Data Analysis through Polyhedral Theory

From Land Consolidation to Circuit Diameters

Kumulative Habilitationsschrift

Steffen Borgwardt
Technische Universität München

Preface

I am interested in mathematical optimization and data analysis, two fields that concern themselves with educated decision-making. They tie into business analytics, life science, machine learning, computer science, and more. My work is based on the studies of high-dimensional objects arising in operations research and the analysis of big data. The geometric properties of these objects contain information about the combinatorial structure of underlying problems, and therefore show approaches for practice.

I see optimization as a beautiful blend of mathematical theory and working in applications. The first part of this thesis, and the application that originally got me interested in data analysis, is on land consolidation. Already in 2001, Prof. Peter Gritzmann and Prof. Andreas Brieden began with the development of algorithms which formed an alternative approach for the classical form of land consolidation in agriculture, which was applied to several communities in Northern Bavaria [33, 34].

In my Ph.D. and postdoctorate time, we refined the methods for land consolidation in agriculture [19, 22, 23, 25, 35, 36], and I moved to studies in land consolidation in forestry. In particular, I lead an R&D-project for the Bavarian State Ministry of Agriculture, Nutrition, and Forests in 2012/13. It was concerned with the additional challenges arising in private forest areas [30, 31] and involved the engineering of software, and the application of methods and software in forest regions. The joint theoretical work on land consolidation and the practical implementation in both agriculture and forestry were recognized by the Euro Excellence in Practice Award 2013.

The methods originally devised for land consolidation (in dimension two) work for all dimensions, and proved to be powerful tools for structuring point sets in arbitrary dimension - showing the way for a broad field of application. These require the ability to deal with big data and noisy data sets. This lead to generalizations of classical methods in machine learning: weight-balanced k-means [24] (a generalization of k-means [76]) and soft power diagrams [21] (a generalization of multiclass separation [12]).

Further, the above methods reveal an intimate connection to transportation polytopes. In particular, both the approaches in [24, 25] are based on linear programming over transportation polytopes as a subroutine. In the context of a best-case performance of the Simplex method, the study of the combinatorial diameter of polyhedra is a classical field in the theory of linear programming. In particular, the famous Hirsch Conjecture is open for transportation polytopes, and edge walks in these polytopes allow intuitive combinatorial interpretations. This lead me to the studies of the diameters of these polyhedra. In particular, I worked on the combinatorial diameter of partition polytopes [20], $3\times N$–transportation polytopes [29] and dual network flow polyhedra [27].

However, the Simplex algorithm is not the only viable approach for linear programming over many polyhedra, for example those with N-fold constraint matrices such as transportation polytopes. Here using steps of improvement along the so-called circuits of the underlying constraint matrix exhibits favorable properties [42, 43]. Circuits are the elementary vectors of a matrix. For a given matrix, they are the directions of edges that appear in at least one polyhedron defined by the given matrix in combination with some (varying) right-hand side.

A circuit walk, and the circuit diameter, then are natural generalizations of an edge walk and the combinatorial diameter, with certain restrictions imposed on these walks. For example, one may require the steps to remain in the polyhedron and to use maximal step lengths, so that one does not stop in the relative interior of the polytope after any step. The different restrictions on circuit walks yield a hierarchy of circuit diameters that exhibits some interesting properties [28]: there are classes of polytopes for which the combinatorial diameter is much larger than all circuit diameters. Further, the Hirsch Conjecture bound holds in large parts of the hierarchy [28]. For this, I believe the circuit diameters can become a key tool in distinguishing which classes of polyhedra satisfy the Hirsch Conjecture for the combinatorial diameter and which do not.

This thesis contains papers on land consolidation, machine learning, combinatorial diameters and circuit diameters. It encompasses five published papers about my work from 2011 to early 2014 [20, 21, 25, 26, 31] and three papers publically available on arXiv since 2014 [24, 28, 29].

Scientific Background

After receiving double diploma degrees in mathematics and computer science in 2007, I joined the chair for Applied Geometry and Discrete Mathematics of Prof. Peter Gritzmann at the Technische Universität München as a research assistant and Ph.D. student. In December 2010, I received my Ph.D. with a thesis in this field [19] and started my postdoctorate. I am currently a visiting assistant professor at the mathematics department of the University of California Davis. Prior to that I held postdoctoral appointments at the Technische Universität München and an acting professor position at the Technische Universität Braunschweig. In 2012/13 I lead the R&D-project 'ArborTec' for the Bavarian State Ministry for Agriculture, Nutrition and Forestry.

Contents

1 Data analysis through applied geometry

The ability to extract new information from large data sets is a key step in today's decision making processes. This makes data analysis important in many fields from genetics over sociology to operations research [13, 47, 56, 67, 86]. Data analysis tasks consist of three steps: a representation of the data set, the definition of a similarity measure and a grouping process [66]. The latter consists of two categories: **clustering** or **classification**. We begin with a brief review of these tasks and some core concepts of constraints and typical objective functions.

1.1 Clustering and classification

Clustering is the actual process of grouping a data set X into clusters $C_1, \dots, C_k$ for a prescribed k, depending on its representation and a chosen similarity measure. The term is also used for the resulting assignment of data vectors to clusters.

There are many ways of approaching a clustering problem, such as hierarchical algorithms that determine a hierarchical structure of clusterings based on the similarity of data vectors [71, 88, 94] or fuzzy clustering where each data vector has a variable degree of membership in each of the clusters [78].

The most popular clustering approach however, and the one we are interested in, are **partitioning algorithms** that determine a single clustering being optimal with respect to some criterion. The input for a partitioning algorithm typically contains the number

k of clusters to be created. A classical example for this type of algorithms is the k-means algorithm [76]. See [13, 46, 67, 74, 79, 98] for some background on partitioning algorithms.

On the other hand, the process of classification begins with a given clustering $C_1, \dots, C_k$ of X. The task then is to derive a so-called **classifier**, a rule that explains to which of the existing clusters a new data point should be assigned. A new data vector then is associated with one of the existing clusters, depending on the similarity measure and given clustering.

Other tasks like prediction, used to complement a data point 'missing' one (or more) coefficients, and outlier detection are special cases of clustering and classification, and benefit from combining methods for both fields [79, 90, 96].

1.2 Constrained clustering

There is a vast amount of literature on general data analysis, and in particular on clustering. However, in many clustering problems one wants to create clusters that satisfy some constraints. This field is called **constrained clustering** [11], which is much less understood. Such constraints are either hard requirements for the application at hand or are used to integrate a-priori knowledge into a clustering algorithm to 'guide' it to create a clustering of favorable properties.

My work on data analysis began with an application in agriculture, which can be modelled as clustering with **balancing constraints** for a weighted data set. Balancing constraints typically take the form of size restrictions on the clusters, which bound the number of data vectors in the clusters in various ways [55, 100, 62]. For example, one could ask for all clusters to contain the same amount of data vectors, or for clusters to satisfy certain lower and upper bounds on the number of data vectors they contain.

The desire to create clusters of prescribed sizes also often arises when one wants to continue working with the derived clustering. A typical situation is that standard operations from statistics are applied to the clusters themselves. Then their running time depends on the size of a largest cluster – and is optimal for clusters of equal sizes. See [11] for an introduction into constrained clustering, and further real-world applications.

When using weighted data sets, the cluster sizes are put into relation to the weights of the points. The use of weighted data sets also allows for a natural representation of identical, repeated points in data sets. Let us motivate our interest in this kind of clustering by the aforementioned application.

1.3 An application in land consolidation

In many agricultural regions in Bavaria, a small number of farmers cultivates a large number of small-sized lots that are scattered over the region. Figure 1 shows an example. The different colors indicate the farmers that cultivate the lots.

Here the farmers face serious disadvantages in considerable overhead driving and prohibitive use of heavy machinery. Hence, the cost of cultivation is much higher than it would be for fewer, larger lots of the same total size. Calculations of the *Bavarian State Institute for Agriculture* show that these additional costs often add up to more than 30% of the part of the farmers' net income from their agricultural production.

In such a situation, typically a classical land consolidation process is initiated. It consists of a complete restructuring of the agricultural area, discarding the current and

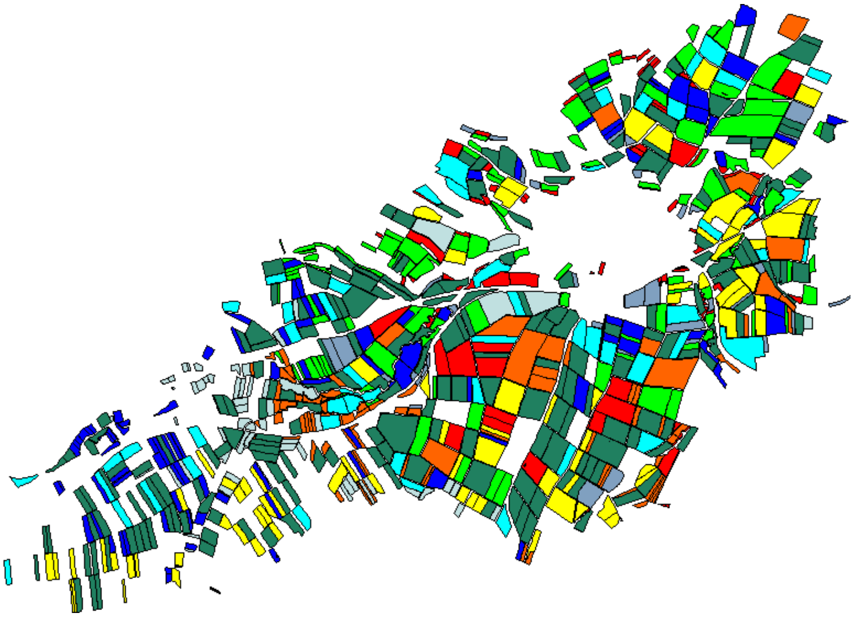


Figure 1: An agricultural region with 9 farmers and 979 lots. Different colors represent different farmers who cultivate the lots.

creating a new lot structure, along with planning of new infrastructure. This involves surveying and reassignment of legal property, which is a costly and time-intensive process (often lasting more than a decade and costing more than 2500 Euro per hectare).

A conceptually much simpler alternative employs voluntary lend-lease agreements. Here the right to till fields is swapped between participating farmers in order to create large connected pieces of land. The existing lot structure remains unchanged. Figure 2 depicts such an improvement over the original situation.

A key aspect is that all the lots vary in value (they have different size, quality of soil, and subsidies attached to them). In the course of redistribution, the total value of each farmer's land should not change much. By thinking of each farmer as a cluster, by representing the lots by points in the Euclidean plane, and by using their values as weights, we arrive at a geometric clustering problem with balancing constraints with respect to weighted points.

1.4 Weight-balanced clustering

We begin with some basic notation and wording. Let $k, n, d \in \mathbb{N}$ with $n \geq k \geq 2$. Let $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ be a data set of distinct points with associated weights $b = (b_1, \ldots, b_n)$ and $b_j > 0$ for all $j \leq n$. Further k denotes the number of clusters, and $a = (a_1, \ldots, a_k)$ the prescribed sizes of clusters to compute. Clearly $\sum\limits_{i=1}^{k} a_i = \sum\limits_{j=1}^{n} b_j$.

In land consolidation, $d = 2$, the x_j are the centers of the lots, and the b_j are the values of the lots. Further, the a_i represent the total value of the i-th farmer's original lots. For the sake of simplicity, we here describe our model with respect to fixed prescribed cluster sizes. The methods can be transferred to clustering with upper and lower bounds on the sizes of the clusters, and thus cover a large range of practical applications. We will exploit this later on.

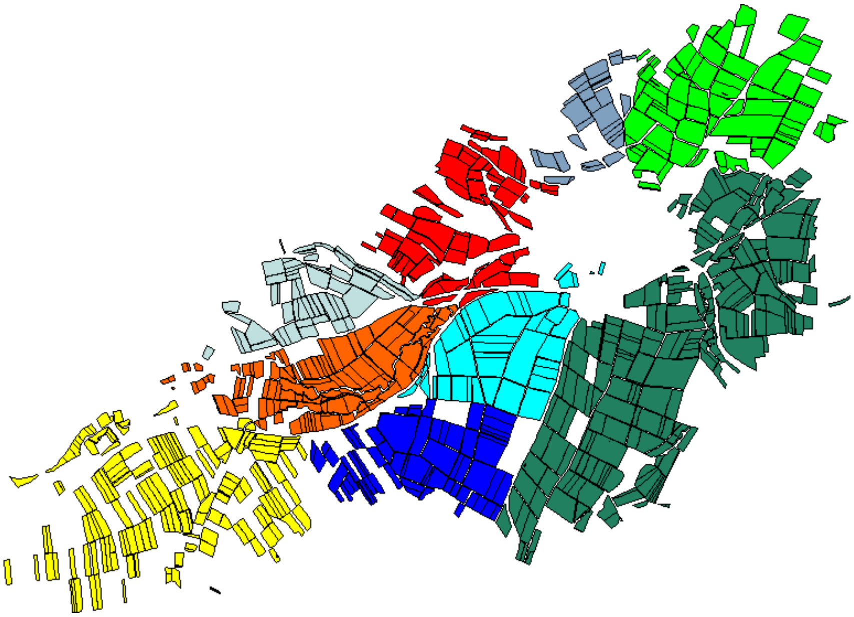


Figure 2: An improved redistribution of lots for the agricultural region of Figure 1.

