Research Article

Crescenzio Gallo* and Vito Capozzi

A Wafer Bin Map "Relaxed" Clustering Algorithm for Improving Semiconductor Production Yield

https://doi.org/10.1515/comp-2020-0175
Received Feb 10, 2020; accepted May 05, 2020

Abstract: The semiconductor manufacturing process involves long and complex activities, with intensive use of resources. Producers compete through the introduction of new technologies for increasing yield and reducing costs. So, yield improvement is becoming increasingly important since advanced production technologies are complex and interrelated. In particular, Wafer Bin Maps (WBMs) presenting specific fault models provide crucial information to keep track of process problems in semiconductor manufacturing. Production control is often based on the "judgement" of expert engineers who, however, carry out the analysis of map templates through simple visual exploration. In this way, existing studies are subjective, time consuming, and are also limited by the capacity of human recognition. This study proposes a network-based data mining approach, which integrates correlation graphs with clustering analysis to quickly extract patterns from WBMs and then bind them to manufacturing defects. An empirical study has been conducted on real production data for validating the proposed clustering algorithm, which showed a perfect correspondence between the malfunction patterns found by the algorithm and those discovered by human experts, so confirming the validity of our approach in its ability of correctly identifying actual defective patterns to help improving production yield.

Keywords: Semiconductor manufacturing, Yield management, Clustering, Wafer Bin Maps

1 Introduction

Semiconductor production involves lengthy and complex processes, employing a significant amount of resources. Manufacturers contend by means of increasing research on new technologies to improve integrated circuit (IC) production yield and to reduce manufacturing costs [1–3].

Yield analysis is an activity common to all manufacturing companies of semiconductor devices on an industrial scale. It consists of identifying problems related to the production cycle by analysing the number and distribution of defective chips on a wafer [4, 5]. This is an extremely expensive activity from the point of view of the employed human resources because it requires the visual exploration of thousands of wafer maps (graphical representations of the distribution of defective chips on the silicon wafers produced) and analysis of a huge amount of (not only) electrical related data [6–8].

In any case, the objective is to increase production yield. This is accomplished in two phases: a) identifying yield "detractors"; b) changing the design processes, or modifying the equipment (maintenance, replacement) to eliminate these detractors from the production line.

On the other hand, yield (percentage of chips working on a wafer) analysis is absolutely essential, because it is linked directly to the marketing and the potential profit margin of produced devices. Schematically, this activity can lead to a problem of recognition of failure patterns in a high noise environment, or how to bind the data to different wafers based on patterns that are not known "a priori" (search for unknown problems, or clustering).

In the past, this was done typically by means of traditional statistical methods [9, 10], which are not much efficient in detecting failure patterns related to production line drawbacks. On the other hand, today there are several "intelligent" machine learning algorithms which are much more suited to analyze wafer maps in order to discover their failure patterns [3, 11–13]. In particular, Kang et al. [6] follow a data-driven approach which simplifies the analysis process because no domain knowledge is required. They consider the die positions in a wafer to char-

***Corresponding Author: Crescenzio Gallo:** Università di Foggia – Dipartimento di Medicina Clinica e Sperimentale, Foggia, Italy; Email: crescenzio.gallo@unifg.it
Vito Capozzi: Università di Foggia, Dipartimento di Medicina Clinica e Sperimentale, Foggia, Italy and Istituto Nazionale di Fisica Nucleare – Sezione di Bari, Italy; Email: vito.capozzi@unifg.it

acterize the failure patterns (and thus the yield) related to the wafer center.

Clustering methods

The method of analysis presented in this paper is part of a broader set of clustering algorithms, which aim at grouping elements of a dataset into homogeneous groups (clusters) in such a way that elements belonging to the same group are similar or related to each other, and elements belonging to different groups are dissimilar or not related to each other [14].

Clustering methods can be divided into basic types according to the logic of clustering construction:

- *Partitioning methods*: they divide the dataset into a fixed and known k number of non-empty clusters. They are applicable to medium-sized datasets. The most representative is the k-means clustering [15].
- *Hierarchical methods*: they derive multiple cluster subdivisions, exploit the tree structure and use different threshold values within each cluster and inhomogeneity thresholds between distinct clusters [16].
- *Density-based methods*: they use local density for one observation, so that it considers not less than a prefixed number of neighbors (established by the decision maker); they are able to isolate outliers [17].
- *Graph-based methods*: they use graph partitioning algorithms, and use a scattered representation of the initial dataset [18].

