An Algorithm to Identify Clusters of Solutions in Multimodal Optimisation

Pedro J. Ballester and Jonathan N. Carter

Imperial College London, Department of Earth Science and Engineering, RSM Building, Exhibition Road, London SW7 2AZ, UK.
{p.ballester,j.n.carter}@imperial.ac.uk

Abstract. Clustering can be used to identify groups of similar solutions in Multimodal Optimisation. However, a poor clustering quality reduces the benefit of this application. The vast majority of clustering methods in literature operate by resorting to a priori assumptions about the data, such as the number of cluster or cluster radius. Clusters are forced to conform to these assumptions, which may not be valid for the considered population. The latter can have a huge negative impact on the clustering quality. In this paper, we apply a clustering method that does not require a priori knowledge. We demonstrate the effectiveness and efficiency of the method on real and synthetic data sets emulating solutions in Multimodal Optimisation problems.

1 Introduction

Many real-world optimisation problems, particularly in engineering design, have a number of key features in common: the parameters are real numbers; there are many of these parameters; and they interact in highly non-linear ways, which leads to many local optima in the objective function. These optima represent solutions of distinct quality to the presented problem. In Multimodal Optimisation, one is interested in finding the global optimum, but also alternative good local optima (ie. diverse high quality solutions). There are two main reasons to seek for more than one optimum. First, real-world functions do not come without errors, which distort the fitness landscape. Therefore, global optima may not correspond to the true best solution. This uncertainty is usually addressed by considering multiple good optima. Also, the best solution represented by a global optimum may be impossible to implement from the engineering point of view. In this case, an alternative good solution could be considered for implementation.

Once a suitable search method is available, an ensemble of diverse, high quality solutions are obtained. Within this ensemble, there are usually several groups of solutions, each group representing a different optimum. In other words, the ensemble of solutions is distributed into clusters. Clustering can be defined as the partition of a data set (ensemble of solutions) into groups named clusters (part of the ensemble associated with an optimum). Data points (individuals) within each cluster are similar (or close) to each other while being dissimilar to the remaining points in the set.

C.C. Ribeiro and S.L. Martins (Eds.): WEA 2004, LNCS 3059, pp. 42–56, 2004.

The identification of these clusters of solutions is useful for several reasons. First, at the implementation stage, we may want to consider distinct solutions instead of implementing similar solutions. This is specially convenient when the implementation costs time and money. On the other hand, the uncertainty associated with one solution can be estimated by studying all the similar solutions (ie. those in the same cluster). Also, one may want to optimise the obtained solutions further. This is done by using a faster (but less good at searching) optimiser on the region defined by all solutions. With the boundaries provided by the clustering algorithm, you could do the same but for each cluster (ie. for each different high performance region). Lastly, an understanding of the solutions is needed. In real world engineering design, for instance, it is common to have many variables and a number of objectives. Packham and Parmee [8] claim that in this context it is extremely difficult to understand the possible interactions between variables and between variables and objectives. These authors pointed out the convenience of finding the location and distribution of different High Performance regions (ie. regions containing high quality solutions).

There are many clustering algorithms proposed in the literature (a good review has been written by Haldiki et al. [7]). The vast majority of clustering algorithms operate by using some sort of a priori assumptions about the data, such as the cluster densities, sizes or number of clusters. In these cases, clusters are forced to conform to these assumptions, which may not be valid for the data set. This can have a huge negative impact on the clustering quality. In addition, there is a shortage in the literature of effective clustering methods for high dimensional data. This problem is known as the Curse of Dimensionality [6] in Clustering. This issue is discussed extensively in [1].

In this work, we apply a new algorithm [2] for clustering data sets obtained by the application of a search and optimisation method. The method is known as CHIDID (Clustering HIgh DImensional Data) and it is aimed at full dimensional clustering (ie. all components have a clustering tendency). To use this algorithm it is not necessary to provide a priori knowledge related to the expected clustering behaviour (eg. cluster radius or number of clusters). The only input needed is the data set to be analysed. The algorithm does contain two tuning parameters. However, extensive experiments, not reported in this paper, lead us to believe that the performance of the algorithm is largely independent of the values of these two parameters, which we believe are suitable for most data sets and are used for all our work. CHIDID scales linearly with the number of dimensions and clusters. The output is the assigment of data points to clusters (the number of clusters is automatically found).

The rest of this paper is organised as follows. We begin by describing the clustering method in Sect. 2. Section 3 explains how the test data sets are generated. The analysis of the results is discussed in Sect. 4. Lastly, Sect. 5 presents the conclusions.

2 Clustering Method

We describe our clustering method in three stages. We start by introducing the notation used. Next, the clustering criterion is presented as a procedure used to find the cluster in which a given data point is included. Finally, the clustering algorithm is introduced. The operation of this algorithm is based on the iterative application of the clustering criterion until all clusters and outliers are found.

