Convex Group Clustering of Large Geo-referenced Data Sets

Vladimir Estivill-Castro
Department of Computer Science & Software Engineering,
The University of Newcastle,
Callaghan, NSW 2308, Australia.
vlad@cs.newcastle.edu.au

Abstract

Clustering partitions a data set $S = \{s_1, \ldots, s_n\} \subset \Re^m$ into groups of nearby points. Distance-based clustering methods use optimisation criteria to define the quality of a partition. Formulations using representatives (means or medians of groups) have received much more attention than minimisation of the *total within group distance* (TWGD). However, this non-representative approach has attractive properties while remaining distance-based.

While representative approaches produce partitions with non-overlapping clusters, TWGD does not. We investigate the restriction of TWGD to producing convex-hull disjoint groups and show that this problem is NP-complete in the Euclidean case as soon as $m \geq 2$. Nevertheless we provide efficient algorithms for solving it approximately.
KEYWORDS: clustering, optimisation, computational geometry, problem complexity, data mining in spatial databases.

1 Introduction

Clustering is a fundamental task in data analysis since it identifies groups in heterogeneous data. Clustering can be seen as a concept formation or class delineation problem. At least the fields of statistics [44, 46], machine intelligence [5, 15, 32] and more recently knowledge discovery and data mining (KDDM) [12, 14, 37, 47] have contributed with algorithms for many clustering approaches. Hierarchical bottom-up approaches form groups by composition or merging items that are close together [10, 29]. However, top-down partition sees clustering as partitioning a heterogeneous data set into smaller more homogeneous groups [2, 19, 40] and is of particular interest for spatial data mining [12, 37, 48].

Clustering typically uses a metric (or distance) to determine the similarity between the items to be clustered. Here we consider the clustering problem in the context of spatial databases, those typically associated with a Geographical Information System (GIS). In spatial settings, the clustering almost invariably makes use of some distance that captures the notion of proximity, as it reflects the essence of spatial association. We say that the clustering problem is *distance based* when a metric is used to formulate an optimisation criterion that describes clustering as an optimisation problem. For example, the criterion may consist of minimising the total dissimilarity in the groups. This criterion has been called *Grouping* [45], *Total Within Group Distance* [40], the *Full-Exchange* [42] and the *Interaction* [35]. Here we will refer to this criterion as the *Total Within Group Distance* (TWGD), since this seems the best description of the measure.

Definition 1.1 *Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects and let $d : X \times X \rightarrow \Re^+ \cup \{0\}$ be a metric. Clustering into p groups by the Total Within Group Distance (*TWGD*) consists of solving the following distance-based clustering optimisation.*

$$\textit{Minimise} \quad \mathrm{TWGD}(P) = \sum_{k=1}^{p} \sum_{i<j \ \wedge \ x_i, x_j \in X_k} w_i w_j d(x_i, x_j)\}, \tag{1}$$

where $P = X_1 | \ldots | X_p$ is a partition of X and w_i is a weight for the relevance of x_i but may have other specific interpretations.

Intuitively, this criterion not only minimises the dissimilarity between items in a group, but also uses all interactions between items in a group to assess group cohesiveness (and thus, uses all the available information). Also, implicitly, it maximises the distance between groups, because the terms $d(x_i, x_j)$ not included in the sum are those where the items belong to different groups. Therefore, it minimises coupling.

However, the TWGD problem is NP-complete (in its graph-theoretic decision formulation [4] and in the Euclidean formulation [27]). Although most other distance-based clustering problems are also NP-complete [4, 22, 28], they are more commonly solved approximately in clustering applications. We believe this is due to several factors.

1. The TWGD problem, when formulated as an integer programming problem, is very difficult to solve optimally by approaches like integer-relaxation and branch and bound [42]. The most popular formulation as a integer-programming problem has $n^2 p$ variables and $n + p$ constraints [33, 35, 42]. Reformulations of the problem may not work well with the relaxation [42] or may rapidly increase the number of constraints. Therefore, only instances were n is small can be attacked with this approach.

2. For each group X_k, $(k = 1, \ldots, p)$, the objective function involves $\Theta(n_k^2)$ terms (where $n_k = |X_k|$, is the

Full version for CCCG-99

size of group X_k). Thus, approximation methods like hill-climbing, simulated annealing, genetic algorithms and so on, face costly function evaluations.

3. Fast approximation algorithms are available for other distance-based clustering approaches. With the emergence of KDDM, where very large data sets are analysed, the computational complexity of the clustering algorithms is certainly crucial.

Because of this last point, the K-MEANS algorithm (of basic isodata) [10] is used extensively in KDDM. K-MEANS is an heuristic to approximately solve the following distance-based clustering optimisation.