A **partial membership k-clustering** $\mathcal{C} = (C_1, \ldots, C_k)$ of X consists of k clusters C_i, and is defined by an assignment vector $y = (y_{11}, \ldots, y_{1n}, \ldots, y_{k1}, \ldots, y_{kn})^T \in [0,1]^{kn}$ of X with $\sum\limits_{i=1}^{k} y_{ij} = b_j$ for all $j \leq n$. Informally, y_{ij} is the partial weight of point x_j that belongs to C_i. Formally, we set $C_i = (y_{i1}, \ldots, y_{in})$. The **support** of the cluster C_i is $\text{supp}(C_i) = \{x_j : y_{ij} > 0\}$, the support of the clustering $\mathcal{C}$ is the tuple $\text{supp}(\mathcal{C}) = (\text{supp}(C_1), \ldots, \text{supp}(C_k))$. We use the notation $|C_i| = \sum\limits_{j=1}^{n} y_{ij}$ to refer to the total weight or the **size** of cluster C_i. The tuple $|C| = (|C_1|, \ldots, |C_k|)$ is the **shape** of C.

With the variables y_{ij} indicating how much of the weight b_j of point x_j is associated to a cluster C_i, the feasible region for the y can be described by the set of linear inequalities

$$\begin{array}{rcll} \sum\limits_{j=1}^{n} y_{ij} & = & a_i & (i \leq k) \\ \sum\limits_{i=1}^{k} y_{ij} & = & b_j & (j \leq n) \\ y_{ij} & \geq & 0 & (i \leq k, j \leq n). \end{array}$$

The first line in this system implies that each cluster has the correct size, whereas the second line guarantees that each point is fully assigned. Note that these constraints define a special transportation polytope, we call it the **weight-balanced partition polytope**. The need to allow for partial membership of points in several clusters comes from the combination of balancing constraints with weighted-points [22, 25]. In fact, it is $\mathbb{NP}$–hard to decide whether there then is an assignment of points to clusters without splitting up a point between two or more clusters. In other words, it is hard to decide whether a weight-balanced partition polytope has an integral point.

It remains to turn to viable objective functions for clustering problems. Here we use the representation of the data set in Euclidean space and begin by aiming for 'linear separation' of clusters.

1.5 Linear separation

Data sets are often represented as sets of points $X \subset \mathbb{R}^d$ in d-dimensional Euclidean space. The Euclidean distance has a high appeal as a similarity measure, as humans intuitively group two- or three-dimensional data vectors according to their Euclidean distance [67]. A prime example for this is Figure 2, which can be identified as a 'good' distribution of farmland within a split-second.

For both clustering and classification, it often is desired to have **linear separation** in these spaces [10, 79], i.e. clusters are separated by hyperplanes that also act as classifiers: depending on which halfspace a new data point lies in, it is considered to be associated to the corresponding clustering.

In many applications other non-linear forms of separation are desired. However even in these cases, the methods for linear separation are transferred by the use of a kernel function [52, 84, 87, 93]. Geometrically, one changes the representation of a data set by mapping it to a higher-dimensional inner product space in which one aims for linear separation once again. But instead of working with this different representation of the data set, a kernelizable algorithm can be performed by replacing all inner products of points of the original input by values of this kernel function.

For two clusters, linear separation is intuitive; two clusters have a single separating hyperplane. Consider Figure 3 *a*). The straightforward way of extending this notion to multiple clusters to have a separating hyperplane for each pair of clusters is depicted in Figure 3 *b*).

However, this approach exhibits an unfavorable property. There may be regions of the underlying space for which these hyperplanes provide conflicting information in classification tasks. While one is able to 'fix' this for the example in Figure 3 *b*) by moving the hyperplanes slightly, one cannot for the example in Figure 3 *c*).

Figure 3 *d*) depicts an improvement over the situation in Figure 3 *b*). Here the space is partitioned into a polyhedral cell complex, where each cluster has its own cell. In fact, this is what was done to obtain the redistributed agricultural region in Figure 2. Let us discuss how to obtain such a mode of separation.

1.6 Separating power diagrams

The best-known polyhedral cell complexes in $\mathbb{R}^d$ are the Voronoi diagrams [7]. They appear in many applications and algorithms such as the classical k-means algorithm [76].

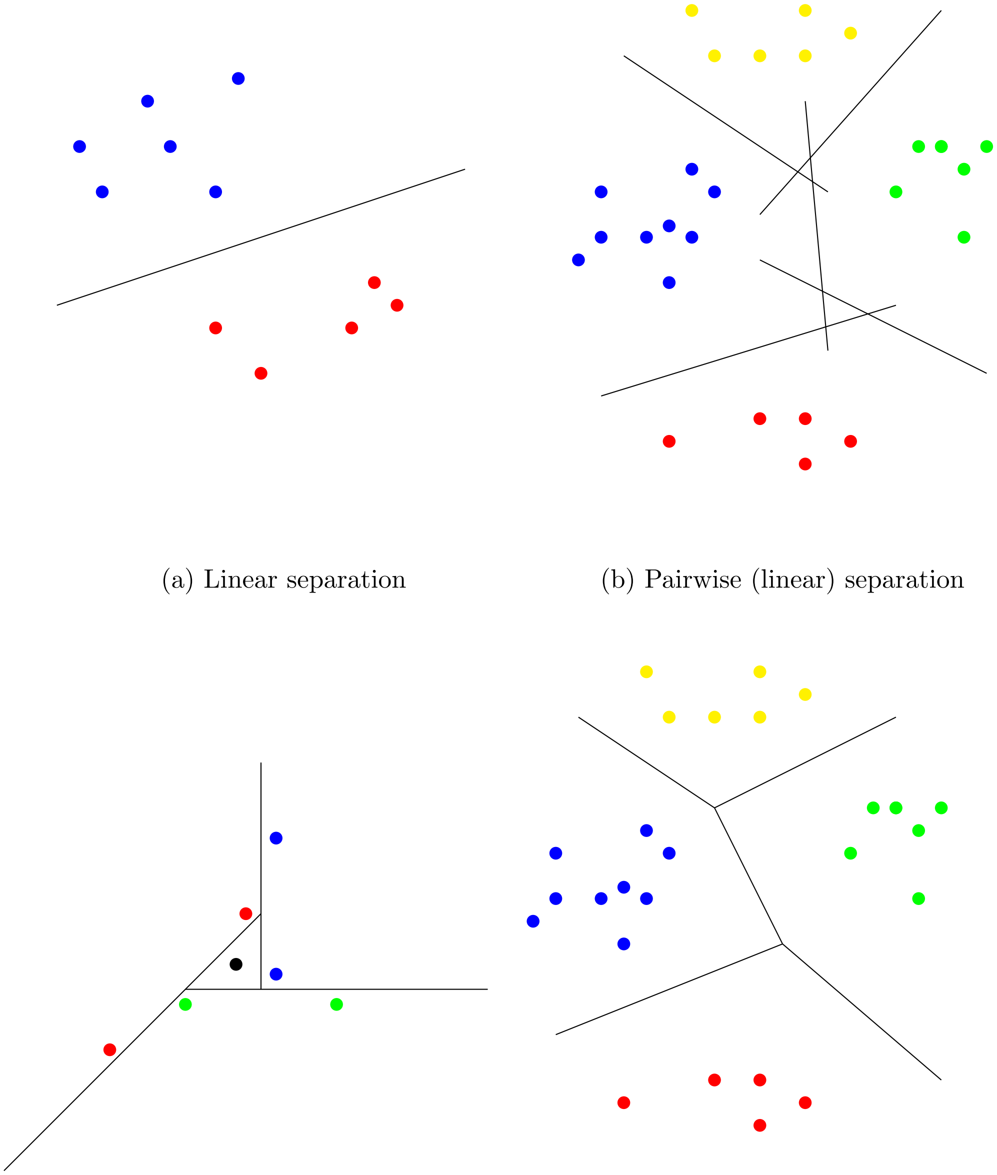
A natural and powerful generalization of Voronoi diagrams are the so-called **power diagrams** [5]. The cell P_i of such a power diagram is defined by a site $s_i \in \mathbb{R}^d$ and a real number $w_i \geq 0$. It consists of all the points $x \in \mathbb{R}^d$ which are closest to the site, where this distance is measured by the so-called **power function**

$$p_i(x) = \|s_i - x\|^2 - w_i.$$

Informally, the power function is the distance of x to the closest point on a sphere of radius $\sqrt{w_i}$ around site s_i. We notate the set of sites as $S = \{s_1, \dots, s_k\}$ and the set of numbers as $\omega = (w_1, \dots, w_k)$.

An (S, ω)**-power diagram** then is a decomposition $P = (P_1, \dots, P_k)$ of $\mathbb{R}^d$ with

$$P_i = \{x \in \mathbb{R}^d : \|s_i - x\|^2 - w_i \leq \|s_j - x\|^2 - w_j \quad \text{for all } j \neq i\}$$



(a) Linear separation

(b) Pairwise (linear) separation

(c) Issues of pairwise linear separation

(d) Separating power diagram

Figure 3: Separation in data analysis.

if $\dim(P_i) = d$ for all $i \leq k$. Here $\dim(P_i)$ refers to the dimension of the affine hull of P_i.

These are the objects we want for the separation of our clusters. In machine learning, they are the classifiers of the so-called altogether models for multiclass support vector machines [32, 40, 91, 97]. In the literature, these kinds of classifiers also appear as piecewise-linear separability [12] and full S-induced cell decompositions [19].

Formally, for a clustering $\mathcal{C}$ of X and P being a power diagram, we say that $\mathcal{C}$ allows the **separating (or feasible) power diagram** P if $\text{supp}(C_i) \subset P_i$ for all $i \leq k$ and $\text{supp}(C_i) \not\subset P_j$ for all $i \neq j$. We also say that P is a separating power diagram for $\mathcal{C}$. Informally, for all $i \leq k$, all points that are at least partially assigned to cluster C_i lie in P_i. Note that points may also lie on the boundary of the cell, i.e. $x \in P_i \not\Rightarrow x \in C_i$. On the other hand, the condition $C_i \not\subset P_j$ for all $i \neq j$ implies that not all points of C_i lie on the common boundary of cells P_i and P_j. If that was the case, C_i would be fully contained in the separating hyperplane, invalidating the interpretation of the power diagram as a classifier. Figure 3 depicts an example for a separating power diagram.

For weighted point sets, we add a somewhat technical extension of a separating power diagram. A feasible power diagram P is a **strongly feasible power diagram** if it has the following additional property: let $G(C)$ be the multigraph with vertices $C_1, \ldots, C_k$ and an edge labeled with x_j incident to C_i and C_l with $i \neq l$ if and only if $x_j \in \text{supp}(C_i) \cap \text{supp}(C_l)$. Then $G(C)$ does not contain a cycle with two or more different edge labels [36].

1.7 Weight-balanced least-squares assignments

The property of allowing a separating power diagram is tied to very special clusterings of point sets. It is well-known that so-called least-squares assignments allow the construction of Voronoi diagrams such that each cluster lies in its own cell. The existence of a separating power diagram corresponds to the clustering being a **weight-balanced least-squares assignment**. A clustering $\mathcal{C}$ is a weight-balanced $(S, |\mathcal{C}|)$-least-squares assignment of X if and only if $\sum\limits_{i=1}^{k} \sum\limits_{j=1}^{n} y_{ij} \|s_i - x_j\|^2$ is minimal for all clusterings of X of the same shape $|\mathcal{C}|$.

Let $X, S \subset \mathbb{R}^d$ and $|S| = k$. Then weight-balanced least-squares assignments are connected to power diagrams in the following way [6]:

1. Let $b \in \mathbb{R}^k$. Then there is an (S, b)-least-squares assignment $\mathcal{C}$ of X, and this $\mathcal{C}$ allows a separating S-power diagram.

2. If there is a separating S-power diagram P for a clustering $\mathcal{C}$ of X, then $\mathcal{C}$ is an $(S, |\mathcal{C}|)$-least-squares assignment of X.