2.1 Notation

We regard the population as a collection of N distinct vectors, or data points in a M-dimensional space, over which the clustering task is performed. We represent this data set as

$$\Omega = \{\boldsymbol{x}^i\}_{i=1}^{N}, \text{ with } \boldsymbol{x}^i \in \mathrm{R}^M \ \forall i \in I = \{1, \ldots, N\} \ . \tag{1}$$

Clustering is the process by which a partition is performed on the data set. Points belonging to the same cluster are similar to each other, but dissimilar to those belonging to other clusters. This process results in C pairwise disjoint clusters, whose union is the input data set. That is

$$\Omega = \bigcup_{k=1}^{C} \Omega_k, \text{ with } \Omega_k = \{\boldsymbol{x}^i\}_{i \in I_k} \ . \tag{2}$$

where Ω_k is the k^{th} cluster, I_k contains the indices for the data points included in the k^{th} cluster and $\Omega_k \bigcap \Omega_l = \emptyset \ \forall k \neq l$ with $k, l \in K = \{1, \ldots, C\}$.

An outlier is defined as a data point which is not similar to any other point in the data set. Hence, an outlier can be regarded as a cluster formed by a single point.

Proximity (also known as similarity) is the measure used to quantify the degree of similarity between data points. A low value of the proximity between two points means these points are similar. Conversely, a high value of their proximity implies that the points are dissimilar. In this work, the Manhattan distance is adopted as the proximity measure. Thus the proximity of $\boldsymbol{x}^i$ with respect to $\boldsymbol{x}^l$ is calculated as

$$p_{il} = \sum_{j=1}^{M} |x_j^l - x_j^i| \ . \tag{3}$$

In practice, the value range of a given component can be very large. This would dominate the contribution of the components with much smaller value ranges. In this case, we would recommend scaling all components to the same range so that every component contributes equally to the proximity measure. However, in this paper the scaling will not be necessary since all components will be in the same interval.

Based on the proximity measure, a criterion is needed to determine if two points should be considered members of the same cluster. The quality of the clustering directly depends on the choice of criterion. In this work, clusters are regarded as regions, in the full dimensional space, which are densely populated with data points, and which are surrounded by regions of lower density. The clustering criterion must serve to identify these dense regions in the data space.

2.2 Clustering Criterion

As we previously described, points belonging to the same cluster have a low proximity between them, and high proximity when compared with points belonging to other clusters. The proximity $\{p_{il}\}_{l=1}^{N}$ from an arbitrary point $\boldsymbol{x}^i$ to the rest of the data set (ie. $\boldsymbol{x}^i$ with $l \neq i$) should split into two groups, one group of similar low values and the other of significantly higher values. This first group corresponds to the cluster to which $\boldsymbol{x}^i$ belongs to. The goal of the clustering criterion is to determine the cluster cutoff in an unsupervised manner. The criterion is also expected to identify outliers as points which are not similar to any other point in the data set.

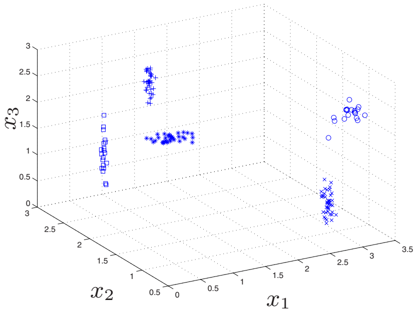

Fig. 1. Example data set with N= 150, M= 3 and C= 5

In order to illustrate the operation of the clustering criterion, consider the data set shown in Fig. 1. It contains 150 points unevenly distributed in 5 clusters in a 3-dimensional space. Let us take a point from one of the clusters (say the k^{th}) in the example, which will be referred to as the cluster representative and denoted by $\boldsymbol{x}^{i_k}$. We apply the following procedure to determine which are its cluster members:

1. Calculate the sequence $\{p_{i_k l}\}_{l=1}^{N}$, using (3), and sort it in increasing order to get $\{d_l\}_{l=1}^{N}$. The sorting operation is represented as a correspondence between sets of point indices, that is, $S : I \to L$. The plot of d_l can be found

at the bottom part of Fig. 2. Note that $d_1 = 0$ corresponds to $p_{i_k i_k}$, d_2 is the nearest neighbour to $\boldsymbol{x}^{i_k}$ and so on.