Minimise

$$\text{EUCLID}^2(P) = \sum_{k=1}^{p} \sum_{i<j \ \wedge \ \vec{x}_i,\vec{x}_j \in X_k} \frac{w_i w_j d_E(\vec{x}_i,\vec{x}_j)^2}{W_k}, \quad (2)$$

where
(a) $P = X_1|\ldots|X_k$ is a partition of $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\} \subset \Re^m$ (i.e. X is a set of n m-dimensional data points),
(b) the weight w_u may reflect relevance of $\vec{x}_u$, and the distance $d_E(\vec{x}_u, \vec{x}_v)$ is the Euclidean distance (i.e. $d_{Eu}(\vec{x},\vec{y}) = (\sum_{j=1}^{m} |x_j - y_j|^2)^{1/2}$), and
(c) $W_k = \sum_{x_i \in X_k} w_i = \sum_{X_k} w_i$.

The computational efficiency of K-MEANS is $O(tmpn)$ time, where t is the number of hill-climbing iterations over the entire data set, m is the dimension, p is the number of clusters, and n is the number of objects. For KDDM applications, $t, m, p << n$ resulting in $O(n)$ time. Moreover, K-MEANS is very easy to implement. This contrasts favourably with hierarchical clustering algorithms whose computational complexity is in $O(n^2)$ [36], or with the recent $O(n \log n)$-algorithms [29].

This paper will present an approach to efficiently find approximate solutions to the TWGD when data items are referenced in one and two dimensions (the case $m \leq 2$). This has applications in GIS, the unidimensional case is used for the construction of choroplethic maps [6] while the bidimensional case has applications in analysis of the spread in zone patterns [35].

The rest of the paper is organised as follows. Section 2 presents terminology from several communities regrading different formulations of distance-based clustering problems. It is here that we present the non-overlaping restricted version of the TWGD problem. Section 3 demonstrates that the Euclidean version of the TWGD problem restricted to non-overlapping convex-hulls remains NP-complete. Section 4 discusses how to approximately solve the TWGD when restricted to disjoint convex hulls. Final remarks are presented in Section 5.

Notation: We will tend to abbreviate $\sum_{x \in X} f(x)$ by simply writing $\sum_X f(x)$ when it is clear the sum is over all elements x in set X.

2 The contiguous restriction

Unidimensional distance-based clustering problems (the case $m = 1$) are 'easy' in the sense that they are solved optimally by polynomial algorithms [30]. In this section we contrast how distance-based clustering changes as we progress to two dimensions.

It is now appropriate to rewrite the problem in Equation (2). Consider $\text{EUCLID}^2(\vec{x}, X) = \sum_{\vec{x} \in X} d_E(\vec{x}, X)^2$, where X is a fixed set of points in $\Re^m$. It is not hard to see (equating the gradient $\bigtriangledown \text{EUCLID}^2(\vec{x}, X)$ to zero and solving for $\vec{x}$) that $\text{EUCLID}^2(\vec{x}, X)$ is minimised when $\vec{x}$ is the centre of mass of X. That is, $\hat{\vec{x}} = \sum_{\vec{x}_i \in X} \vec{x}_i / \|X\|$. Moreover, algebraic manipulation shows that

$$\text{EUCLID}^2(\hat{\vec{x}}, X) = \frac{1}{2\|X\|} \sum_{\vec{x}_i \in X} \sum_{\vec{x}_j \in X} d_E(\vec{x}_i, \vec{x}_j)^2.$$

Thus, the problem in Equation (2) is equivalent to

$$\text{Minimise} \quad \text{EUCLID}^2(P) = \sum_{i=1}^{n} w_i d_E(\vec{x}_i, rep[\vec{x}_i, C])^2, \quad (3)$$

where
(a) the solution $C = \{\vec{c}_1, \ldots, \vec{c}_k\}$ is a set of p *representative* points in $\Re^m$, and
(b) $rep[\vec{x}_i, C]$ is the closest point in C to $\vec{x}_i$.

In this formulation, the partition into clusters is defined by assigning each $\vec{x}_i$ to its representative $rep[\vec{x}_i, C]$. Those data items assigned to the same representative are in the same cluster. Thus, the p representatives encode the partition of the data and each representative is the centre of mass of its cluster.

However, it is important to note that the proximity of $\vec{x}_i$ to $\vec{x}_j$ is the square of the Euclidean distance. The statistics community refers to the problem in Equation (2) (equivalently Equation (3)) as the *within groups sum of squares* problem [40]. In particular, for geographical data, it is very important to note this aspect [35]. This squared version is known as the gravity problem [17] (or as the centroid problem [18]) in facility location literature because $\text{EUCLID}^2(\vec{x}, X)$ is minimised by the centre of gravity. However, if the problem involves the direct Euclidean distance as in

$$\text{Minimise} \quad \text{EUCLID}(P) = \sum_{i=1}^{n} w_i d_E(\vec{x}_i, rep[\vec{x}_i, C]), \quad (4)$$

then it receives the name of the Webber problem [17] (or the minimum distance problem [31]). This non-squared EUCLID problem has no simple algebraic solution [17] and, even in the case $p = 1$, no algorithm can find the exact solution [31].