The redistribution of lots in Figure 2 is an example for such a weight-balanced least-squares assignment. Note that when the sites S are given, the objective function

$$\min \sum_{i=1}^{k} \sum_{j=1}^{n} y_{ij} \cdot \|x_j - s_i\|^2$$

is linear. This means that the computation of a weight-balanced least-squares assignment for a given set of sites can be done by linear programming over the weight-balanced partition polytope.

1.8 Norm-maximal vertices of gravity polytopes

While there are educated apriori choices for the set of sites [22], the big question at hand is: which sites are an optimal choice? This question can be answered satisfactory by studying the so-called gravity polytopes [19, 36] and gravity bodies [36]. For the variant of weight-balanced clustering at hand, these will be polytopes.

Let $k, n, d \in \mathbb{N}$, $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$ and $a \in \mathbb{R}^k, b \in \mathbb{R}^n$ be the input for a weight-balanced clustering problem and let $\mathcal{C} = \{C_1, \dots, C_k\}$ be a feasible clustering. The **center of gravity** c_i of cluster C_i is given by $c_i = \frac{1}{|C_i|} \sum\limits_{j=1}^{n} y_{ij} x_j$. The **gravity vector** of $\mathcal{C}$ is then given by $c = (c_1^T, \dots, c_k^T)^T$, and the **gravity polytope** Q is defined by

$$Q = \operatorname{conv}\big\{c \in \mathbb{R}^{kd} : c \text{ is the gravity vector of a feasible clustering}\big\}.$$

The vertices of gravity polytopes correspond precisely to those clusterings that allow separating power diagrams [19], respectively strongly feasible power diagrams for weighted point sets [36]. (These results extend [6, 10].) This transforms our search for a best choice of sites to a search for a best vertex of the gravity polytope.

Here the vertices that are norm-maximal with respect to some ellipsoidal function exhibit favorable properties: their clusterings not only allow so-called centroidal power diagrams [36], where the sites correspond to the cluters' centers of gravity, but can also be characterized in terms of maximizing the so-called inter-cluster distance. However, due to the potentially exponential number of local maxima, convex maximization is in general $\mathbb{NP}$-hard. Hence it is necessary to resort to approximations.

1.9 Geometric clustering for land consolidation [25]

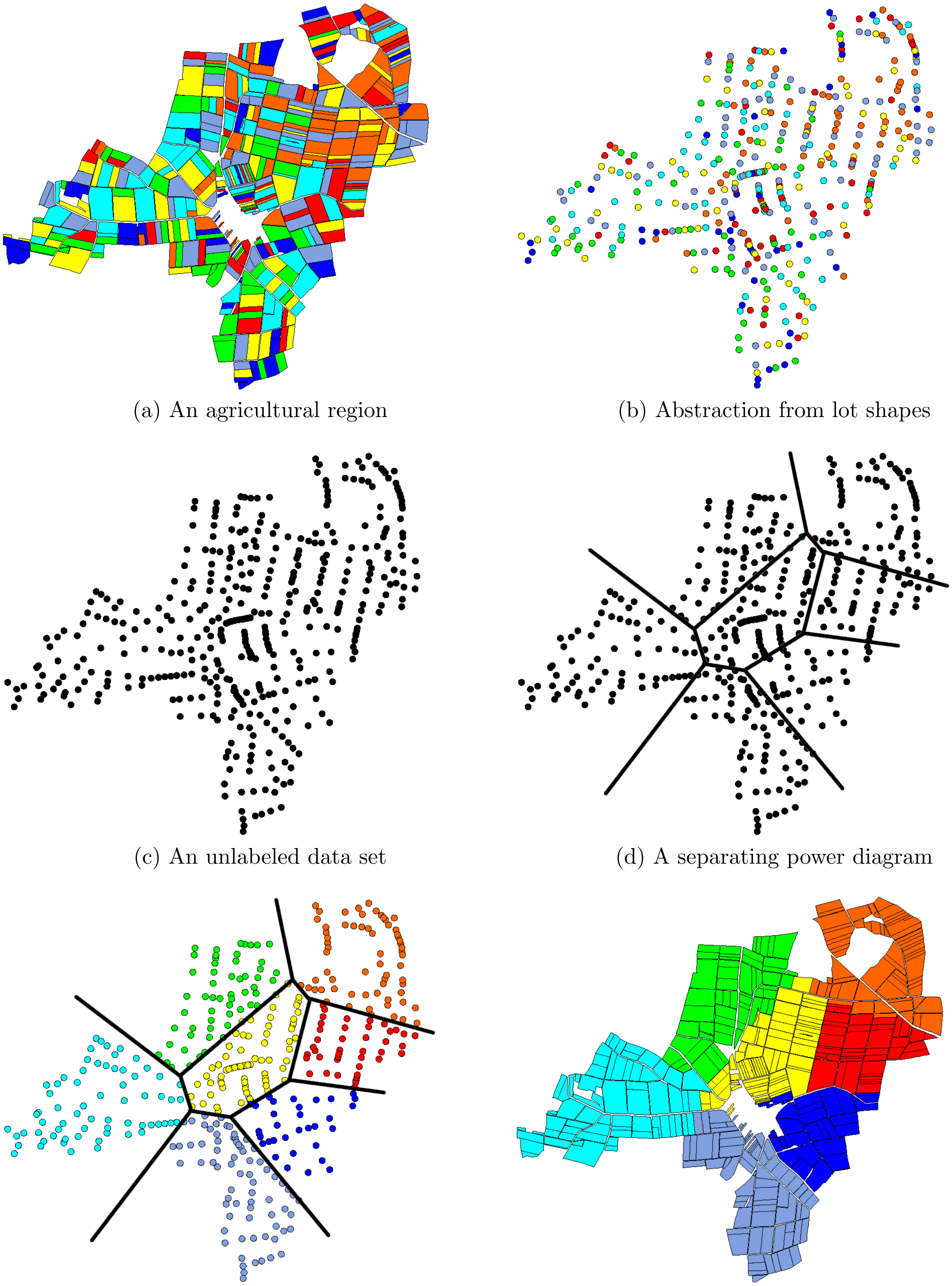
For this, the intuition behind the approach in [25] is to use an objective function that represents moving apart the centers of gravity of the clusters. Its construction involves a norm $\|\cdot\|$ on $\mathbb{R}^d$ and a second norm $\|\cdot\|_\diamond$ on $\mathbb{R}^{k(k-1)/2}$, where k is again the number of clusters. $\|\cdot\|_\diamond$ is required to be monotone, i.e., $\|x\|_\diamond \leq \|y\|_\diamond$ whenever $x, y \in \mathbb{R}^{k(k-1)/2}$ with $0 \leq x \leq y$. Then the objective function is of the form

$$\max \ \Big\| \big(\|c_1 - c_2\|, \|c_1 - c_3\|, \dots, \|c_{k-1} - c_k\|\big)^T \Big\|_\diamond .$$

Such an objective function leads to a nonlinear integer maximization problem (over a gravity polytope). However, its level sets are the **clustering bodies**, for which one can construct polyhedra (with only polynomially many facets in the dimension) that are approximations of low worst-case error [35]. This leads to a polynomial-time approximation algorithm by solving a linear program for each facet of the polyhedron and taking the maximum of the obtained values.

Note that a gravity polytope is a linear projection of a weight-balanced partition polytope. Essentially, linear optimization over a gravity polytope is the computation of a weight-balanced least-squares assignment, so a vertex of a gravity polytope and the corresponding clustering can be computed by linear optimization over a weight-balanced partition polytope.

We call this approach **geometric clustering**. Figure 4 sums up its application in land consolidation [25]. We abstract from the lot shapes and model the underlying problem



(a) An agricultural region

(b) Abstraction from lot shapes

(c) An unlabeled data set

(d) A separating power diagram

(e) Assignment of points to cell's cluster

(f) Lot redistribution

Figure 4: Geometric clustering for land consolidation.

as a weight-balanced clustering problem (Section 1.4). Here, we remember the values of lots and total value of each farmer. The original lot distribution is discarded, and we obtain a data set in the Euclidean plane. We then perform geometric clustering to obtain a separating power diagram for approximately optimal sites. Then it only remains to assign the lots of each cell to the corresponding farmer.

Naturally, the quality of the approximate solution depends greatly on the quality of the polyhedral approximation of the clustering body. The approach performs very well for land consolidation in agriculture in practice due to working in low dimension: the dimension $\mathbb{R}^{k(k-1)/2}$ of the polyhedron to approximate only depends on the low number k of farmers (typically $7-20$) – and not on the number n of lots (typically 500-2000).

For higher dimension, the concepts of separating power diagrams, weight-balanced least-squares assignments and norm-maximal vertices of gravity polytopes are just as powerful tools for structuring data, however one has to make the computations efficient in practice. We will turn to two ways of doing so: in the first one, motivated by an application in forestry, we reduce the number k and thus the dimension in a special way. In the second one, we resort to computing locally norm-maximal vertices of gravity polytopes, and in doing so generalize the k-means algorithm.

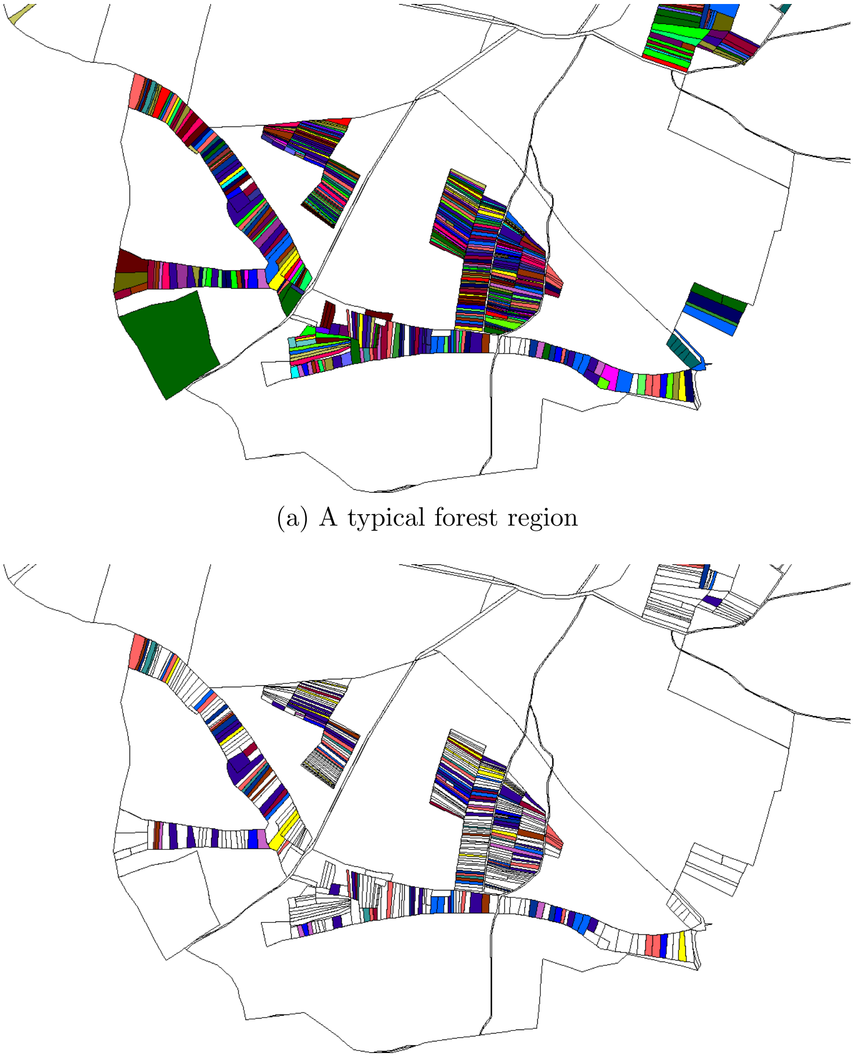
1.10 Key owners in forestry [31]

Like in agriculture, in many forest regions in Germany, an efficient and sustainable cultivation has become impossible due to inheritance regulations and frequent change of ownership. The average sizes of the lots have become less than a hectare. Further, the lots themselves often are of a bad shape, e.g. long but narror. Hence an approach for land consolidation can be similar to the one in agriculture. However, one faces additional challenges in practice related to the different time frame of production, the different number of owners and the different relation of the owners to their particular lots.

In an R&D-project of the Bavarian State Ministry for Agriculture, Nutrition and Forestry, I focused on a practical problem in that even in small regions (40-50 ha) there are often several hundred different owners. Figure 5 a) depicts an example. It consists of 460 lots that belong to 127 owners. In practice it is not realistic to enter negotiations with all stake holders.

