Proc. 8th Annual ACM Symp. on Theory of Computing*, pp. 10–22, 1976.

[158] F. K. H. Ding-Zhu Du, "A proof of the Gilbert-Pollak conjecture on the Steiner ratio," *Algorithmica*, vol. 7, pp. 121–135, 1992.

[159] A. Kansal and S. Torquato, "Globally and locally minimal weight spanning tree networks," *Physica A*, vol. 301, pp. 601–619, 2001.

[160] J. F. McCarthy, "Invasion percolation on a random lattice," *Journal of Physics A*, vol. 20, pp. 3465–3469, 1987.

[161] S. H. Strogatz, *Nonlinear Dynamics and Chaos*. Cambridge, MA: Westview Press, 1994.

[162] W. R. Black, *Transportation: A Geographical Analysis*. New York, NY: Guilford Press, 2003.

[163] http://www.uni-konstanz.de/zwn/winterschool2004/CONTR/Villamil_HAND.pdf.

[164] C. D. Murray, "The physiological principle of minimum work. I. The vascular system and the cost of blood volume," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 12, pp. 207–214, 1926.

[165] F. Nekka, S. Kyriacos, C. Kerrigan, and L. Cartilier, "A model of growing vascular structures," *Bull. Math. Biol.*, vol. 58, pp. 409–426, 1996.

[166] D. Moore, L. J. McNulty, and A. Meskauskas, *Branching Morphogenesis*. Georgetown: Landes Bioscience/Eurekah.com, 2005.

[167] M. Minoux, "Network synthesis and optimum network design problems: Models, solution methods and applications," *Networks*, vol. 19, pp. 313–360, 1989.

[168] M. Los and L. C., "Combinatorial programming, statistical optimization and the optimal transportation network problem," *Transportation Research-B*, vol. 16B, pp. 89–124, 1982.

[169] T. C. Hu, "Optimum communication spanning trees," *SIAM Journal on Computing*, vol. 3, pp. 188–195, 1974.

[170] D. E. Boyce, A. Farhi, and R. Weischedel, "Optimal network problem: A branch-and-bound algorithm," *Environment and Planning*, vol. 5, pp. 519–533, 1973.

[171] H. H. Hoang, "A computational approach to the selection of an optimal network," *Management Science*, vol. 19, pp. 48–498, 1973.

[172] R. Dionne and F. M., "Exact and approximate algorithms for optimal network design," *Networks*, vol. 9, pp. 37–59, 1979.

[173] D. S. Johnson, J. K. Lenstra, and A. H. G. Rinnoy Kan, "The complexity of the network design problem," *Networks*, vol. 8, pp. 279–285, 1978.

[174] G. Ramalingam and T. Reps, "On the computational complexity of dynamic graph problems," *Theoretical Computer Science A*, vol. 158, pp. 233–277, 1996.

[175] P. L. Krapivsky, S. Redner, and F. Leyvraz, "Connectivity of growing random networks," *Physical Review Letters*, vol. 85, pp. 4629–4632, 2000.

[176] H. Jeong, Z. Néda, and A.-L. Barabási, "Measuring preferential attachment in evolving networks," *Europhysics Letters*, vol. 61, pp. 567–572, 2003.

[177] D. Braess, "Über ein Paradoxon der Verkehrsplanung," *Unternehmensforschung*, vol. 12, pp. 258–268, 1968.

[178] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences in the United States of America*, vol. 99, pp. 7821–7826, 2002.

[179] M. Numata, "Forest vegetation in the vicinity of choshi. coastal flora and vegetation at choshi, chiba prefecture," *IV. Bull. Choshi Marine Lab. Chiba Univ.*, vol. 3, pp. 28–48, 1961.

[180] D. J. Strauss, "A model for clustering," *Biometrika*, vol. 62, pp. 467–475, 1975.

[181] B. D. Ripley, "Modelling spatial patterns," *Journal of the Royal Statistical Society B*, vol. 39, pp. 172–212, 1977.

[182] F. H. C. Crick and P. A. Lawrence, "Compartments and polychones in insect development," *Science*, vol. 189, pp. 340–347, 1075.

[183] B. D. Ripley, *Spatial Statistics*. New York: Wiley & Sons, 1981.

[184] B. Hopkins, "A new method for determining the type of distribution of plant individuals," *Annals of Botany*, vol. 18, pp. 213–227, 1954. Appendix by J. G. Skellam.

[185] P. J. Diggle, J. Besag, and J. T. Gleaves, "Statistical analysis of spatial point patterns by means of distance methods," *Biometrics*, vol. 32, pp. 659–667, 1976.

[186] K. Byth and B. D. Ripley, "On sampling patterns by distance methods," *Biometrics*, vol. 36, pp. 279–284, 1980.