The difference between the Webber problem and EUCLID^2 can be simply appreciated in the case $m = 1$. It is not hard to see that the value that minimises $E(\vec{x}, X) = \sum_X d_E(\vec{x}_i, \vec{x})$ is the median (a point in X), while the centre of gravity $\hat{\vec{x}} = \sum_X x_i / \|X\|$ (which may not be in the data set) minimises $E^2(\vec{x}, X) = \sum_{\vec{x}_i \in X} d_E(\vec{x}_i, \vec{x})^2$.

Nevertheless, problems like Equation (3) and Equation (4) minimise a sum; thus, their generic name is p-median problems without much regard to weather the cost of $\vec{x}_i\vec{x}_j$ is the squared Euclidean distance, just the Euclidean distance or any other of the Minkowski distances d_α (i.e. $d_\alpha(\vec{x},\vec{y}) = (\sum_{j=1}^{m} |x_j - y_j|^\alpha)^{1/\alpha}$). At least this is the case in the computational geometry community and the theoretical computer science community [1, and references] (perhaps after Megiddo [30] or after Kariv and Hakimi [25] or perhaps by analogy to the case $m = 1$). The computational geometry community reserves the name p-centres problem for when $\sum$ is replaced by max in Equation (3). This is because we are minimising the radius of a circle to be copied p times and centred at the p representatives to cover the n points in the data set. There is a p-median problem

amongst the operations research community [24, 41] for the problem in Equation (2) when representatives are restricted to be data points (this is referred to as the facility location problem by theoreticians [1]). In clustering for spatial data, medians that are data points have been referred to as medoids [11, 12, 37, 26]. In the case $m = 1$, the problems are typically solved by different variants of dynamic programming in $O(n^2p)$ time [6, 30, and references]. For example, when $m = 1$, the problem in Equation (2) is solvable in $O(n^2p)$ time.

The dynamic programming strategy dates back to 1958, when Fisher observed the so-called *contiguous partition restriction* [16]. This simply states that, in the optimal solution, the groups do not overlap each other. We say that a partition $P = X_1|\ldots|X_p$ is CH-DISJOINT if the convex hull $CV(X_i)$ of X_i does not intersect the convex hull $CV(X_j)$ of any other cluster X_j $(i \neq j)$. This CH-DISJOINT property resulted in many polynomial algorithms for many distance-based clustering approaches in the case $m = 1$ [6].

It is easy to imagine that the restriction to CH-DISJOINT partitions does not change the p-median problem in the case $m = 1$. Namely, if a solution to the unidimensional p-median problem (or the problem in Equation (2)) is assumed to be optimum and two clusters overlap, then we can swap two data points (chosen with some care) between two overlapping clusters and obtain an improved solution. Thus, in the real line, solving p-median (or the problem in Equation (2)) with the added *contiguous partition restriction* is equivalent to its original unrestricted problems. But, the restriction provides the clue to solve these problems by dynamic programming in polynomial time.

At some point there was some debate over how to extend this approach to higher dimensions. The so-called *string property* [45] was revised to a property emphasising the notion of representative present in the problem in Equation (2) [40]. At the time, the direct generalisation was conjectured; namely, "... for a partition to be optimal, the convex hulls of the subsets must be non-overlapping" [40]. This was apparently proved by Bock [3], but it is not hard to prove (because the Voronoi regions of representatives in an optimal solution must contain the points of their respective clusters).

Apparently, attention drifted away from the TWGD problem as the NP-hardness results emerged for the graphical and geometric (Euclidean and even bidimensional, i.e. $m = 2$) versions of representative clustering problems [4, 22, 27, 28, 30, 39]. Work concentrated on suitable approximation algorithms for them [8, 41]. Others concentrated on special cases where polynomial algorithms can be found (for example, the case $p = 2$ [23]). More recently, theoretical results have concentrated on polynomial approximation schemes for the representative-based clustering approaches [1, and references].

3 TWGD restricted to CH-DISJOINT partitions is NP-hard

TWGD is very different from representative approaches like p-median and p-centres. Although Equation (2) looks like a weighted version of Equation (1), even in the Euclidean case the solution for TWGD may be a partition whose clusters are not CH-DISJOINT. For example, even for the case $m = 1$, there are point sets where no optimal solution may be CH-DISJOINT. Consider $x_1 = -10$, $x_2 = 10$, $x_3 = -1$, $x_4 = 0$ and $x_5 = 1$. The optimal TWGD solution is $X_1 = \{x_1, x_2\}$ and $X_2 = \{x_3, x_4, x_5\}$ with TWGD=24 (any other partition puts together any two points of X_2 with a point of X_1 and will incur in a cost of at least 29).

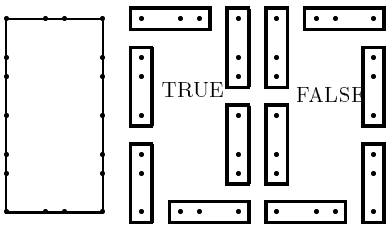

Figure 1: A variable component (circuit) with $3 \times 6 = 18$ points and its two optimal CH-DISJOINT partitions of 6 groups.