2. Define the relative linear density up to the l^{th} data point as $\theta_l = \frac{l}{N}\frac{\beta+d_N}{\beta+d_l}$ and compute $\{\theta_l\}_{l=1}^N$. This expression is equivalent to the cumulative number of points divided by an estimation of the cluster size relative to the same ratio applied over all N points. β is a parameter that will be discussed later (we recommend that $\beta = \frac{2}{3}d_2$).
3. Define the relative linear density increment for the l^{th} point as $\triangle\theta_l = |\theta_l - \theta_{l-1}|$ and calculate the sequence $\{\triangle\theta_l\}_{l=2}^N$. The plot of $\triangle\theta_l$ is presented at the top part of Fig. 2. If $\triangle\theta_l$ is high then it will be unlikely that the l^{th} point forms part of the cluster.
4. Calculate $l_1 = \{l \in \{2,\dots,N\} \mid \triangle\theta_l = max(\{\triangle\theta_{l'}\}_{l'=2}^N)\}$ (ie. the position of the highest value of $\triangle\theta_l$ in Fig. 2). Define $(l_1 - 1)$ as the provisional cluster cutoff, and it means that only the $l_1 - 2$ closer points to the cluster representative would be admitted as members of the cluster.
5. Define the significant peaks in $\{\triangle\theta_l\}_{l=2}^N$ as those above the mean by α times the standard deviation, ie. $\{l \in \{2,\dots,N\} \mid \triangle\theta_l > \overline{\triangle\theta_l} + \alpha\ \sigma_{\triangle\theta_l}\}$. α is a parameter that will be discussed later (we recommend $\alpha = 5$).
6. Identify the significant peak with the lowest value of l, that is, the most left significant peak and take it as the definitive cluster cutoff l_c. In Fig. 2 (top), the example shows two significant peaks, which are above the horizontal dotted line given by $\overline{\triangle\theta_l} + \alpha\ \sigma_{\triangle\theta_l}$. If all the peaks are below $\overline{\triangle\theta_l} + \alpha\ \sigma_{\triangle\theta_l}$, then take the highest as the cluster cutoff, ie. $l_c = l_1 - 1$.
7. Finally, the k^{th} cluster is given by $L_k = \{1, ..., l_c\}_k$. Invert the sorting operation ($I_k = S^{-1}(L_k)$) to recover the original point indices which define the k^{th} cluster as $\Omega_k = \{\boldsymbol{x}^i\}_{i\in I_k}$.

Under the $\triangle\theta_l$ representation pictured in Fig. 2 (top), the natural cluster to which the cluster representative $l = 1$ belongs becomes distinguishable as the data points to the left of the most left significant peak. As has been previously discussed, cluster members share a similar value of d_l among themselves and are dissimilar when compared to non-members. As a consequence, the sequence $\{\triangle\theta_l\}_{l=2}^{l_c}$ contain low values, whereas $\triangle\theta_{l_c+1}$ is higher in comparison. Nevertheless, $\triangle\theta_{l_c+1}$ is not necessarily the highest in the sequence $\{\triangle\theta_l\}_{l=2}^N$ and therefore Step 6 is required to localise $\triangle\theta_{l_c+1}$ as the most inner peak. A decrease in the value of α will result in clusters with a lower proximity between them, in other words, more restrictive clusters. In the light of these considerations, α can be regarded as a threshold for the clustering resolution. Despite the fact that α plays a relevant role in data sets containing few points and dimensions, as the number of points and dimensions increase the plot of $\triangle\theta_l$ tend to show a single sharp peak corresponding to the cluster cutoff. It suffices to take any high value of α.

Another issue is the role of β in the definition of θ_l at Step 2. d_l is a better estimation of the cluster diameter than $\beta + d_l$. However, the latter is preferred since with the former neither θ_1 nor $\triangle\theta_2$ would be defined due to the zero value of d_1. These quantities lack meaning in the sense that a density cannot be defined

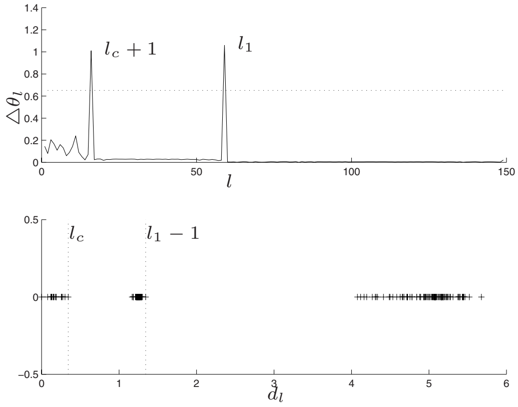

Fig. 2. Characteristic $\{\triangle\theta_l\}_{l=2}^{N}$ (top) and $\{d_l\}_{l=1}^{N}$ (bottom, represented by '+' signs) plots for a given point in the example data set

for only one point. Without $\triangle\theta_2$, it would not be possible to find out whether the cluster representative is an outlier or not. A suitable value of β will vary depending on the cluster and hence it is preferable to link it to an estimation of the proximity between points in the cluster. We choose to fix $\beta = \gamma d_2$ with $0 < \gamma < 1$ (a high γ implies a higher tendency to include outliers in clusters). In appendix A, γ is shown to be a threshold for outlier detection. Note that an additional advantage of posing β in this form is that it becomes negligible with respect to high values of d_l and hence ensures the density effect of θ_l. We set $\gamma = \frac{2}{3}$ throughout this work.

Finally, the detection of outliers in very high dimensional spaces is a difficult task that requires a preliminary test in order to ensure the effectiveness of the outlier detection for every value of γ. If this additional test, placed before Step 2, concludes that the cluster representative is an outlier, that is, $l_c = 1$ then Steps 2 to 6 are skipped. This test consists in checking whether the proximity from the cluster representative to the nearest neighbour d_2 is the highest of all consecutive differences in d_l, $\{\triangle d_l\}_{l=2}^{N}$, calculated as $\triangle d_l = |d_l - d_{l-1}|$. Note that this constitutes a sufficient condition for the cluster representative to be an outlier since it implies that the considered point is not similar to its nearest neighbour. This issue is explained further in [1].