Instead the goal is to identify a small group (5-20) of owners with a large 'potential' in a land exchange process. Thus, the forestry offices want to select some smaller subset of owners that are asked to participate. It is desirable to identify owners who provide sufficient and somehow best room for improving the cost structure in the region. Once this is done, a land exchhange process similiar to the one in agriculture can be initiated.

The key measure for this potential is the common boundary length in between adjacent lots of different owners (i.e. neighbours) [30, 31]. It is a measure for the improvement of the cost-effective structure is the creation of larger areas of lots that belong to the same owner during a redistribution process: by assigning adjacent lots to the same owner, one may be able to get rid of their common boundary. Further, large boundary lengths are often connected to unfavorable lot shapes — a very long lot that is only a couple of meters wide may have a small area, but nonetheless have a very large boundary to another lot owned by a different owner. By preferring owners with large total boundary lengths, one also implicitly improves on these bad lot shapes.



(a) A typical forest region

(b) Ten key owners with their 162 lots.

Figure 5: Selection of key owners in forestry.

It is possible to model the selection of a small group of owners with a maximal joint length of boundaries between each other as a graph-theoretical problem: each owner is a vertex, and vertices are connected by an edge if and only if the corresponding owners share a common boundary. The edges are weighted with the summed-up total length of all the joint boundaries of the corresponding pair of owners. Then the problem at hand is a version of the NP-hard dense k-subgraph problem with weighted vertices (or the computation of a weighted k-core):

Dense k-subgraph problem Graph $G = (V, E)$, weight function $w : E \to \mathbb{N}$, and $k \in \mathbb{N}$. Induced subgraph $G' = (V', E')$ of G such that $|V'| = k$ and $\sum\limits_{e \in E'} w(e)$ is maximal.

The dense k-subgraph problem is well-studied for its significance in many fields of operations research and data analysis [3, 81]. The problem is well-known to be $\mathbb{NP}$-hard in general, and there has been a lot of work on approximation algorithms [4, 14, 15, 38, 48, 49, 50, 53, 57, 58, 64, 68, 69, 73, 75, 89]. All of the approaches in the literature benefit significantly from smaller instances.

In [31], we develop a $O(k \cdot |V|^2)$-preprocessing routine which reduces a graph to a smaller subgraph that still contains a $(1-\frac{1}{k})$-approximation for the problem. For example, if we select $k = 20$ owners, our solution will be at most 5% from the optimal selection. The idea is to identify vertices that cannot contribute a lot to an optimal solution for the problem due to falling below the **threshold** of a sequence of vertices.

Let $\mathcal{S} = (v_1, \ldots, v_k)$ be a sequence of k vertices in a weighted graph $G = (V, E, w)$, and let $V_i = \{v_1, \ldots, v_i\}$. The threshold $\Delta = \Delta(\mathcal{S})$ of this sequence is

$$\Delta(\mathcal{S}) = \min_{i=1,\ldots,k-1} \max\{w(V_i) - w(V_{i-1}), \frac{1}{2}(w(V_{i+1}) - w(V_{i+1} \backslash \{v_i\}))\}.$$

The success of this preprocessing depends on a finding a large threshold, which ensures that many vertices can be removed. For this purpose, we devise an efficient algorithm which, given a graph, computes a sequence of k vertices with provably maximal threshold among all sequences of k vertices from the graph.

Using this information, we initiate a chain-reaction of vertex eliminations to reduce the number of vertices in the input graph without losing a lot of information. This yields an algorithm that iteratively deletes more and more vertices. By using arguments about submodular sets, one sees that the threshold is an upper bound on the edge weight lost in a (dense) k-subgraph that does not use any of the deleted vertices.

While one cannot guarantee the existence of vertices below such a threshold for general graphs, the special graphs of the forest application perform very favorably, typically allowing $90 - 95\%$ of all vertices to be removed from consideration. In view of the approach in [25], this means a reduction of the dimension of the clustering body by more than 99%.

1.11 Weight-balanced k-means [24]

In some data analysis tasks both the number of clusters k and the dimension d of the underlying data cannot be significantly reduced by preprocessing. The ability to work with such ‘big data’ is one of the challenges in machine learning. A simplification of the presented methods leads to a viable way of attacking such problems by introducing the **weight-balanced k-means** [24]. For the sake of simplicity, we here only present the algorithm for prescribed cluster sizes. It works for any lower and upper bounds on the cluster sizes (and even no bounds at all), and thus is a direct generalization of the well-known k-means algorithm [76].

The k-means algorithm begins with a set of n points and k different initial sites in $\mathbb{R}^d$, and then iteratively performs two steps: first, every point is assigned to a closest site. This partitions the points into k clusters, one for each site. Second, the sites are updated to be the arithmetic means of the clusters. These two-step iterations are performed until the sites do not change anymore. The algorithm exhibits many favorable properties, and is well-accepted for its simplicity and fast convergence in practice.

One can use the same idea by replacing the trivial assignment step of k-means by the computation of a weight-balanced least-squares assignments; see Algorithm 1. It computes a weight-balanced least-squares assignment in each iteration and terminates as soon as it finds one that allows a strongly feasible centroidal power diagram. This corresponds to the computation of a locally norm-maximal vertex in the gravity polytope (as opposed to a provably approximate global norm-maximal vertex as in [25]).

Algorithm 1 weight-balanced k-means

```
• Input: $d, k, n \in \mathbb{N}$, $X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, $a = (a_1, \ldots, a_k)$, $b = (b_1, \ldots, b_n)$,
  $S = \{s_1, \ldots, s_k\} \subset \mathbb{R}^d$
• Output: weight-balanced least-squares assignment of $X$ for its centers of gravity
  as sites
1. Compute a weight-balanced least-squares assignment $y$ for the current set of sites.
2. Update each site $s_i$ as the center of gravity of cluster $C_i$. If the objective function
   value decreased during the last iteration, go to (1.); else return the current
   assignment and sites.
```

It is easy to show termination of the standard k-means algorithm. The assignment of points to a closest site is readily interpreted as the computation of an unconstrained least-squares assignment. As $c_i = \frac{1}{|C_i|} \sum\limits_{j=1}^{n} y_{ij} x_j = \arg \min\limits_{c \in \mathbb{R}^d} \sum\limits_{j=1}^{n} y_{ij} \|x_j - c\|^2$, the objective function values of these least-squares assignments form a strictly decreasing sequence over the course of the algorithm. As there is only a finite number of different clusterings of (the unweighted) X, the algorithm terminates.

For weighted points and balanced, partial membership clustering, there is an infinite number of such clusterings, so that arguing with a decreasing sequence of objective function values does not suffice! However, as in each iteration we compute a different vertex of the weight-balanced partition polytope, we obtain termination.

Like k-means, this generalized version performs well in practice. It is kernelizable and can be sped up significantly by using the assignments in previous iterations for a warm start of later computations. Further, its worst-case number of iterations is in the same order $n^{O(dk)}$ as for k-means [65], so in a sense our extension is obtained at marginal additional cost and is polynomial for fixed k and d. This should be contrasted with the hardness of the k-means problem even in special cases [1, 77, 92].

More percisely, with e denoting the Euler number we obtain the bound $(40ek^2n)^{(d+1)k-1}$ on the number of iterations of our algorithm. Its proof is based on two steps. First, one bounds the number of different supports of strongly feasible power diagrams for weight-balanced clusterings. This can be done by estimating the number of different point-cell incidence structures, which we call **power patterns**, that can possibly be realized by power diagrams.

In order to provide an upper bound for the number of different power patterns, we use a well-known bound on the number of so-called sign-patterns of a set of polynomials by Warren [95] and a simple extension; see e.g. [2]:

Let $p_1, \ldots, p_t$ be a system of real polynomials in s variables. For a point $z \in \mathbb{R}^s$, the sign-pattern of $p_1, \ldots, p_t$ is a tuple $v(z) = (v_1, \ldots, v_t)^T \in \{+1, 0, -1\}^t$ defined by $v_i = -1$ if $p_i(z) < 0$, $v_i = 0$ if $p_i(z) = 0$ and $v_i = 1$ if $p_i(z) > 0$.

Let $p_1, \ldots, p_t$ be a system of real polynomials in s variables, all of degree at most l. If $s \leq 2t$, then the number of different sign-patterns of this system is bounded above by $\left(\frac{8e \cdot l \cdot t}{s}\right)^s$.

Herewith one can show that there are at most $\left(\frac{8e\cdot(k-1)n}{d}\right)^{(d+1)k-1}$ different power patterns. Refining the arguments one sees that the number of strict weight-balanced least-squares assigments and the number of iterations of Algorithm 1 is bounded by $(40ek^2n)^{(d+1)k-1}$.

1.12 Soft power diagrams [21]

Where [25] introduces an accessible method for clustering in machine learning, the underlying methods also have appeal for classification. Assume you have a clustering that allows a separating power diagram. The cells of the power diagram then are a classifier for new data points in the partitioned space. The quality of such a classifier is intuitively measured by the **margin**, the smallest Euclidean distance of a point of the data set to the boundary of its cell: the larger the margin, the better a classifier typically performs in practice. One of the goals then is to compute a separating power diagram of optimal margin for a given balanced least-squares assignment.

Further, a key necessity in classification is the ability to work with noisy data. This is typically by using **soft separation**, which has been studied well for binary classification tasks (i.e. $k = 2$). One uses penalty terms for misclassified points to bound and control the number of **support vectors** and **margin errors** that are allowed in the construction of a separating hyperplane for the two clusters [39, 85]: informally, support vectors are the points of the data set whose removal would change the optimal separating hyperplane, margin errors are the points which lie within a distance of at most the margin of the separating hyperplane or are on the wrong side of the separating hyperplane. Note that there is a direct tradeoff between the margin and the number of margin errors; the larger the margin, the more margin errors exist.

For $k > 2$, the situation is much more complicated; see e.g. [63]. This begins with different possible definitions of what margin errors are: a first approach is to measure the misclassification of a point with respect to each separating hyperplane [91, 97]. A second one is to only consider the 'worst' violation of a separating hyperplane by a point [40]. The associated altogether models for multiclass classification construct power diagrams that allow for soft-margin separation, yet they do not use a shared margin for the cluster

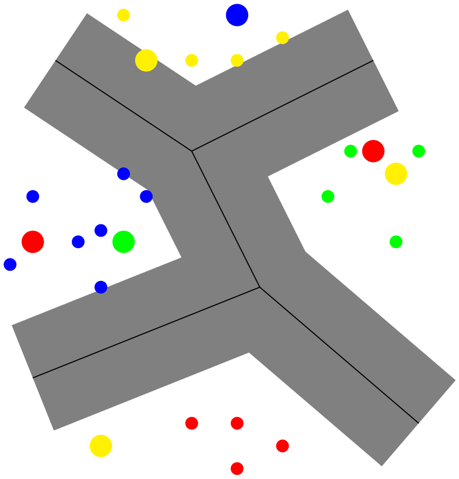


Figure 6: A soft power diagram.

pairs, but instead optimize the sum of squared pairwise margin sizes. This means that one loses the ability to compare quantitatively the misclassification of points with each other and loses control over the number of margin errors in the power diagram; however these are fundamental for applications such as outlier detection.

For this, we introduce the **soft power diagrams**, which transfer the concept of a parameter-controlled soft separation to power diagrams. Figure 6 shows an example. The width of the gray area around the hyperplanes of the diagram depicts the margin and the large dots are the so-called margin error points that are considered to be noisy due to not lieing in the interior of their cluster's cell, with at least the margin of distance to the boundary of the cell. Even though only few points are misclassified, a (non-soft) separating power diagram does not exist for these clusters.

Soft power diagrams are based on an alternate representation of power diagrams [21]: let $\gamma = (\gamma_1, \ldots, \gamma_k)$ with $\gamma_i \in \mathbb{R}$ for all $i \leq k$. An (S, γ)-power diagram is a decomposition $P = (P_1, \ldots, P_k)$ of $\mathbb{R}^d$ with

$$P_i = \{x \in \mathbb{R}^d : (s_j - s_i)^T x \leq \gamma_j - \gamma_i \quad \text{for all } j \neq i\}$$

if $\dim(P_i) = d$ for all $i \leq k$. For a given clustering and fixed sites, the search for a separating power diagram can be modelled by adding a slack variable ϵ and normalizing the vectors $s_j - s_i$ to $s_{ij} = \frac{(s_j - s_i)}{\|s_j - s_i\|}$. This normalization is necessary when optimizing over the sites, and makes the corresponding programs non-convex. One obtains the program

$$\begin{array}{llcll} (P_{\mathrm{SPD}}) & & \max \epsilon & & \\ & s_{ij}^T x_l + \epsilon & \leq & \gamma_{ij} & (\forall i, j, l : x_l \in C_i, j \neq i) \\ & (s_j - s_i)^T (c_j - c_i) & \geq & 1 & (\forall i, j : i < j) \end{array}$$

The second type of constraints rules out trivial solutions. It uses the fact the power diagrams are invariant under scaling of all the x_j and s_i by the same parameter.