Thus, the TWGD problem and its restriction to CH-DISJOINT partitions are not equivalent. Since now we know that the graph version (even for $p = 2$ [4]) and the Euclidean version [27] of the TWGD are NP-complete, the equivalence would have shown that the TWGD problem restricted to CH-DISJOINT partitions is NP-complete. Since the restriction to CH-DISJOINT partitions results in a different problem, perhaps we can hope for polynomial algorithms. We now show that the hope is the same as the hope for finding polynomial algorithms for NP-complete problems.

Theorem 3.1 *The* TWGD *Euclidean problem restricted to* CH-DISJOINT *partitions in NP-complete.*

Proof: The instance we consider provides a set S with n points in the plane and integers p and B. The decision question is if there exists a CH-DISJOINT partition of S into p parts whose TWGD value is less than B.

The proof is a component design [20] proof that reduces 3-satisfiability to the CH-DISJOINT-restricted TWGD problem. In fact, the proof is similar to other proofs for Euclidean clustering problems [30]. An instance of 3-SAT is converted into an instance of the CH-DISJOINT-restricted TWGD decision problem, thus the set S and the constants p and B are built as we go along. We first define a component for each variable. These components (the *Boolean variable* components) will be called *circuits* for reasons that will be obvious soon. The variable component is optimally solved in only two ways that correspond to the assignment of true or false to the Boolean variable. These two ways can be thought as travelling the circuit clockwise or counterclockwise (Fig. 1).

A second type of component is a *clause component* to ensure each clause is satisfied. The circuit for a variable visits each of the clause components in which the variable (or its negation) appears. Moreover, it arrives at the clause component in one of two ways. These two ways represent if it is the variable or if it is its negation what appears in that particular clause. In this way, the assignment of Boolean values to variables (and their negations) is consistent.

Finally, a third type of component is needed. This is just because of the embedding of the instance in the Euclidean plane. These components ensure that they can be placed wherever needed to make two circuits cross without affecting the meaning (orientation) of the circuits. Thus, we will call them crossing components.

Fig. 1 shows a circuit (or variable) component and the two ways in which it can be optimally clustered into CH-DISJOINT partitions. The circuit $C_i = \langle s_0^i, s_1^i, \ldots, s_{3p_i}^i = s_0^i \rangle$ for variable u_i is made of $3p_i$ points (p_i an integer).

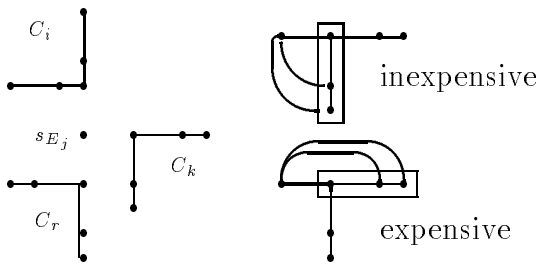

Figure 2: A clause component (point) with 3 circuit corners and the two ways a corner incorporates the clause.

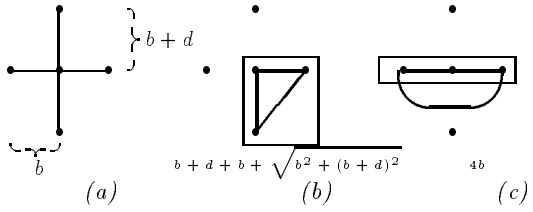

Figure 3: The component to allow circuits to cross.

Points along the circuit are spaced 1 or $b > 1$ by the rule: $d(s^i_j, s^i_{j+1}) = 1$ if $j \mod 3 = 0$ and $d(s^i_j, s^i_{j+1}) = d$ otherwise. The circuit makes orthogonal turns at points where the previous and next point in the circuit are at distance b. Clusters have a cost of $2(1+d)$ because the number of groups requested is p_i and each cluster has 3 consecutive points in the circuit. Thus p_i is accumulated into p.

Clause components are very simple, they correspond to just one point; see Fig. 2. Because we are reducing 3-SAT, we know that at most 3 circuits arrive at a clause. A circuit arriving at a clause E_j places one of its corners at distance b from s_{E_j}. No more clusters are allowed (p is not increased), thus the point s_{E_j} for the clause component must be included in a cluster of at least one of the 3 circuits. Moreover, each circuit can incorporate the clause point in an expensive and an inexpensive way. The bound B will be set so that at least one of the 3 circuits includes s_{E_j} in an inexpensive way. The inexpensive or expensive way of connecting a circuit is chosen according to whether the variable or its negation appears in the clause. It is not hard to see that because at most 3 circuits arrive at a clause, all combinations of expensive and inexpensive arrangements for the 3 circuits can be configured (perhaps with some circuit crossings, but this is solved by the next component).

Crossing components are also very simple. They replace a point in each of the crossing circuits by a cross as illustrated in Fig. 3. The cross has length b in one dimension and length $b+d$ in the orthogonal dimension with $d \leq 0.5$. The crossing components are allowed to be covered with one cluster and the bound B allows for only a triangle in the cross to be covered at cost of $b+b+d+\sqrt{b^2+(b+d)^2}$.