2.3 Clustering Algorithm

In the previous section we have presented a criterion to determine the cluster to which a given data point belongs. We describe now an algorithm based on the designed criterion to carry out the clustering of the whole data set, whose flow diagram is presented in Fig. 3.

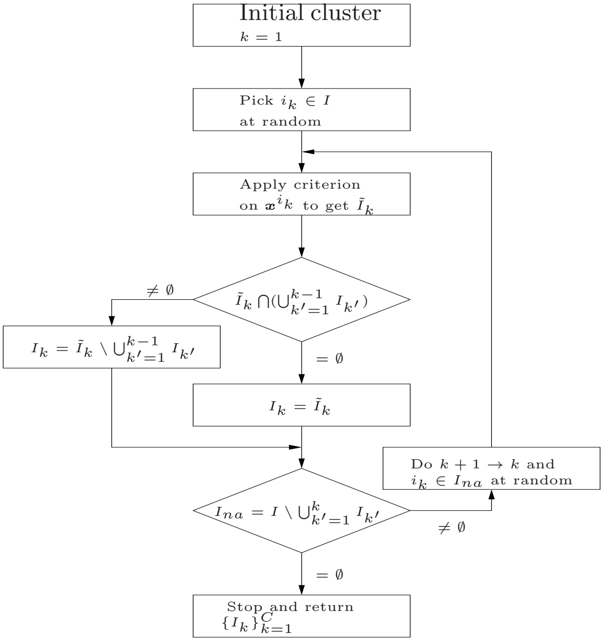
Initial cluster
$k = 1$
Pick $i_k \in I$ at random
Apply criterion on $\boldsymbol{x}^{i_k}$ to get $\tilde{I}_k$
$\tilde{I}_k \bigcap (\bigcup_{k'=1}^{k-1} I_{k'})$
$\neq \emptyset$
$I_k = \tilde{I}_k \setminus \bigcup_{k'=1}^{k-1} I_{k'}$
$= \emptyset$
$I_k = \tilde{I}_k$
Do $k + 1 \rightarrow k$ and $i_k \in I_{na}$ at random
$I_{na} = I \setminus \bigcup_{k'=1}^{k} I_{k'}$
$\neq \emptyset$
$= \emptyset$
Stop and return $\{I_k\}_{k=1}^{C}$

Fig. 3. Flow diagram of CHIDID

1. Randomly choose a point, $\boldsymbol{x}^{i_k}$, from the data set, that has not been allocated to a cluster.
2. Apply the clustering criterion *over the whole data set* to determine the cluster associated to $\boldsymbol{x}^{i_k}$.
3. In case that the proposed cluster, $\tilde{I}_k$, contains any points already included in any of the $k-1$ previous clusters, delete them from the proposed cluster. The points form the definitive k^{th} cluster given by the index vector I_k.
4. Form the non-allocated index vector $I_{na} = I \setminus (\bigcup_{k'=1}^{k} I_{k'})$. If $I_{na} \neq \emptyset$, pass to next cluster by randomly choosing a point $i_{k+1 \to k}$ from I_{na} and go back to Step 2. Otherwise stop the algorithm and return $\{I_k\}_{k=1}^{C}$, which constitutes the solution described in (2).

Let us go back to the example in Fig. 1 to illustrate the algorithm. Step 1 picks at random a point to initiate the clustering process. Apply Step 2 to find the associated cluster to the selected point. Step 3 deletes no points. In Step 4, $I_{na} \neq \emptyset$, since I_{na} contains points corresponding to the four clusters yet to be discovered. Therefore another point is selected at random among those in I_{na}. We repeat Steps 1, 2 and 3, but this time Step 3 checks whether the actual cluster intersects with the previous one or not, after which the second cluster is formed. The latter procedure is repeated until all points have been allocated to their respective clusters.

As clusters can be arbitrarily close, it is possible to have points from different clusters within a proximity d_{l_c} from the cluster representative. In such cases, the algorithm is likely to merge all these natural clusters together in a single output cluster. The aim of Step 3 is to ensure that formed clusters are pairwise disjoint. This is achieved by removing from the current cluster those points already allocated in previously found clusters. However, it must be highlighted that a merge is a very unlikely event. Firstly, a merge can only happen if the representative is located in the cluster region closest to the neighbouring cluster. Even in low dimensions, few points have this relative location within a cluster. Therefore, the likelihood of selecting one of these problematic representatives is also low. Secondly, if neighbouring clusters present a significant difference in the value of just one component then a merge cannot occur. Thus, while it is strictly possible, the likelihood of having a merge quickly diminishes as the dimensionality increases.

CHIDID is a method aiming at full dimensional clustering (ie. all components exhibit clustering tendency). It is expected to be able to handle data containing a few components without clustering tendency. However, the contribution of these irrelevant components to the proximity measure must be much smaller than that of the components with clustering tendency. In this work, we restrict the performance tests to data sets without irrelevant components. This issue will be discussed further when generating the test data sets in Sect. 3.2.