For soft separation in the above model, we first have to define what **support vector points** and **margin error points** are: for a given clustering and power diagram, a $x \in C_i$ is a support vector point if and only if there is a $j \neq i$ such that $s_{ij}^T x + \epsilon \geq \gamma_{ij}$, and a margin error point if and only if there is a $j \neq i$ such that $s_{ij}^T x + \epsilon > \gamma_{ij}$.

Note that by relaxing the first type of constraints to $s_{ij}^T x_l + \epsilon \leq \gamma_{ij} + \xi_l$, one can always choose a sufficently large $\xi_l \geq 0$ to satisfy it. Herewith one is able to allow for misclassified points in the construction of a soft power diagram. By adding a special penalty term that depends on a parameter $t \in \mathbb{N}$ on the ξ_l one arrives at the system

$$(P_{\text{MEP}}) \qquad \begin{array}{rcll} & \max & \epsilon - \frac{t+\frac{1}{2}}{t(t+1)} \sum\limits_{l=1}^{n} \xi_l & \\ s_{ij}^T x_l + \epsilon & \leq & \gamma_{ij} + \xi_l & (\forall j, l : x_l \in C_i, j \neq i) \\ (s_j - s_i)^T (c_j - c_i) & \geq & 1 & (\forall i, j : i < j) \\ \xi_l & \geq & 0 & (\forall l). \end{array}$$

This system is non-convex and thus difficult to solve to global optimality. However, all of its local optima exhibit the desired properties [21]:

Let $\mathcal{C}$ be a clustering of X, and let $t \in \mathbb{N}$. Let $(S^*, \gamma^*, \epsilon^*, \xi^*)$ be a local optimum of (P_{MEP}). Then $(S^*, \gamma^*, \epsilon^*, \xi^*)$ yields a soft power diagram P of maximal margin ϵ^* for fixed S^*, γ^* such that t is an upper bound on the number of margin error points for P, and $t + 1$ is a lower bound on the number of support vector points for P.

The proof of this claim is based on the properties of optimal primal-dual solutions for (P_{MEP}). For fixed sites, the above program is linear. This gives rise to efficient algorithms for cluster outlier detection, where one now is able to prescribe the number of points one wants to label as outliers aprior.

2 Diameters of polyhedra

The Simplex method is the most common algorithm for solving linear programs. It can be viewed as a family of local search algorithms on the graph of a polyhedron, which consists of the zero- and one-dimensional faces of the feasible region (called vertices and edges). The search moves from a vertex of the graph to a better neighbouring vertex joined by an edge.

In the context of a best-case performance of the Simplex method, the study of the combinatorial diameter of polyhedra is a classical field in the theory of linear programming (recall, the combinatorial diameter of a polytope is the maximal length of a shortest edge walk between any pair of vertices). Despite great effort of analysis, it remains open whether there is always a polynomial bound on the combinatorial diameter. See for example [70] for a survey of the field.

Our interest in diameters comes from the reliance of our clustering methods on being able to efficiently solve transportation problems. Recall that the computation of a weight-balanced least-squares assignment is possible by linear programming over the corresponding weight-balanced partition polytope. Both the computation of approximate norm-maximal vertices [25] and of locally norm-maximal vertices [24] of a gravity polytope require the repeated solution of many linear programs over these polytopes. Recall further that (our variant of) weight-balanced partition polytopes are transportation polytopes.

Transportation problems are among the most fundamental problems in mathematical programming, operations research, and statistics [45, 44, 59, 60, 72, 99]. Originally, the **transportation problem** was posed by Hitchcock [59]:

Given a set of m suppliers and n demanders of a product, where the suppliers provide quantities $a_1, \dots, a_m \in \mathbb{R}$ and the demanders ask for quantities $b_1, \dots, b_n \in \mathbb{R}$, find an optimal distribution pattern from the suppliers to the demanders. By using variables y_{ij} representing the quantity sent from the i-th supplier to the j-th demander, we obtain the same formal representation of the set of solutions as before, which is called an $m \times n$-**transportation polytope** in this context.

$$\begin{array}{rcll} \sum\limits_{j=1}^{n} x_{ij} & = & a_i & (i \leq m) \\ \sum\limits_{i=1}^{m} x_{ij} & = & b_j & (j \leq n) \\ x_{ij} & \geq & 0 & (i \leq m, j \leq n). \end{array}$$

In the classical transportation problem, one optimizes a linear objective function over this polytope: if we have values c_{ij} representing the cost of shipping a unit of the product from the i-th supplier to the j-th demander, the task is to minimize $\sum\limits_{i=1}^{m} \sum\limits_{j=1}^{n} c_{ij} x_{ij}$. The vectors $a = (a_1, \dots, a_m), b = (b_1, \dots, b_n)$ are called the **margins** of the problem.

In 1957, W. Hirsch stated his famous conjecture (e.g. [41]) claiming that the diameter of a polytope is at most $f - d$, where d is its dimension and f its number of facets. As of today, there are counterexamples for both unbounded polyhedra [71] and (bounded) polytopes [83], but its validity for general $m \times n$–transportation polytopes is still open. The best bounds are linear [37].

On the other hand, the Hirsch conjecture is true for some related classes of polytopes. In particular, it holds for dual transportation polyhedra [9] and for $0, 1$-polytopes [80]. In the later sections, we will present proofs for additional special cases of transportation polytopes.

In an attempt to understand the behavior of the combinatorial diameter both in general and in view of transportation polytopes, we begin by introducing and investigating a hierarchy of distances and diameters for polyhedra that extend the usual edge walk [26, 28]. Instead of only walking along actual edges of the polyhedron, we walk along so-called **circuits** that allow us to enter the interior of the polyhedron, with different restrictions on these walks. We will see that this hierarchy includes the traditional combinatorial diameter.

Before turning to this hierarchy, let us remark that circuits have been useful in algorithms for linear optimization [16, 17, 18, 51, 82]. For a linear program, augmentation along circuit directions is a generalization of the Simplex method. While in the Simplex method one walks only along the graph (so in particular on the boundary) of the polyhedron, the circuit steps are allowed to go through the interior of the polyhedron (along *potential* edge directions). Such an approach reveals favorable properties for transportation polytopes, as they fall into the framework of N-fold linear programming [43].

2.1 A hierarchy of circuit diameters [28]

We consider polyhedra of the general form $P(\mathbf{b}, \mathbf{d}) = \{\mathbf{z} \in \mathbb{R}^n : A\mathbf{z} = \mathbf{b},\, B\mathbf{z} \leq \mathbf{d}\}$ for matrices $A \in \mathbb{Z}^{m_A \times n}$, $B \in \mathbb{Z}^{m_B \times n}$. The **circuits** or elementary vectors associated with matrices A and B are those vectors $\mathbf{g} \in \ker(A) \setminus \{\mathbf{0}\}$, for which $B\mathbf{g}$ is support-minimal in the set $\{B\mathbf{z} : \mathbf{z} \in \ker(A) \setminus \{\mathbf{0}\}\}$, where $\mathbf{g}$ is normalized to coprime integer components. Clearly, there are only finitely many such vectors. It is not difficult to show that the set of circuits consists exactly of all edge directions of $P(\mathbf{b}, \mathbf{d})$ when one lets $\mathbf{b}$ and $\mathbf{d}$ vary.

Let P be a polyhedron and let $\mathcal{C}$ be the set of circuits for the associated matrices A and B. For a pair of two vertices $\mathbf{v}^{(1)}, \mathbf{v}^{(2)}$ of P, we call a sequence $\mathbf{v}^{(1)} = \mathbf{y}^{(0)}, \ldots, \mathbf{y}^{(k)} = \mathbf{v}^{(2)}$ a **circuit walk** of length k if for all $i = 0, \ldots, k-1$ we have $\mathbf{y}^{(i+1)} - \mathbf{y}^{(i)} = \alpha_i \mathbf{g}^i$ for some circuit $\mathbf{g}^i$ and some $\alpha_i > 0$. The **circuit distance** from $\mathbf{v}^{(1)}$ to $\mathbf{v}^{(2)}$ is the minimum length of a circuit walk from $\mathbf{v}^{(1)}$ to $\mathbf{v}^{(2)}$. We call a circuit walk that realizes the circuit distance an optimal walk. The **circuit diameter** of P is the maximum circuit distance between any two vertices of P.

We consider different notions of circuit distances which arise by having circuit walks that satisfy additional properties:

(e) If $\mathbf{y}^{(i)}$ and $\mathbf{y}^{(i+1)}$ are neighbouring vertices in the graph of the polyhedron for all $i = 0, \ldots, k-1$, we have a classical *edge walk*.

(f) If $\mathbf{y}^{(i)} \in P$ for all $i = 0, \ldots, k-1$, then we say the circuit walk is *feasible*.

(m) If the extension multipliers α_i are maximal, i.e. if $\mathbf{y}^{(i)} + \alpha \mathbf{g}^i$ is infeasible for all $\alpha > \alpha_i$, we say that the walk is *maximal*.

(r) If no circuit is repeated, then we say the walk is *non-repetitive*.

(b) If no pair of circuits $\mathbf{g}^i, -\mathbf{g}^i$ is used, then we say the walk is *non-backwards*.

(s) If all the circuits are pairwise sign-compatible and are sign-compatible with the vector $\mathbf{v}^{(2)} - \mathbf{v}^{(1)}$, we say the walk is *sign-compatible*. (This is a somewhat technical definition, for details see [28].)

In what follows, we consider circuit distances restricted to different combinations of these properties and relate them to each other. Figure 7 depicts some walks for different combinations of these properties.

We use $\mathcal{CD}$ to refer to the circuit distance from $\mathbf{v}^{(1)}$ to $\mathbf{v}^{(2)}$ with no further restrictions. When considering only circuit walks on which we impose some of the above restrictions, we denote these restrictions by small subscript letters as used in the above list of properties. For example $\mathcal{CD}_{fs}$ refers to the feasible sign-compatible circuit distance, where the corresponding walk is feasible and sign-compatible, while $\mathcal{CD}_{fmr}$ means we have to use a feasible, maximal and non-repetitive walk.

Note that $\mathcal{CD}_{efm}$ is the classical *combinatorial distance*, while $\mathcal{CD}_{fm}$ corresponds to the *circuit distance* originally introduced in [26]. Further, we call $\mathcal{CD}_f$ the *weak circuit distance* and $\mathcal{CD}$ the *soft circuit distance*.

It is easy to see that many of these circuit distances bound each other, just by imposing additional constraints. For example the weak circuit distance $\mathcal{CD}_f$ is at least as large as the soft circuit distance $\mathcal{CD}$. We denote this $\mathcal{CD}_f \geq \mathcal{CD}$. If there are polyhedra with vertices such that these two values differ, we write $\mathcal{CD}_f > \mathcal{CD}$.

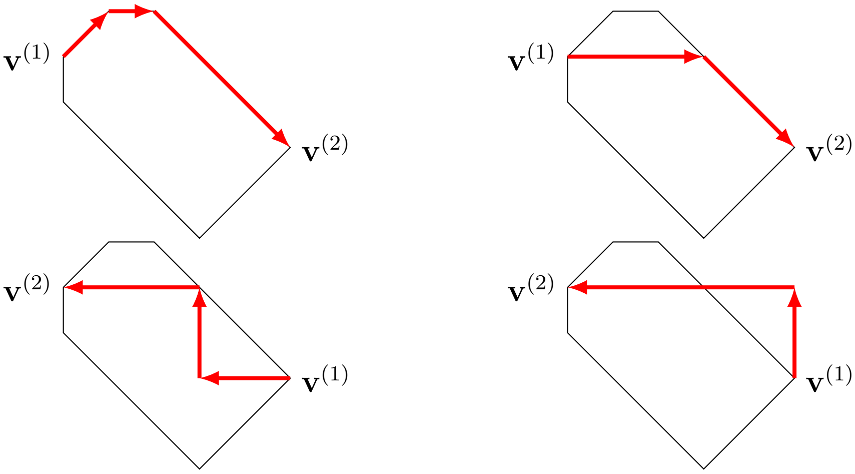


Figure 7: An edge walk and a feasible maximal walk (first row). A feasible (repetitive) walk and an unrestricted walk (second row).

As this notation is transitive – for example $\mathcal{CD}_{fm} \geq \mathcal{CD}_f \geq \mathcal{CD}$ implies $\mathcal{CD}_{fm} \geq \mathcal{CD}$ and $\mathcal{CD}_{fm} > \mathcal{CD}_f \geq \mathcal{CD}$ implies $\mathcal{CD}_{fm} > \mathcal{CD}$ – we obtain a hierarchy of circuit distances as depicted in Figure 8. Almost all relations in it are by strict inequalities, which is an indicator for a natural, viable categorization of the distances. We conjecture that the remaining inequalities are strict, as well.