These crossing components are inserted between consecutive segments of distance b in a circuit. Fig. 4 illustrates the crossing of two circuits. This figure illustrates the role of the small constant d in adjusting the circuits to allow them to comply with the requirement that corners have only points that are distance b from its predecessor and successor in the circuit. This also illustrates that circuits may need to be padded with more points in order to travel further away (shown in Fig. 4 with ▫). However, for a circuit with $3p_i$ points only p_i clusters will be allowed.

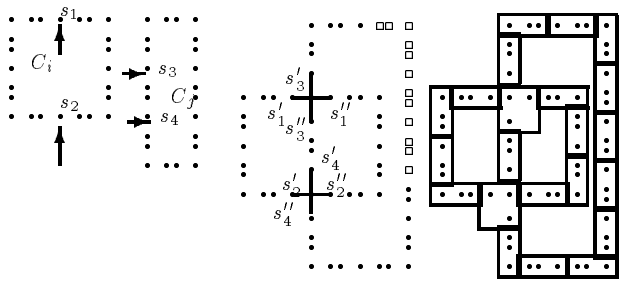

Figure 4: Crossing two circuits.

It remains to be shown that the optimal TWGD clustering that is CH-DISJOINT must preserve the direction of the circuit through crossings. We show that the grouping shown in Fig. 3 (c), while apparently producing a smaller TWGD value for the crossing component, is actually suboptimal. In reference to Fig. 3 (c), the circuit passing horizontally would replace two groups of cost $2(b+1)$ each for one group of cost $4b$ and one group of cost 1. However, the circuit passing vertically would replace a group of cost $2(b+1)$ by a group of cost $4b+2$ (because the one group would be extended by an item at least b apart). Thus, the new cost of Fig. 3 (c) is $8b+4$ while the cost of Fig. 3 (b) is $6b+2+d+\sqrt{b^2+(b+d)^2}$ and because $d \leq 0.5$ it is not hard to show that this cost is no more that $(6+\sqrt{2})b+3$, as required.

The rest of the proof follows standard proofs of NP-completeness. □

4 Algorithms for approximating optimal CH-DISJOINT partitions

In spatial clustering, there are many situations where the desired clusters are expected to be convex. In fact, distance-based clustering approaches that define the groups by representatives and assign data point to the nearest representative construct clusters that are convex. Clusters are in direct correspondence with the Voronoi regions of the Voronoi diagram of the representatives. Thus, solving approximately the TWGD clustering problem restricted to CH-DISJOINT partitions is interesting.

Although we just have shown that the problem complexity for the CH-DISJOINT restriction remains NP-hard, we believe that the restriction to CH-DISJOINT partitions can produce approximation algorithms that are at least as efficient as previous attempts in solving the unrestricted TWGD clustering problem. We illustrate this point now. We adapt well studied local search hill-climbers known as *interchange heuristics* [8, 24, 34, 43] to TWGD restricted to CH-DISJOINT partitions. These heuristics are typically used for the p-medians problem (solving Equation (2) with the added restriction that the representative be data points) and recently they have been used for the general TWGD problem [33].

Similar adaptations will carry over to alternative methods such as genetic algorithms [13, and references] and simulated annealing [34], as these methods have extended the

local search hill-climbers. Due to their ability to escape from and improve over local optima, simulated annealing and genetic algorithms open the possibility of better approximation; however, the computation time required is longer than that of hill-climbers. A critical factor in the efficiency of all these algorithms is the computational effort for evaluating the change in the TWGD value for a new partition resulting from a previous partition.

The hill-climbing nature of the local search heuristics is clearly revealed if we structure the search space of TWGD as a graph. The nodes of this graph are all partitions $P = X_1|\ldots|X_p$ representing a choice of p groups. The edges of the graph are defined as follows: two nodes P and P' are adjacent if and only if they differ in only the assignment of one data point. Namely, a partition of X can be considered a function of X onto a set of p colours. Two partitions are adjacent if they differ in only the colour of one data point.

The interchange heuristics start at a randomly-chosen solution P^0 (that is, a random node in the graph), and explore the graph by moving from the current node to one of its neighbours. Letting P^t be the current node at time step t, the heuristic examines a set $N(P^t)$ of neighbouring nodes of P^t, and considers the best alternative to P^t in this neighbourhood: the node $M(P^{t+1}) = \min_{P\in N(P^t)} \mathrm{TWGD}(P)$. Provided that the new node P^{t+1} is an improvement over the old (that is, if $M(P^{t+1}) < M(P^t)$), P^{t+1} becomes the current node for time step $t+1$. When no better solution is found in the neighbourhood $N(P^t)$, the search halts.

The interchange hill-climbers proposed to date define the neighbourhood set $N(P^t)$ in varying ways. In finding a local optimum of high quality, an original heuristic proposed in 1968 by Teitz and Bart [43] has proven the most effective. We will refer to this heuristic as TAB.