Since α and γ are just thresholds, the only input to the algorithm is the data set containing the population. No a priori knowledge is required. The output is the assignment of data points (solutions) to clusters (optima or high performance regions), whose number is automatically found. The performance of the algorithm does not rely on finding suitable values for the parameters. A wide range of threshold values will provide the same clustering results if clusters and outliers are clearly identificable. The algorithm naturally identifies outliers as byproduct of its execution. Within this approach, an outlier is found whenever the additional outlier test is positive or the criterion determines that the cluster representative is actually the only member of the cluster.

CHIDID carries an inexpensive computational burden. It is expected to run quickly, even with large high dimensional data sets. There are several reasons for this behaviour. First, the algorithm only visits populated regions. This constitutes a significant advantage in high dimensional spaces. Second, it scales linearly with M, the dimensionality of the space. Finally, it only calculates C times N distances between points, where C is the number of found clusters and outliers.

3 Generation of Test Data Sets

3.1 Synthetic Data Sets

These data sets emulate the output of an ideal search and optimisation method on a multimodal optimisation problem. The assumed structure is most of data points distributed into clusters, with a small subset points as outliers.

The generation of these synthetic data sets is beneficial for two reasons. First, emulated outputs can be designed with very diverse characteristics. This is important because, in practice, there is not a unique form for these outputs. Second, it allows us to test the clustering algorithm with data sets of higher dimensionality than those achievable by current search methods.

It is reasonable to assume that each cluster of solutions is confined within the basin of attraction of a different function minimum (we restrict to minimisation without loss of generality). The Rastrigin function is suitable to generate such synthetic outputs because the location and extent of its minima is known. It is defined as

$$f(\boldsymbol{x}) = MA + \sum_{j=1}^{M} (x_j^2 - A\cos(2\pi x_j)) \ . \tag{4}$$

This function is highly multimodal. It has a local minimum at any point with every component x_j taking an integer value k_j and a single global minimum, at $\boldsymbol{x} = \mathbf{0}$. The function has a parabolic term and sinusoidal term. The parameter A controls the relative importance between them (we set $A = 10$).

We generate the first synthetic data set (SD1) as follows. It contains $N_c = 100$ data points distributed into clusters, and $N_o = 10$ outliers with each component selected at random within $[-2.5, 2.5]$. The total number of points is $N = N_c + N_o$ and the number of variables is $M = 20$. Points are distributed among five clusters in the following way: $N_1 = 0.4N_c$, $N_2 = N_3 = 0.2N_c$ and $N_4 = N_5 = 0.1N_c$. Each cluster is associated with a minimum, whose components are picked at random among $c_j^k \in \{-2, -1, 0, 1, 2\}$. The basin of attraction of the $\boldsymbol{c}^k$ minimum is defined by the interval $[c_j^k - 0.25, c_j^k + 0.25]$. The size of the interval is given by half the distance between consecutive function minima. Point components in each cluster are generated at random within the basin of attraction of the corresponding minimum.

In addition, we want to consider data sets with very diverse characteristics to check the performance robustness of the clustering method. We start by describing an automatic way to construct data sets. This is based on the previous data set, but with two variants. First, each point component is now generated within $[c_j^k - \triangle x, c_j^k + \triangle x]$, where $\triangle x$ is a random number between $(0, 1/2)$. Second, the number of points of each cluster N_k is now determined as follows. We first determine r_k, which controls the proportion of points in cluster k, r_k is drawn from a uniform distribution in the range $(1, 5)$. Next, we determine the number of points N_k in cluster k using the formula $N_k = N_c r_k / \sum_{k'=1}^{C} r_{k'}$, where C is the number of generated clusters. Note that each of these data sets contain clusters of different sizes, cardinalities, densities and relative distances between them.

Finally, we define eight groups of data sets from all possible combinations of $N_c = 100, 1000$; $M = 10, 100$ and $C = 3, 7$. For all of them we add $N_o = 10$ randomly generated outliers within $[-2.5, 2.5]$. For each group (N_c, M, C), we create ten realisations of the data set. We will refer to these data sets groups as SD2 to SD9. All these data sets are summarised in Table 1.

Table 1. Test data sets. N_c is number of points in clusters, M is the dimensionality and C is the number of clusters. In the real data sets, RD1 and RD2, C is unknown.

Set	SD1	SD2	SD3	SD4	SD5	SD6	SD7	SD8	SD9	RD1	RD2
N_c	100	100	100	100	100	1000	1000	1000	1000	177	203
M	20	10	100	10	100	10	100	10	100	5	7
C	5	3	3	7	7	3	7	3	7	?	?