For all pairs of strict inequalities, we present a corresponding polytope in [28]. In contrast, there are families of polyhedra for which many categories coincide. For simplicies or zonotopes, the whole hierarchy collapses. We also prove that in dimension $n = 2$, the hierarchy consists of only few distinct categories. For (two-dimensional) polygons, it is possible to explicity state the possible ranges of distances of vertices [28].

One of the benefits of considering the core chain of the hierarchy $\mathcal{CD}_{efm} > \mathcal{CD}_{fm} > \mathcal{CD}_f > \mathcal{CD}$ is that the distance concepts iteratively drop restrictions. This makes them tools for finding lower bounds on the combinatorial diameter; often they are much easier to bound. In fact, we recover validity of the Hirsch conjecture for large parts of the hierarchy:

Let $P = \{\, \mathbf{z} \in \mathbb{R}^n : A\mathbf{z} = \mathbf{b},\, B\mathbf{z} \leq \mathbf{d} \,\}$ with $A \in \mathbb{Z}^{m_A \times n}$, $B \in \mathbb{Z}^{m_B \times n}$ be a polyhedron in $\mathbb{R}^n$. For all pairs of vertices of P the distances $\mathcal{CD}_f$, $\mathcal{CD}_{fb}$, $\mathcal{CD}_{fr}$, $\mathcal{CD}_{fbr}$, and $\mathcal{CD}$ are bounded above by the distance $\mathcal{CD}_{fs}$. Moreover, all these distances are smaller or equal to $\min\{m_B - (n - \operatorname{rank}(A)), n - \operatorname{rank}(A)\}$.

Note that $n - \operatorname{rank}(A)$ is an upper bound on the affine dimension of P and $m_B - (n - \operatorname{rank}(A))$ is an upper bound on the number of facets minus the affine dimension, which implies the Hirsch bound. All circuit distances in the third layer and lower in Figure 8 satisfy the bound. In contrast, we know that $\mathcal{CD}_{efm}$ does not satisfy the Hirsch conjecture. This immediatly raises the question 'where' the bound is lost: this is either from $\mathcal{CD}_{efm}$ to $\mathcal{CD}_{fm}$, or from $\mathcal{CD}_{fm}$ to $\mathcal{CD}_f$:

Circuit diameter bound conjecture [26] For any d-dimensional polyhedron with f facets the circuit diameter $\mathcal{CD}_{fm}$ is bounded above by $f - d$.

It is an interesting open question whether the counterexamples to the Hirsch conjecture [71, 83] give rise to counterexamples to this conjecture or not.

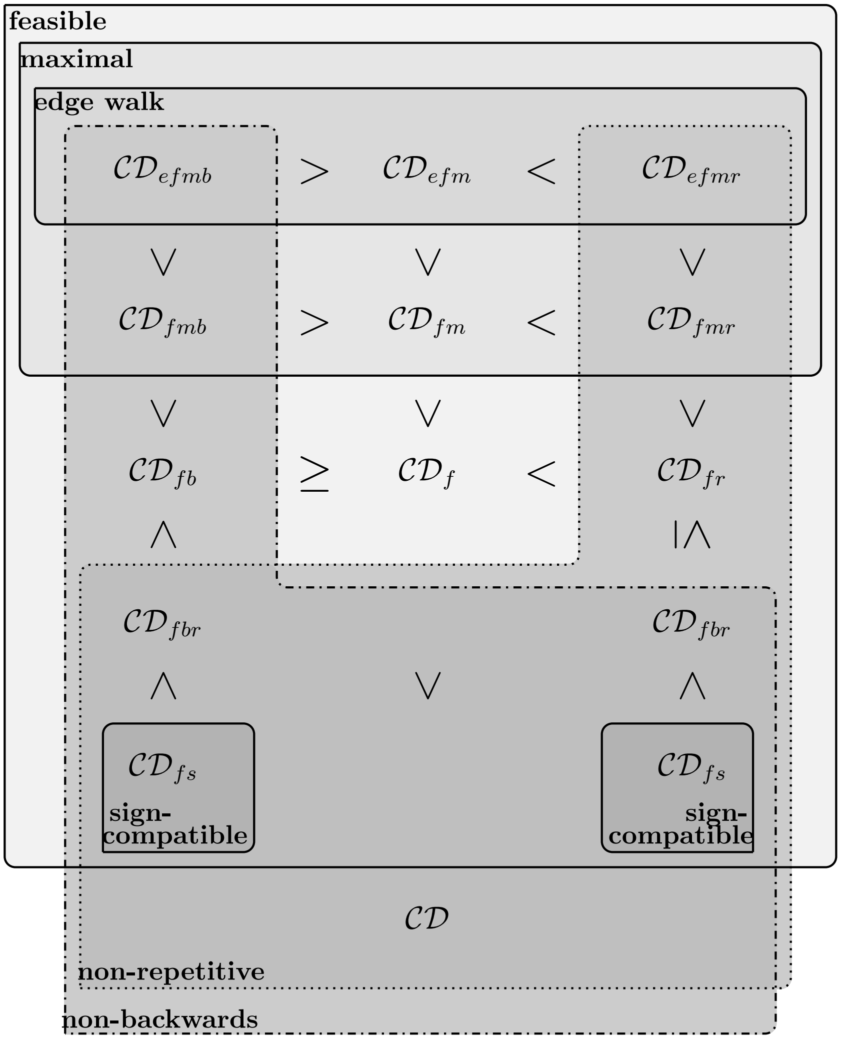

feasible
maximal
edge walk
$\mathcal{CD}_{efmb}$ $>$ $\mathcal{CD}_{efm}$ $<$ $\mathcal{CD}_{efmr}$
$\vee$ $\vee$ $\vee$
$\mathcal{CD}_{fmb}$ $>$ $\mathcal{CD}_{fm}$ $<$ $\mathcal{CD}_{fmr}$
$\vee$ $\vee$ $\vee$
$\mathcal{CD}_{fb}$ $\geq$ $\mathcal{CD}_{f}$ $<$ $\mathcal{CD}_{fr}$
$\wedge$ $\mid\wedge$
$\mathcal{CD}_{fbr}$ $\mathcal{CD}_{fbr}$
$\wedge$ $\vee$ $\wedge$
$\mathcal{CD}_{fs}$ $\mathcal{CD}_{fs}$
sign-compatible sign-compatible
$\mathcal{CD}$
non-repetitive
non-backwards

Figure 8: A hierarchy of circuit distances.

2.2 Circuits of transportation polytopes

Transportation polytopes exhibit a rich combinatorial structure. This leads to many specialized approaches and stronger results for bounds on their combinatorial diameters. For $m{\times}n$–transportation polytopes, validity of the Hirsch conjecture implies an upper bound of $m+n-1$ on the combinatorial diameter. The best upper bound in literature is $8(m+n-2)$, i.e. linear by a factor 8 [37]. Before we turn to our advances in diameters for transportation polytopes, let us recall some background on the graph-theoretical representation of their feasible solutions, and put it into perspective with respect to circuits.

It is common practice to think of the supply and demand points as nodes in the complete bipartite graph $K_{m,n}$. We denote the nodes corresponding to the supply points $\{\mathfrak{s}_1, \dots, \mathfrak{s}_m\}$ and the nodes corresponding to the demand points $\{\mathfrak{d}_1, \dots, \mathfrak{d}_n\}$. For every feasible solution $y \in \mathbb{R}^{m \times n}$ we define the **support graph** $B(y)$ as the subgraph of $K_{m,n}$ with edges $\{\{\mathfrak{s}_i, \mathfrak{d}_j\} \ : \ y_{ij} > 0, i \leq m, j \leq n\}$ of non-zero flow. We use the term **assignment** to refer to a feasible solution y and its support graph $B(y)$ at the same time, and use names such as O, C, F (for 'original', 'current', or 'final') for them. In this sense, assignments O can be vertices or lie in the interior of the polytope, we can count their number of edges by $|O|$, and so on.

When studying circuit distances, the vertices of the polytope are of special interest. They can be characterized by their support graphs: a feasible point y is a vertex if and only if its support graph contains no cycles, that is, $B(y)$ is a spanning forest. In particular the vertices y of non-degenerate transportation polytopes are given by spanning trees (see for example [72]). Then a vertex y is uniquely determined by (the edge set of) its support graph $B(y)$.

The edges of transportation polytopes are easy to characterize in terms of assignments [72]: two vertices O and C are connected by an edge if $O \cup C$ contains a unique cycle. This cycle describes an edge direction of the transportation polytope. Every cycle of $K_{m,n}$ can appear as an edge of some $m{\times}n$–transportation polytope for suitable vertices. Thus the set of circuits of an $m{\times}n$–transportation polytope consists of all simple cycles in $K_{m,n}$. Applying such a cycle $(\mathfrak{s}_{i_1}, \mathfrak{d}_{j_1}, \mathfrak{s}_{i_2}, \mathfrak{d}_{j_2}, \dots, \mathfrak{s}_{i_k}, \mathfrak{d}_{j_k}, \mathfrak{s}_{i_1})$ at a (feasible) point y changes the flow on the edges of $K_{m,n}$. We increase flow y_{i_l,j_l} on all edges $\{\mathfrak{s}_{i_l}, \mathfrak{d}_{j_l}\}$ and decrease flow $y_{i_l,j_{l+1}}$ on all edges $\{\mathfrak{d}_{i_l}, \mathfrak{s}_{j_{l+1}}\}$ by the same arbitrary amount. Informally, we *increase* or *decrease* edges; if the flow drops to zero, we *delete* an edge.

Observe that for $\mathcal{CD}_{efm}$ every circuit step inserts one edge and deletes one edge and hence the corresponding support graphs are always cycle free. In contrast to this, $\mathcal{CD}_{fm}$ can insert multiple edges while deleting at least one edge, such that there can be cycles. In circuit walks $\mathcal{CD}_f$ we can insert multiple edges and we do not have to delete an edge at all. Finally, infeasible points can appear in walks $\mathcal{CD}$. In this case, it is not well-defined to consider a support graph.

Clearly the combinatorial distance from an assignment O to an assignment F is at least $|O \backslash F|$: by applying a single pivot at an assignment O, one obtains an assignment C which shares at most one additional edge (the new, inserted one) with F. In contrast, the circuit distance $\mathcal{CD}_{fm}$ can be less than $|O \backslash F|$.

Figure 9 depicts one of the reasons why it is difficult to analyze the combinatorial diameter of transportation polytopes. The given example was first mentioned in [37]. It shows an edge walk from an assigment O to an assignment F. The nodes are labeled with the margins, the edges are labeled with the current flow, the bold edges highlight the circuit we apply, and the dashed edges are those we insert.

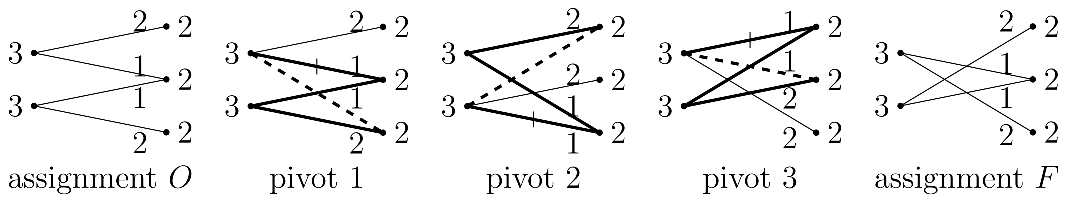


Figure 9: An edge walk from vertex O to vertex F of length three.

In the first step, no matter which edge in $F \backslash O$ we insert, we have to delete an edge that is contained in F. Hence this is a walk of minimum length and thus the combinatorial diameter is strictly larger than $|O \backslash F|$. In contrast, we can go from O to F in only one (feasible, maximal) circuit step that inserts and deletes two edges.

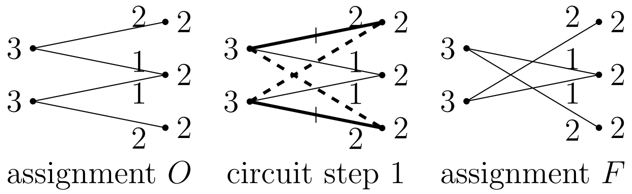


Figure 10: A feasible maximal circuit walk from vertex O to vertex F of length one.

The weak and the soft circuit diameter of transportation polytyopes are easier to analyze. We show that the Hirsch conjecture bound of $m+n-1$ is tight for these diameters in the sense that it is a general upper bound for all $m \times n$–transportation polytopes and that there exist margins for any m, n such that it is at least $\min\{m+n-1, (m-1)(n-1)\}$ [29]. The proof is based on bounding the length of sign-compatible walks. Of course, not all margins can realize the bounds [8, 20].