Its adaptation to solving TWGD works as follows [33]. When searching for a profitable interchange, it considers the data points in turn, according to a fixed circular ordering $(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$ of the data. Whenever the turn belonging to a data point $\vec{x}_i$ comes up, it is considered for changing its group (changing its colour to any of the other $p-1$ colours). The most advantageous interchange P^j of these $p-1$ alternatives is determined. If P^j is better than P^t, then P^j becomes the new current solution P^{t+1}; otherwise, $P^{t+1} = P^t$. In either case, the turn then passes to the next data point in the circular list, $\vec{x}_{i+1}$ (or $\vec{x}_1$ if $i = n$). If a full cycle through the data set yields no improvement, a local optimum has been reached, and the search halts.

Rather than computing TWGD on the $p-1$ neighbours of P^t (which potentially requires $\Theta(pn^2)$ time, what is computed is $\mathrm{TWGD}(P^t) - \mathrm{TWGD}(P)$ (for $P \in N(P^t)$). The time required to compute this difference is $O(n)$ operations. Therefore, the time required to test replacing the colour of $\vec{x}_i$ is $O(pn)$ time. In most situations, p can be viewed as a small constant, and thus the test can be considered to take linear time. Since the heuristic halts with a complete scan of the data set and empirical evidence suggest that the total number passes to the list is constant, this heuristic requires $\Theta(n^2)$ time in total. The TAB heuristic forbids the reconsideration of $\vec{x}_i$ for inclusion until all data points have been considered as well. The heuristic can, therefore, be regarded as a local variant of *tabu search* [21]. TAB's careful design balances the need to explore a variety of possible interchanges against the 'greedy' desire to improve the solution as quickly as possible. The TAB heuristic has been remarkably successful in its application to facility location problems [34, 41], as well as the clustering of large sets of low-dimensional spatial data [12].

Given the above description it is not hard to adapt TAB to TWGD restricted to CH-DISJOINT partitions. The nodes of the search graph are all partitions into p groups that have disjoint convex hulls. Two nodes are adjacent if they differ in the colour of only one data point. We again order the data points in a circular list, evaluating each $\vec{x}_i$ in turn for a change of group. If a change of group for $\vec{x}_i$ results in a CH-DISJOINT partition with a lower TWGD value, we adopt the change of colour that reduces TWGD the most. In any case, the turn passes to the successor of $\vec{x}_i$ in the circular list. A complete scan of the circular list with no improvement forces the search to stop.

We now provide some details on how this can be implemented for $m = 2$. Although we have said that empty groups are not accepted, the initial solution can be $p-1$ empty groups and a group with all elements. This is rapidly adjusted to $p-1$ points of the convex hull of X in singleton groups and a group with the remaining data points, or something better. Alternatively, in $O(n \log n)$ preprocessing the convex hull of X can be found and then $p-1$ points on the hull assigned to singleton groups. Also, initialisation with a guessed solution is possible, for example, one derived from the Delaunay Triangulation of the points [11].

Examining a colour swap for a point $\vec{x}_i$ can be performed in $O(n)$ time. In fact, only those $\vec{x}_i$ that are in the convex-hull of their current group need to be examined. Otherwise, they can be skipped immediately, since no new colour will result in a CH-DISJOINT partition. This test will require $O(p \log n)$ time in the worst case since intersection of convex polygons can be tested in logarithmic time. A colour swap of $\vec{x}_i$ implies a difference in TWGD values that can be computed in $O(n)$ time. As in the general TWGD case, $\vec{x}_i$ total distance to the points in its new group and $\vec{x}_i$ total distance to the points in its old group are the only terms participating in determining the difference in TWGD value. These are at most $O(n)$ terms.

Finally, data structures for dynamically maintaining the p convex hulls (with respect to insertions and deletions) are possible. Overmars and van Leewen structures will suffice since they require $O(\log^2 n)$ time per operation [38].

Alternatively, the restricted CH-DISJOINT TWGD problem that we have presented here may be amendable to polynomial approximation schemes [1]. These randomised algorithms produce, with very high probability, a solution that is within a constant factor c from the optimum. For example, for the p-medians problem as in Equation (2) (where the points in the set C are anywhere in space but $m = 2$) there is an approximation scheme that for any $c > 0$ produces a solution C with cost $M(C)$ at most $1 + 1/c$ times the optimum in $O(n^{O(c+1)})$ time [1]. While for $c = 2$ this result seems less effective than the success observed in practice by interchange hill-climbing heuristics, it does offer guarantees on the quality of the solution. These approximation schemes are based on techniques that apply to problems where (a) the objective function is a sum of edge lengths and (b) some form or variant of a patching lemma holds. Clearly, (a) holds for the TWGD problem and also for its restriction to CH-DISJOINT partitions. We are currently working on (b). However, we believe that (b) will hold better for the CH-DISJOINT version since the techniques are used to develop bidimensional dynamic programming algorithms. Recall that the motivation for CH-DISJOINT partitions comes from the fact that, in the case $m = 1$, this leads to polynomial-time dynamic programming algorithms for clustering problems.