3.2 Real Data Sets

We generate two real data sets (RD1 and RD2) by applying a search method on a analytical function with four global minimum. This function is used in [5] and defined as

$$f(\boldsymbol{x}) = A\left[1 - \sum_{k=1}^{C_G} \exp\left(\frac{x_1 - a^k}{\sigma_k^2}\right)\right] + B\,\frac{d(\boldsymbol{x}, \boldsymbol{b}(x_1))}{\sqrt{M}} \quad . \tag{5}$$

with

$$\{\sigma_k\}_{k=1}^{C_G} = 0.5 \qquad a^k = \begin{cases} 2 & \text{if } k = 1 \\ 5 & \text{if } k = 2 \\ 7 & \text{if } k = 3 \\ 9 & \text{if } k = 4 \end{cases} \qquad \boldsymbol{b}(x_1) = \begin{cases} \mathbf{2} & \text{if } 0 < x_1 \le 4 \\ \mathbf{5} & \text{if } 4 < x_1 \le 6 \\ \mathbf{7} & \text{if } 6 < x_1 \le 8.5 \\ \mathbf{9} & \text{if } 8.5 < x_1 \le 10 \end{cases}$$

where $d(\cdot,\cdot)$ is the euclidean distance, $C_G = 4$ (number of global minima), M is the dimensionality, $A = 5$ and $B = 1$. As a search method, we use a Real-parameter Genetic Algorithm (GA) [3] [4] that has been modified [5] to be effective at finding multiple good minima. In brief the details are: a steady-state population of size N, parents are selected randomly (without reference to their fitness), crossover is performed using a self-adaptative parent-centric operator, and culling is carried out using a form of probabilistic tournament replacement involving NREP individuals.

The output of this GA is formed with all individuals that enter the population during the run. Thus, many of them are not optimised. From the clustering point of view, these points contain many irrelevant components which harm the clustering. Inevitably, one cannot find clusters where there are not clusters to find. Therefore, a preprocessing is needed to include most of the points with clustering tendency. The preprocessing consists in defining a threshold value for the objective function and include in the data set all points in the outputs below that threshold. The chosen threshold is $f(\boldsymbol{x}) < 1$.

The first data set, RD1, is obtained by applying the GA (N= 100 and NREP= 30, using $40,000$ function evaluations) on the 5-dimensional instance of the function. RD2 comes from the application of GA (N= 150 and NREP= 40, using $200,000$ function evaluations) on the 7-dimensional instance of the function.

4 Analysis of the Results

In the experiments performed in this study, we set the resolution of the clustering as $\alpha = 5$ and the threshold for outlier detection as $\gamma = \frac{2}{3}$. We required only one algorithm run for each data set to provide the presented results. These experiments were performed in an Intel Pentium IV 1.5GHz, with memory of 256MB.

Firstly, the clustering was performed on all the synthetic data sets described in the previous section (SD1 to SD9). The correct clustering was achieved for all of them, meaning that the correct assignment of individuals to clusters was found. Outliers were also identified correctly. No merge, as defined in Sect. 2.3, occurred. Also, due to their linkage to cluster properties, a single choice of parameters was valid for all experiments. The high variety of test data sets demonstrates the robustness of the method. The highest CPU time used was 0.8 seconds (with SD9).

Figure 4 provides a visualisation of the obtained clustering quality with SD1. The five clusters in the data set are correctly revealed by CHIDID, which has correctly assigned data points to clusters as well as discard the outliers.

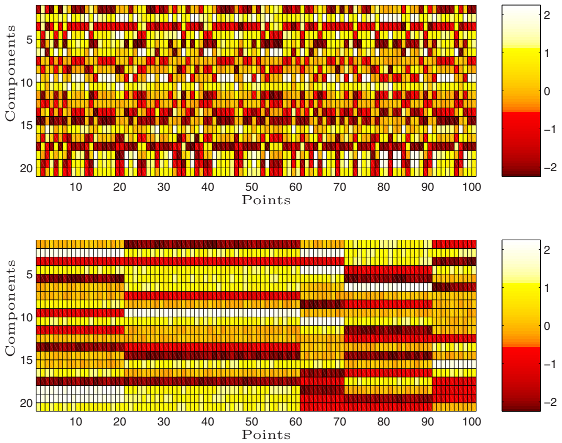

Fig. 4. Results for SD1. The upper plot shows the input data set without outliers, with colors corresponding to the component value. The bottom plot shows the five clusters found with CHIDID (outliers were correctly identified and are not included)

Likewise, Fig. 5 shows equally good performance on SD5, a data set with the same number of points as SD1 but a much higher dimensionality (M= 100 instead of M= 20).