Knowing this, it remains interesting to study the original circuit diameter and the combinatorial diameter of transportation polytopes. In the next two sections, we present our results for some classes of transportation polytopes that appear frequently in practice. We then conclude with some results for dual transportation polyhedra.

2.3 $2 \times n$– and $3 \times n$–transportation polytopes [29]

First, we consider transportation problems with a low number of suppliers. For both $2 \times n$– and $3 \times n$–transportation polytopes, the set of circuits consists of only few distinct types of circuits. For example, note that all cycles in $K_{2,n}$ run through both supply nodes and are of length four; they are fully characterized by the included demand nodes. This allows us to refine the upper bound $n+1$ on the diameter of $2 \times n$–transportation polytopes [44] by one and prove that this bound is realized by a monotone path. Thus, $2 \times n$–transportation polytope satisfy the monotone Hirsch conjecture and are not Hirsch-sharp [61]. For the circuit diameter, we obtain an upper bound of $n-1$.

Further, we prove the Hirsch bound of $n+2$ for the combinatorial diameter of $3 \times n$–transportation polytopes. As a byproduct, this also is an upper bound on the circuit diameter. The key ingredient in the proof is a marking system in $K_{3,n}$. During the walk

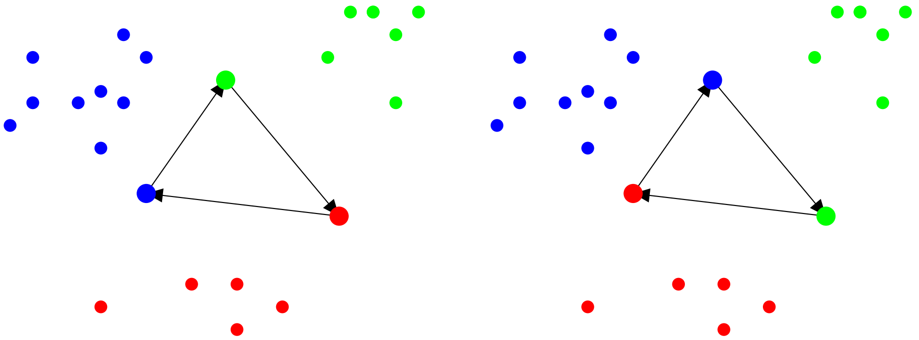


Figure 11: Pivoting in a partition polytope.

from an assignment O to an assignment F, we distinguish marked and unmarked edges in $K_{3,n}$ for the current assignment C. Unmarked edges may be deleted, marked edges must not be deleted.

For every marked edge $\{\mathfrak{s}_i, \mathfrak{d}_j\}$, it either is the only edge incident to $\mathfrak{d}_j$ in F or all (other) edges $\{\mathfrak{s}_i, \mathfrak{d}_l\}$ that are the single edges incident to the respective $\mathfrak{d}_l$ in F are already contained and marked in C. The technical, graph-theoretical proof exhibits that after at most one pivot we may always mark an edge in our current assignment, while keeping up this invariant: throughout the whole process we do not delete any marked edges. Thus, we will need at most $|F| \leq m + n - 1 = n + 2$ steps to arrive at the final assignment.

2.4 Partition polytopes [20]

In another important special case of transportation problems, the product is a set of discrete indistinguishable objects (e.g. teddys), and all customers ask for a single object. Such a problem also arises in the partitioning of an unweighted data set into clusters of prescribed number of points. Then $a \in \mathbb{N}^m$ and $b = (1, \ldots, 1)$, and we talk about a **partition polytope**. It is a $0, 1$–polytope due to its underlying constraint matrix being totally unimodular. Its vertices correspond to (hard) partitions of the underlying set. All circuits appear as edges in this polytope – pivoting from one vertex to a neighbouring one corresponds to applying a single circuit to exchange points as depicted in Figure 11.

As the partition polytopes are $0, 1$–polytopes, the Hirsch conjecture holds for them [80], however one can obtain three much stronger bounds on their combinatorial diameter, one of which only depends on the sizes of the two largest clusters:

Let $a = (a_1, \ldots, a_m)$ with $a_i \geq a_j$ for $i \geq j$. Then a partition polytope P has a diameter of at most $\min\{a_1 + a_2, n - a_1, \lfloor \frac{n}{2} \rfloor\}$ [20].

Informally, its diameter is bounded by the prescribed size of the largest cluster plus the size of the second-largest cluster. Note that this is a direct generalization of the diameter of the Birkhoff polytope being two (for $n \geq 4$) [8]. Here $m = n$ and $a = b = (1, \ldots, 1)$, so $a_1 + a_2 = 1 + 1 = 2$.

The bound $a_1 + a_2$ can be proved by a constructive, graph-theoretical approach. In the discussion, the following decision problem takes a key role.

Disjoint Cycle Subset Cover problem
Let $G = (V, E)$ be a digraph and let $M \subset V$.
Decide: Is there a set of (pairwise) (vertex-)disjoint cycles in E covering M?

The problem can solved in time $O(|V| + |E|\sqrt{|V|})$ [20]. More importantly for us, a certain class of graphs combined with a certain kind of vertex sets M always yields a yes-instances. In particular, the set M of vertices of maximal degree in a graph that decomposes into cycles yields a yes-instance. Such a graph, the **clustering difference graphs**, arises when describing the difference of a current assignment of points to another one. This allows us to develop an iterative algorithm to transfer one clustering into another.

An important part of this algorithm is that any set of vertex-disjoint cycles in a graph can be replaced as a 'sum' of at most two cycles. In algebraic terminology, this corresponds to the fact that any permutation is the product of at most two indecomposable permutations [8, 54]. In each iteration step, the vertices of maximal degree in the corresponding cluster difference graph have a cover by disjoint cycles, to which one can apply the above construction. This yields a sequence of such graphs of monotonely decreasing maximal degree until the clusterings are identical. A corresponding edge walk can be constructed in time $O(n(a_1 + a_2(\sqrt{m} - 1)))$.

We also give exact diameters for partition polytopes with $k = 2$ or $k = 3$ and exhibit a lower bound of $\lceil \frac{2}{3}a_2 \rceil + \lceil \frac{2}{3}a_4 \rceil$ on the combinatorial diameter.

2.5 Dual transportation polyhedra [26]

Using the Simplex method on a dual transportation polyhedron is another viable option for solving a transportation problem in practice. And just like the primal transportation polytopes, they exhibit a lot of combinatorial structure. For dual transportation polyhedra (on complete $m \times n$ bipartite graphs) the Hirsch bound $(m-1)(n-1)$ on the combinatorial diameter is proved and known to be tight [9]. In [26], we show the much stronger bound $m+n-2$ on the circuit diameter for all dual transportation polyhedra defined on (any, not necessarily complete) bipartite graphs with $m + n$ nodes. Thus these polyhedra are a family of examples whose circuit diameter is much smaller than their combinatorial diameter.

A graph-theoretical representation of these polyhedra is intimately related to the one of primal transportation polytopes. Let $G = (V, E)$ be a connected bipartite graph on node sets $V_1 = \{ 0, \dots, m-1 \}$ and $V_2 = \{ m, \dots, m+n-1 \}$ with edges E having one endpoint in V_1 and one endpoint in V_2. A dual transportation polyhedron associated to G is given by some vector $\mathbf{c} \in \mathbb{R}^{|E|}$ via

$$P_{G,\mathbf{c}} = \{ \mathbf{u} \in \mathbb{R}^{m+n} : -u_a + u_b \leq c_{ab} \ \forall \ a \in V_1, b \in V_2 \text{ and } ab \in E, u_0 = 0 \}.$$

One puts $u_0 = 0$ to make $P_{G,\mathbf{c}}$ pointed. The vertices of $P_{G,\mathbf{c}}$ are determined by sets of inequalities $-u_a + u_b \leq c_{ab}$ that become tight. For $\mathbf{u} \in P_{G,\mathbf{c}}$, we denote by $G(\mathbf{u})$ the graph with nodes V and with edges $ab \in E$ for which $-u_a + u_b \leq c_{ab}$ is tight. For a vertex $\mathbf{u}$ of $P_{G,\mathbf{c}}$, $G(\mathbf{u})$ is a spanning subgraph of G which is a spanning tree of G if $P_{G,\mathbf{c}}$ is generic.

The possible edge directions of $P_{G,\mathbf{c}}$ can be described as follows: let $R, S \subseteq V$ be node sets such that the underlying undirected subgraphs of the respective node sets are connected and satisfy $R \cup S = V$ and $R \cap S = \emptyset$. We may assume $0 \in R$. Then the vector $\mathbf{g} \in \mathbb{R}^{M+N}$ with $g_i = 0$ if $i \in R$ and $g_i = 1$ if $i \in S$ is an edge direction of $P_{G,\mathbf{c}}$ for some right-hand side $\mathbf{c}$ [9]. In fact, it can be shown that these are all possible edge directions and hence they constitute the set of circuits associated to the matrix defining the polyhedron $P_{G,\mathbf{c}}$.

Note that for each vertex of $P_{G,\mathbf{c}}$ there is a spanning tree of G with edges corresponding to the inequalities $-u_a + u_b \leq c_{ab}$ that are tight at the vertex. This uniquely determines the vertex $\mathbf{u}$, since we normalized $u_0 = 0$. This allows us to characterize vertices by spanning trees, and leads to an upper bound of $|V| - 2$ on the circuit diameter of $P_{G,\mathbf{c}}$:

Let $\mathbf{u}^{(1)}$ and $\mathbf{u}^{(2)}$ be two vertices of $P_{G,\mathbf{c}}$ given by the spanning trees $T_1 = G(\mathbf{u}^{(1)})$ and $T_2 = G(\mathbf{u}^{(2)})$ of G. The core part of the proof is to construct a circuit walk $\mathbf{u}^{(1)} = \mathbf{y}^{(0)}, \dots, \mathbf{y}^{(k)} = \mathbf{u}^{(2)}$, such that $G(\mathbf{y}^{(i)})$ has at least i edges in common with T_2. This immediately implies $k \leq |V| - 1$, and one obtains a a bound of $|V| - 2$ by noting that T_1 and T_2 have to overlap in at least one edge.

Acknowledgement

My first thanks go to Prof. Peter Gritzmann, who is an outstanding mentor and gave me the opportunity of completing this habilitation thesis in the first place. I would like to thank Prof. Jesús De Loera and Prof. Andreas Brieden for the continuous support and brilliant cooperation. My thanks also go to Prof. Shmuel Onn for the ongoing support, to Prof. Michael Ulbrich for being part of my habilitation committee, and to Prof. Uwe Zimmermann for the opportunity of acting as a professor in his group as part of this habilitation.

My special thanks go to Dr. Stefan Schaffner and Markus Lechner, who have done a remarkable job in running the communication for project ArborTec. I would also like to thank all of my cooperation partners and co-authors: Elisabeth Finhold, Prof. Raymond Hemmecke, Jake Miller, Felix Schmiedl, and everyone I worked with outside of this thesis.

I am very grateful for having enjoyed a productive and friendly atmosphere in all of my working places. I here would like to specifically thank Rene Brandenberg, Elisabeth Finhold, Viviana Ghiglione and Michael Ritter. Finally, my heartfelt thanks go to my friends and family for their love and support.

References

[1] D. Aloise, A. Deshpande, P. Hansen, and P. Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75:245–249, 2009.

[2] N. Alon. Tools from higher algebra. In R. L. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 2, pages 1749–1783. MIT Press, 1995.

[3] R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In K. Avrachenkov, D. Donato, and N. Litvak, editors, *Algorithms and Models for the Web-Graph*, volume 5427 of *Lecture Notes in Computer Science*, pages 25–37. Springer Berlin, 2009.

[4] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[5] F. Aurenhammer. Power diagrams: Properties, algorithms and applications. *SIAM Journal on Computing*, 16(1):78–96, 1987.

[6] F. Aurenhammer, F. Hoffmann, and B. Aronov. Minkowski-type theorems and least-squares clustering. *Algorithmica*, 20:61–76, 1998.

[7] F. Aurenhammer and R. Klein. Voronoi diagrams. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 201–290. Elsevier Science, 1999.

[8] M. Balinski and A. Russakoff. On the assignment polytope. *SIAM Review*, 16(4), 1974.

[9] M. L. Balinski. The Hirsch conjecture for dual transportation polyhedra. *Mathematics of Operations Research*, 9(4):629–633, 1984.

[10] E. R. Barnes, A. J. Hoffman, and U. G. Rothblum. Optimal partitions having disjoint convex and conic hulls. *Mathematical Programming*, 54(1):69–86, 1992.