5 Final Remarks

Today, the most popular method for clustering in KDDM is K-MEANS [2, 14]. However, the interchange heuristic proposed here for solving TWGD has all the desired properties for KDDM. It is stoppable and resumable, with an approximated solution always ready, and can work on incremental datasets. K-MEANS has some foundation on Maximum Likelihood. Namely, it alternates a step that assigns means (representative) for approximately maximising

$$\prod_{i=1}^{p} Prob(rep[s_i, C]|S, \theta, M),$$

with a step that assigns the parameters θ (a vector of parameters) of a mixture model M so as to maximises

$$Prob(\theta|rep[s_i, C], S, M).$$

In fact, its popularity may be attributed to its simplicity for implementation and that it takes linear time. However, not much else favours this algorithm.

1. From an optimisation point of view, it often converges to a local optimum of poor quality.
2. It is very sensitive to the presence of noise and outliers, as well as to the initial random clustering.
3. The method is statistically biased (this has favoured the emergence of other statistical methods such as 'expectation maximization' [7]) and statistically inconsistent (this has favoured the emergence of Bayesian and Minimum Message Length methods[9]). However, these alternative methods require the user to define a probabilistic model of the classes and their high sensitivity to the initial random solution has prompted researchers to incorporate initialisation mechanisms [14].

KDDM is exploratory and may involve exploration of alternative models. The application at hand determines much of the modelling the analyst may find suitable. TWGD offers an alternative distance-based clustering criterion that does not need representatives. Representatives are commonly adopted as prototypes of the data points of their cluster. However, it is possible that this may have no valid interpretation; for example, the average of the coordinates of a group of schools may indicate that the representative school lies in the middle of a lake. Thus, it is necessary to have a toolkit of clustering methods and approaches. In particular, the TWGD seems informative along other criteria in analysing geographically referenced data [33].

The result presented here fills in a space on the complexity of Euclidean clusters problem whose parts must be CH-DISJOINT. We also have shown that it can be solved approximately by efficient algorithms. We look forward to see renewed interest in this clustering criterion.

References

[1] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean k-medians and related problems. In *30th Annual ACM Symposium on the Theory of Computing (STOC)*. ACM Press, 1998.

[2] M.J.A. Berry and G. Linoff. *Data Mining Techniques — for Marketing, Sales and Customer Support*. John Wiley & Sons, NY, USA, 1997.

[3] H.H. Bock. *Automatische Klassifikation*. Vandenhoek & Ruprecht, Göttingen, 1974.

[4] P. Brucker. On the complexity of clustering problems. In R. Henn, B.H.B Korte, and W.W. Oetti, editors, *Optimization and Operations Research: workshop held at the University of Bonn*, Berlin, 1978. Springer Verlag Lecture Notes in Economics and Mathematical Systems.

[5] P. Chessman, J. Kelly, M. Self, J. Stuts, W. Taylor, and D. Freedman. Autoclass: A bayesian classification system. In *5th Int. Conf. on Machine Learning*, pages 54–64, San Mateo, CA, 1988. Morgan Kaufmann Publishers.

[6] R.G. Cromley. A comparison of optimal classification strategies for choroplethic displays of spatially aggregated data. *Int. J. of Geographical Information Systems*, 10(4):405–424, 1996.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likehood from incomplete data via the EM algorithm. *J. Royal Statistical Society B*, 39:1–38, 1977.

[8] P. Densham and G. Rushton. A more efficient heuristic for solving large p-median problems. *Papers in Regional Science*, 71:307–329, 1992.

[9] D. Dowe, R.A. Baxter, J.J. Oliver, and C. Wallace. Point estimation using the Kullback-Leibler loss function and MML. In X. Wu, R. Kotagiri, and K.K. Korb, editors, *. of Second Pacific-Asia Conf. on Knowledge Discovery and Data Mining PAKDD-98*, pages 87–95, Melbourne, Australia, 1998. Springer-Verlag LNAI 1394.

[10] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, NY, USA, 1973.

[11] V. Estivill-Castro and M.E. Houle. Roboust clustering of large geo-referenced data sets. In N. Zhong and L. Zhow, editors, *3rd Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD-98)*, pages 327–337. Springer-Verlag Lecture Notes in Artificial Intelligence 1574, April 1999.

[12] V. Estivill-Castro and A.T. Murray. Discovering associations in spatial data - an efficient medoid based approach. In X. Wu, R. Kotagiri, and K.K. Korb, editors, *. of the 2nd Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD-98)*, pages 110–121, Melbourne, Australia, 1998. Springer-Verlag LNAI 1394.

[13] V Estivill-Castro and R. Torres-Velázquez. Hybrid genetic algorithm for solving the p-media problem. In A Yao, R.I. McKay, C.S. Newton, J.-H Kim, and T. Furuhashi, editors, *. of Second Asia Pacific Conf. On Simulated Evolution and Learning SEAL-98*. Springer Verlag LNAI, 1999. to appear.