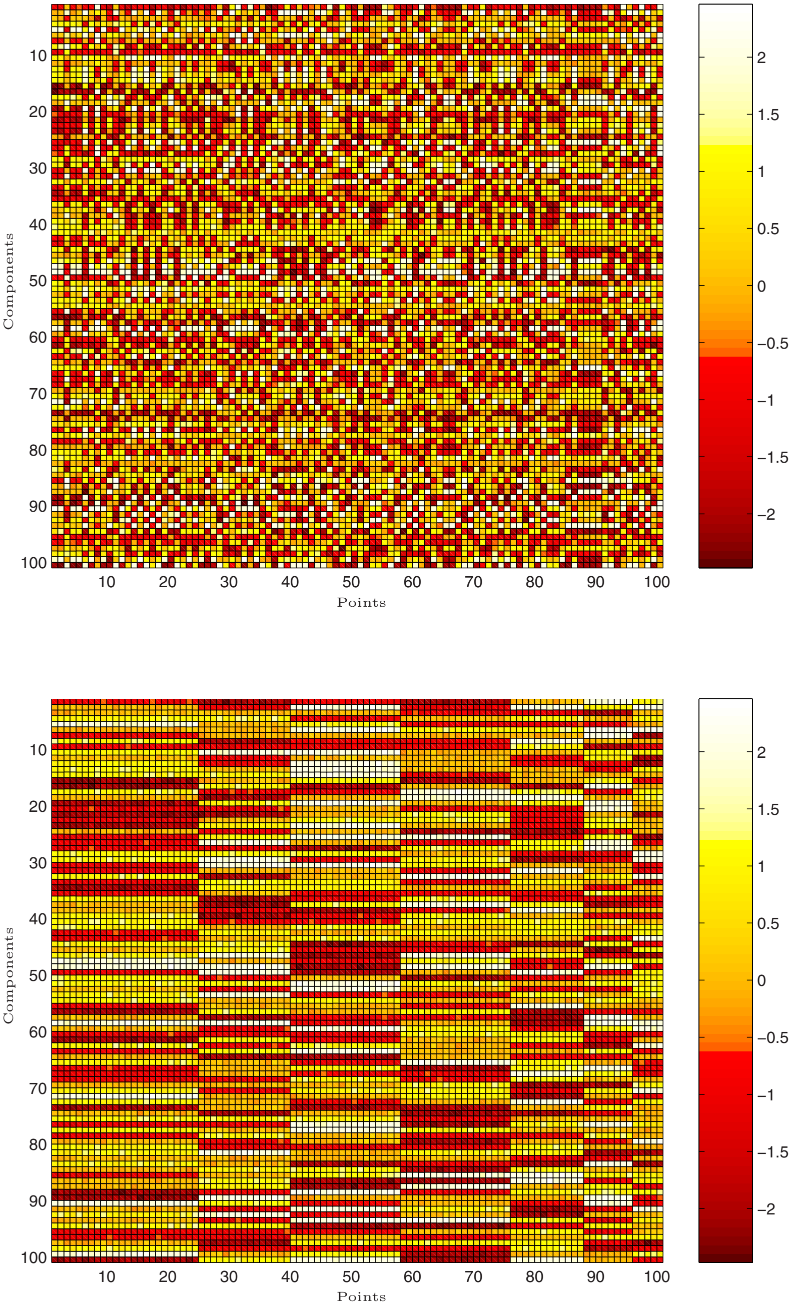

Fig. 5. Results for SD5. The upper plot shows the input data set without outliers, with colors corresponding to the component value. The bottom plot shows the seven clusters found with CHIDID (outliers were correctly identified and are not included)

Finally, we carry out clustering on the real data sets. Figures 6 and 7, respectively. In both cases, four clusters were found, each of them associated with a different global optimum. The quality of the clustering is clear from the figures.

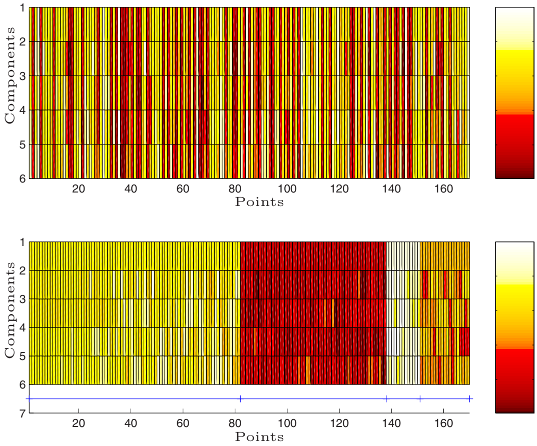

Fig. 6. Results for RD1. The upper plot shows the input data set without outliers, with colors corresponding to the component value. The bottom plot shows the four clusters found with CHIDID (outliers were correctly discriminated and are not included). The horizontal line underneath the plot marks the separation between clusters

5 Conclusions

This paper proposes a new algorithm (CHIDID) to identify clusters of solutions in Multimodal Optimisation. To use this algorithm it is not necessary to provide a priori knowledge related to the expected clustering behaviour (eg. cluster radius or number of clusters). The only input needed is the data set to be analysed. The algorithm does contain two tuning parameters. However, extensive experiments, not reported in this paper, lead us to believe that the performance of the algorithm is largely independent of the values of these two parameters, which we believe are suitable for most data sets and are used in this work. The output is the assignment of data points to clusters, whose number is found automatically. Outliers are identified as a byproduct of the algorithm execution.

CHIDID was tested with a variety of data sets. Synthetic data sets were used to study the robustness of the algorithm to different number of points, variables and clusters. Also, real data sets, obtained by applying a GA on an analytical function, were considered. CHIDID has been shown to be efficient and effective at clustering all these data sets.