[11] S. Basu, I. Davidson, and K. L. Wagstaff. *Clustering with Constraints: Advances in Algorithms, Theory and Applications.* Chapman & Hall, 2009.

[12] K. P. Bennett and O. L. Mangasarian. Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3:27–39, 1992.

[13] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, 2002.

[14] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 201–210. ACM, 2010.

[15] A. Billionnet, S. Elloumi, and M.-C. Plateau. Improving the performance of standard solvers for quadratic 0-1 programs by a tight convex reformulation: The QCR method. *Discrete Applied Mathematics*, 157(6):1185–1197, 2009.

[16] R. G. Bland. *Complementary orthogonal subspaces of n-dimensional Euclidean space and orientability of matroids.* 1974. PhD Thesis.

[17] R. G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977.

[18] R. G. Bland and D. L. Jensen. On the computational behavior of a polynomial-time network flow algorithm. *Mathematical Programming*, 54(1):1–39, 1992.

[19] S. Borgwardt. *A Combinatorial Optimization Approach to Constrained Clustering.* 2010. PhD Thesis.

[20] S. Borgwardt. On the diameter of partition polytopes and vertex-disjoint cycle cover. *Mathematical Programming, Ser. A*, 141(1):1–20, 2013.

[21] S. Borgwardt. On Soft Power Diagrams. *Mathematical Modelling and Algorithms in Operations Research*, 14(2):173–196, 2015.

[22] S. Borgwardt, A. Brieden, and P. Gritzmann. Constrained minimum-k-star clustering and its application to the consolidation of farmland. *Operational Research*, 11(1):1–17, 2011.

[23] S. Borgwardt, A. Brieden, and P. Gritzmann. Mathematics in agriculture and forestry: Geometric clustering for land consolidation. *IFORS news*, Dec. issue, 2013.

[24] S. Borgwardt, A. Brieden, and P. Gritzmann. A balanced k-means algorithm for weighted point sets. eprint arXiv:1308.4004, 2014.

[25] S. Borgwardt, A. Brieden, and P. Gritzmann. Geometric clustering for the consolidation of farmland and woodland. *The Mathematical Intelligencer*, 36(2):37–44, 2014.

[26] S. Borgwardt, E. Finhold, and R. Hemmecke. On the circuit diameter of dual transportation polyhedra. *Siam Journal on Discrete Mathematics*, 29(1):113–121, 2014.

[27] S. Borgwardt, E. Finhold, and R. Hemmecke. Quadratic diameter bounds for dual network flow polyhedra. eprint arXiv:1408.4184, 2014.

[28] S. Borgwardt, J. De Loera, and E. Finhold. Edges vs circuits: a hierarchy of diameters in polyhedra. eprint arXiv:1409.7638, 2014.

[29] S. Borgwardt, J. De Loera, E. Finhold, and J. Miller. The Hierarchy of Circuit Diameters and Transportation Polytopes. eprint arXiv:1411.1701, 2014.

[30] S. Borgwardt, S. Schaffner, and M. Suda. Geometric measures for the assessment of fragmentation of private forest areas - Geometrische Kennzahlen für die forstfachliche Bewertung der Zersplitterung von Privatwaldarealen. *Forstarchiv*, 06-14, 2014.

[31] S. Borgwardt and F. Schmiedl. Threshold-based preprocessing for approximating the weighted dense k-subgraph problem. *European Journal of Operational Research*, 234:631–640, 2014.

[32] E. J. Bredensteiner and K. P. Bennett. Multicategory classification by support vector machines. *Computational Optimizations and Applications*, 12:53–79, 1999.

[33] A. Brieden. *On the approximability of (discrete) convex maximization and its contribution to the consolidation of farmland.* 2003. Habilitationsschrift.

[34] A. Brieden and P. Gritzmann. A quadratic optimization model for the consolidation of farmland by means of lend-lease agreements. In *Operations Research Proceedings 2003: Selected Papers of the International Conference on Operations Research*, pages 324–331, 2004.

[35] A. Brieden and P. Gritzmann. On clustering bodies: Geometry and polyhedral approximation. *Discrete Computational Geometry*, 44(3):508–534, 2010.

[36] A. Brieden and P. Gritzmann. On optimal weighted balanced clusterings: Gravity bodies and power diagrams. *SIAM Journal on Discrete Mathematics*, 26:415–434, 2012.

[37] G. Brightwell, J. Heuvel, and L. Stougie. A linear bound on the diameter of the transportation polytope. *Combinatorica*, 26(2):133–139, 2006.

[38] D. G. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, 1984.

[39] C. Cortes and V. Vapnik. Support-vector networks. In *Machine Learning*, volume 20, pages 273–297, 1995.

[40] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal on Machine Learning Research*, 2:265–292, 2001.

[41] G. Dantzig. *Linear Programming and Extensions.* Princeton Univ. Press, 1963.

[42] J. A. De Loera, R. Hemmecke, and M. Köppe. *Algebraic and geometric ideas in the theory of discrete optimization*, volume 14 of *MOS-SIAM Series on Optimization.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 2013.

[43] J. A. De Loera, R. Hemmecke, and J. Lee. Augmentation in Linear and Integer Linear Programming: From Edmonds-Karp to Bland and Beyond. eprint arXiv:1408.3518, 2014.

[44] J. A. De Loera and E. D. Kim. Combinatorics and geometry of transportation polytopes: An update. *eprint arXiv:1307.0124*, 2013.

[45] J. A. De Loera, E. D. Kim, S. Onn, and F. Santos. Graphs of transportation polytopes. *Journal of Combinatorial Theory, Ser. A*, 116(8):1306–1325, 2009.

[46] U. Elsner. Graph partitioning - a survey. Technical Report 97-27, Tech. Univ. Chemnitz, 1997.

[47] B. Everitt, S. Landau, and M. Leese. *Cluster Analysis.* John Wiley & Sons, 2001.

[48] U. Feige and M. Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *Journal of Algorithms*, 41(2):174–211, 2001.

[49] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[50] U. Feige and M. Seltser. On the densest k-subgraph problems. Technical Report CS97-16, Weizmann Institute, Rehovot, Israel, 1997.

[51] K. Fukuda and T. Terlaky. Criss-cross methods: A fresh view on pivot algorithms. *Mathematical Programming, Ser. B*, 79(1-3):369–395, 1997.

[52] M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

[53] A. V. Goldberg. Finding a maximum density subgraph. Technical Report UCB/CSD-84-171, EECS Department, University of California, Berkeley, 1984.

[54] R. Guralnick and D. Perkinson. Permutation polytopes and indecomposable elements in permutation groups. *Journal of Combinatorial Theory, Ser. A*, 113:1243–1256, 2006.

[55] N. Guttmann-Beck and R. Hassin. Approximation algorithms for min-sum p-clustering. *Discrete Applied Mathematics*, 89:125–142, 1998.

[56] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[57] R. Hassin, S. Rubinstein, and A. Tamir. Approximation algorithms for maximum dispersion. *Operations Research Letters*, 21(3):133–137, 1997.

[58] R. Hemmecke. On the computation of Hilbert bases of cones. In A. M. Cohen, X. S. Gao, and N. Takayama, editors, *Mathematical Software, ICMS 2002.* World Scientific, 2002.

[59] F. K. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal on Mathematical Physics*, 20:224–230, 1941.

[60] A. J. Hoffman. On simple linear programming problems. In *Proceedings of Symposia in Pure Mathematics*, volume VII, pages 317–327. American Mathematical Society, Providence, RI, 1963.

[61] F. B. Holt and V. Klee. Many polytopes meeting the conjectured Hirsch bound. *Discrete and Computational Geometry*, 20:1–17, 1998.

[62] F. Höppner and F. Klawonn. Clustering with size constraints. *Computational Intelligence Paradigms, Innovative Applications*, 2008.

[63] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.

[64] M. Hunting, U. Faigle, and W. Kern. A Lagrangian relaxation approach to the edge-weighted clique problem. *European Journal of Operational Research*, 131(1):119–131, 2001.

[65] M. Inaba, N. Katoh, and H. Imai. Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. In *Proceedings of the tenth Annual Symposium on Computational Geometry*, SCG '94, pages 332–339, 1994.

[66] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[67] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering - a review. In *ACM Computing Surveys*, volume 31-3, pages 264–323, 1999.

[68] J. M. Keil and T. B. Brecht. The complexity of clustering in planar graphs. *The Journal of Combinatorial Mathematics and Combinatorial Computing*, 9:155–159, 1991.

[69] S. Khot. Ruling out PTAS for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.

[70] E. D. Kim and F. Santos. An update on the Hirsch conjecture. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 112(2):73–98, 2010.

[71] B. King. Step-wise clustering procedures. In *J. Amer. Stat. Assoc.*, volume 69, pages 86–101, 1967.

[72] V. Klee and C. Witzgall. Facets and vertices of transportation polyhedra. In G. B. Dantzig and A. F. Veinott, editors, *Mathematics of the decision sciences*, volume 1, pages 257–282. American Mathematical Society, Providence, RI, 1968.

[73] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In *Proceedings of the 1993 IEEE 34th Annual Foundations of Computer Science*, SFCS '93, pages 692–701, Washington, DC, 1993. IEEE Computer Society.

[74] S. B. Kotsiantis and P. E. Pintelas. Recent advances in clustering: A brief survey. In *WSEAS Transactions on Information Science and Applications*, volume 1-1, pages 73–81, 2004.

[75] E. M. Macambira and C. C. de Souza. The edge-weighted clique problem: Valid inequalities, facets and polyhedral computations. *European Journal of Operational Research*, 123(2):346–371, 2000.

[76] J. B. MacQueen. Some methods of classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[77] M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is NP-hard. *Machine Learning*, 75:245–249, 2009.

[78] D. Man and M. Tosa-Abrudan. Fuzzy clustering algorithms: A survey. In 7*th Joint Conference on Mathematics and Computer Science*, 2008.

[79] B. Mirkin. *Clustering for data mining: A data recovery approach.* Chapman & Hall, 2005.

[80] D. Naddef. The Hirsch Conjecture is true for $(0,1)$-polytopes. *Mathematical Programming*, 45(1):109–110, 1989.

[81] D. Pisinger. Upper bounds and exact algorithms for p-dispersion problems. *Computers & Operations Research*, 33(5):1380–1398, 2006.

[82] R. T. Rockafellar. The elementary vectors of a subspace of R^N. In *Combinatorial Mathematics and its Applications*, pages 104–127. University of North Carolina Press, Chapel Hill, NC, 1969.

[83] F. Santos. A counterexample to the Hirsch conjecture. *Annals of Mathematics*, 176(1):383–412, 2011.

[84] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[85] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neuronal Computation*, 12:1207–1245, 2000.

[86] J. Scoltock. A survey of the literature of cluster analysis. *The Computer Journal*, 25(1):130–134, 1982.

[87] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[88] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, 1973.

[89] M. M. Sörensen. New facets and a branch-and-cut algorithm for the weighted clique problem. *European Journal of Operational Research*, 154(1):57–70, 2004.

[90] H. M. M. ten Eikelder and A. A. van Erk. Unification of some least squares clustering methods. *J. Mathematical Modelling and Algorithms*, 3:105–122, 2004.

[91] V. Vapnik. *Statistical learning theory*. Wiley, 1998.

[92] A. Vattani. k-Means requires exponentially many iterations even in the plane. *Discrete Computational Geometry*, 45:596–616, 2011.

[93] J.-P. Vert, K. Tsuda, and B. Schölkopf. A primer on kernel methods. In *Kernel Methods in Computational Biology*, pages 35–70. MIT Press, 2004.

[94] J. H. Ward. Hierarchical grouping to optimize an objective function. In *Journal of the American Statistical Association*, volume 58, pages 236–244, 1963.

[95] H. E. Warren. Lower bounds for approximation by nonlinear manifolds. *Transactions of the American Mathematical Society*, 133(1):167–178, 1968.

[96] I. Wasito and B. Mirkin. Nearest neighbour approach in the least-squares data imputation algorithms. *Information Sciences*, 169:1–25, 2004.

[97] J. Weston and C. Watkins. Multi-class support vector machines. Technical report, University of London, 1998.

[98] R. Xu and D. Wunsch. Survey of clustering algorithms. In *IEEE Transactions on Neural Networks*, volume 16:3, pages 645–678, 2005.

[99] V. A. Yemelichev, M. M. Kovalëv, and M. K. Kravtsov. *Polytopes, graphs and optimisation*. Cambridge University Press, Cambridge, 1984. Translated from the Russian by G. H. Lawden.

[100] S. Zhong and J. Ghosh. Scalable, balanced model-based clustering. In *SIAM International Conference on Data Mining*, volume SDM-03, pages 71–82, 2003.