[14] U. Fayyad, C. Reina, and P.S. Bradley. Initialization of iterative refinement clustering algorithms. In R. Agrawal and P. Stolorz, editors, *Fourth Int. Conf. on Knowledge Discovery and Data Mining*, pages 194–198. AAAI Press, 1998.

[15] D.H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.

[16] W.D. Fisher. On grouping for maximum homogeneity. *J. American Statistical Association*, 53:789–798, 1958.

[17] R.L. Francis. *Facility layout and location: An analytical approach.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1974.

[18] R.L. Francis. *Facility layout and location: An analytical approach.* Prentice-Hall, Inc., Englewood Cliffs, NJ, second edition, 1992.

[19] A.A. Freitas and S.H. Lavington. *Mining Very Large Databases with Parallel Processing.* Kluwer Academic Publishers, London, 1998.

[20] M.R. Garey and D.S. Johnson. *Computers and Intractability — A guide to the Theory of NP-Completeness.* Freeman, NY, 1979.

[21] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5:533–549, 1986.

[22] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[23] J. Hershberger and A. Suri. Finding tailored partitions. *J. of Algorithms*, 12:431–463, 1991.

[24] M. Horn. Analysis and computation schemes for p-median heuristics. *Environment and Planning A*, 28:1699–1708, 1996.

[25] O. Kariv and L. Hakimi. An algorithmic approach to network location problems. I:the p-medians. *SIAM J. of Applied Mathematics*, 37(3):539–560, December 1979.

[26] L. Kaufman and P.J. Rousseuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley & Sons, NY, USA, 1990.

[27] M. Krivánek. Hexagonal unit network - a tool for proving the NP-completeness results of geometric problems. *Information Processing Letters*, 22:37–41, 1986.

[28] M. Krivánek. On the complexity of clustering. In E. Diday, Y. Escoufier, L. Lebart, J. Pages, and R. Schektman, Y.and Tomassone, editors, *Data Analysis and Informatics, IV, Fourth Int. Symposium on Data Analysis and Informatics*, pages 89–96, Versailles, October 1986. INRIA, North-Holland.

[29] D. Krznaric and C. Levcopoulos. Fast algorithms for complete linkage clustering. *Discrete & Computational Geometry*, 19:131–145, 1998.

[30] N. Megiddo and K.J. Supowit. On the complexity of some common geometric location problems. *SIAM J. on Computing*, 13(1):182–196, February 1984.

[31] Z.A. Melzak. *Companion to concrete mathematics; mathematical techniques and varios applications.* John Wiley & Sons, NY, USA, 1973.

[32] R.S. Michalski and R.E. Stepp. Automated construction of classifications: Clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(5):683–690, 1983.

[33] A. Murray. Spatial characteristics and comparisons of clustering models. *Geographical Analysis.* to appear.

[34] A.T. Murray and R.L. Church. Applying simulated annealing to location-planning models. *J. of Heuristics*, 2:31–53, 1996.

[35] A.T. Murray and V. Estivill-Castro. Cluster discovery techniques for exploratory spatial data analysis. *Int. J. of Geographic Information Systems*, 12(5):431–443, 1998.

[36] F. Murtagh. Comments of "Parallel algorithms for hierarchical clustering and cluster validity". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10):1056–1057, 1992.

[37] R.T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *. of the 20th Conf. on Very Large Data Bases (VLDB)*, pages 144–155, San Francisco, CA, 1994. Santiago, Chile, Morgan Kaufmann Publishers.

[38] M.H. Overmars and J. van Leewen. Maintenance of configurations in the plane. *J. of Computer and System Sciences*, 23:166–204, 1981.

[39] C.H. Papadimitriou. Worst-case and probabilistic analysis of geometric location problems. *SIAM J. of Computing*, 10:542–557, 1981.

[40] M. Rao. Cluster analysis and mathematical programming. *J. American Statistical Association*, 66:622–626, 1971.

[41] D. Rolland, E. Schilling and J. Current. An efficient tabu search procedure for the p-median problem. *European J. of Operations Research*, 96:329–342, 1996.

[42] K. Rosing and C. ReVelle. Optimal clustering. *Environment and Planning A*, 18:1463–1476, 1986.

[43] M.B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.

[44] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *statistical Analysis of finite Mixture Distributions.* John Wiley & sons, UK, 1985.

[45] H. Vinod. Integer programming and the theory of grouping. *J. American Statistical Association*, 64:506–517, 1969.

[46] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *J. Royal Statistical Society, Series B*, 49(3):223–265, 1987.

[47] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In M. Jarke, editor, *23rd Int. Conf. on Very Large Data Bases*, pages 186–195, Athens, Greece, August 1997. VLDB, Morgan Kaufmann Publishers.

[48] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH:an efficient data clustering method for very large databases. *SIGMOD Record*, 25(2):103–114, June 1996. 1996 ACM SIGMOD Int. Conf. on Management of Data.