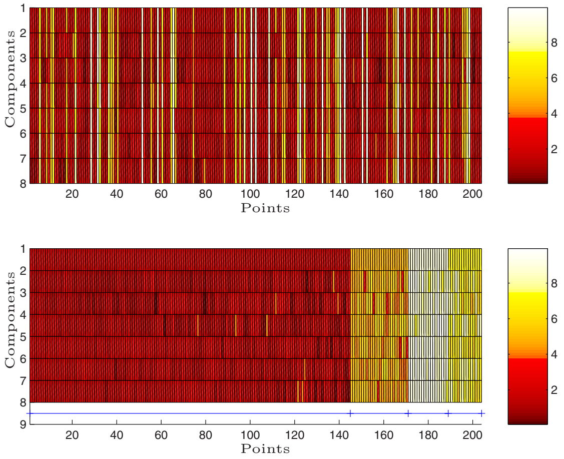

Fig. 7. Results for RD2. The upper plot shows the input data set without outliers, with colors corresponding to the component value. The bottom plot shows the four clusters found with CHIDID (outliers were correctly discriminated and are not included). The horizontal line underneath the plot marks the separation between clusters

References

1. P. J. Ballester. *The use of Genetic Algorithms to Improve Reservoir Characterisation.* PhD thesis, Department of Earth Science and Engineering, Imperial College London, 2004.
2. P. J. Ballester and J. N. Carter. Method for managing a database, 2003. UK Patent Application filed on 5th August 2003.
3. P. J. Ballester and J. N. Carter. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. In *Genetic and Evolutionary Computation Conference, Lecture Notes in Computer Science 2723*, July 2003.
4. P. J. Ballester and J. N. Carter. An effective real-parameter genetic algorithms for multimodal optimization. In Ian C. Parmee, editor, *Proceedings of the Adaptive Computing in Design and Manufacture VI*, April 2004. In Press.
5. P. J. Ballester and J. N. Carter. Tackling an inverse problem from the petroleum industry with a genetic algorithm for sampling. In *the 2004 Genetic and Evolutionary Computation COnference (GECCO-2004), Seatle, Washington, U.S.A.*, June 2004. In Press.
6. R. Bellman. *Adaptive control processes : a guided tour.* Princeton University Press, 1961.
7. M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In *Proceedings Thirteenth International Conference on Scientific and Statistical Database Management*, pages 3–22. IEEE Computer Soc, Los Alamitos, 2001.
8. I. S. J. Packham and I. C. Parmee. Data analysis and visualisation of cluster-oriented genetic algorithm output. In *Proceedings of the 2000 IEEE International Conference on Information Visualization*, pages 173–178. IEEE Service Center, 2000.

A Outlier Detection

In Sect. 2.2, we introduced the parameter γ as a threshold aimed at discriminating outliers. We chose to fix $\beta = \gamma d_2$ with $0 < \gamma < 1$. In this section, we justify these settings.

The relative linear density was defined as $\theta_l = \frac{l}{N}\frac{\beta+d_N}{\beta+d_l}$. Note that in case $\gamma > 1$, β might be greater than d_l and hence the density might be notably distorted. Thus, γ is restricted to the interval $(0,1)$ to ensure the accuracy of the density measure.

Next, we develop the relative linear density increment as

$$\triangle\theta_l \equiv |\theta_l - \theta_{l-1}| = \frac{\beta + d_N}{N}\left|\frac{l}{\beta + d_l} - \frac{l-1}{\beta + d_{l-1}}\right| \ . \tag{6}$$

and evaluate the resulting expression for its first value

$$\triangle\theta_2 = \frac{\beta + d_N}{N}\left|\frac{2}{\beta + d_2} - \frac{1}{\beta}\right| \ . \tag{7}$$

From the latter equation, if $\beta \to 0$ (ie. $\gamma \to 0$) then $\triangle\theta_2 \to +\infty$ and thus $\triangle\theta_2$ will be the highest of all $\{\triangle\theta_l\}_{l=2}^{N}$. This would make the representative to appear always as an outlier. By contrast, if $\beta \to d_2$ (ie. $\gamma \to 1$) then $\triangle\theta_2 \to 0$ and thus $\triangle\theta_2$ will be the lowest of all $\{\triangle\theta_l\}_{l=2}^{N}$. This situation corresponds to the representative being never an outlier.

We next substitute $\beta = \gamma d_2$ in (7) to give

$$\triangle\theta_2 = \frac{\gamma d_2 + d_N}{N d_2} f(\gamma), \ \text{with } f(\gamma) \equiv \frac{1-\gamma}{(1+\gamma)\gamma} \ . \tag{8}$$

Since $0 < \gamma < 1$, it is clear that $f(\gamma)$ controls the tendency of regarding a representative as an outlier. Figure 8 presents the plot of $f(\gamma)$ against γ. The figure shows that a higher value of γ implies a higher tendency to include outliers in clusters.

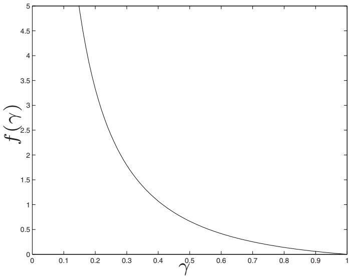

Fig. 8. Plot of $f(\gamma)$ against γ