A second breakdown concerns the way in which observations are allocated to individual clusters:

- *Exclusive assignment*: each observation is assigned to a single cluster.
- *Soft attribution*: each observation can belong to several clusters with different degrees of belonging.
- *Full attribution*: Each observation is assigned to at least one cluster.
- *Partial assignment*: some observations may not be assigned to any cluster; they are very useful to identify the presence of outliers in the dataset.

Most clustering methods are heuristic in nature, generating good quality clusters but it is difficult to speak of "optimality". An exhaustive method for the subdivisions of m observations into k clusters requires examining a (prohibitive) number of solutions [17]. It is therefore necessary to use algorithms that do not follow a merely exhaustive approach; they should rather take into account the structure of the dataset in order to perform clustering efficiently and effectively. Beyond these considerations, clustering models are based on a measure of similarity (or distance) between observations. By assigning a dataset D consisting of m observations, it is possible to represent each observation using an n-dimensional vector, where n represents the number of attributes measured for each observation. We can then represent the dataset through a rectangular matrix $X = [m \times n]$ and compute the square symmetrical matrix $S = [m \times m]$ where each element (i, k) represents the similarity (or distance) between the i-th and the k-th element. The definition of an appropriate notion of similarity or distance depends on the nature of the attributes that are measured in correspondence to the elements that belong to the dataset to be analyzed (e.g. Euclidean, Manhattan, Mahalanobis distance, etc. [17]).

It is usual to evaluate clustering models by calculating certain performance indicators. One of the most common is the silhouette coefficient [19], which combines the measurement of intra-cluster cohesion with the measurement of inter-cluster separation, varying in the range $[-1, +1]$: positive values close to one are an index of ideal clustering. The overall silhouette coefficient is computed as the average of the coefficients of the individual observations of the dataset.

The interpretation of the large amounts of data produced by semiconductor manufacturing requires new efficient strategies to reduce the sizes involved. Clustering algorithms typically group wafers into clusters of similar profiles in order to identify possible functional relationships between them. Similar problems are found in the analysis of large networks, where the subnets that meet certain criteria can be extracted (such as the search for web pages on the same subject) [20].

The clustering approach for discovering defective patterns in Wafer Bin Maps is a widely covered approach in literature. Kim et al. [21] analyse the detection and separation of mixed-type defect patterns into clusters, with the aim of matching each cluster to a well-known defect type or discovering a new defect pattern. In this paper, authors propose the connected-path filtering method to remove "noise" from WBMs; then the infinite warped mixture model is adopted for the clustering of mixed-type defect patterns. Chia-Yu Hsu [22] uses "clustering ensemble", in which Wafer Bin Map pattern extraction is made in order to recognize systematic defect patterns efficiently. This approach integrates data transformation with k–means and particle swarm optimization (PSO) clustering algorithms, assessing results through adaptive response theory (ART) neural networks. In [23] Hang Dong *et al.* analyse the spatial patterns of defective chips on Wafer Bin Maps, proposing a wafer yield prediction model that incorporates the spatial clustering information in functional test-

ing. Liukkonen and Hiltunen [24] combine Self-Organizing Map (SOM) network and k–means clustering algorithm for analysing systematic defect patterns on spatially oriented wafer maps.

Beyond the above clustering methods, we propose a method which belongs to the class of "graph based" clustering methods. It performs a non-exclusive (each wafer can be assigned to more than one cluster) and partial (some wafers — typically outliers — may not belong to any cluster) attribution. In particular, our method deals with a computationally simple method for clustering defect patterns based on a "linearization" of the positions of the dies on the wafer map surface. This allows a generalization of the analysis, regardless of the individual spatial distribution of defective chips, also mitigating the effects of "excursions" through the monitoring and diagnosis of the causes of failure due to extraordinary events such as incorrect operation, malfunction of machinery or contamination.

2 Materials and Methods

2.1 Wafer manufacturing critical factors

The critical factors that help maintaining a competitive advantage for semiconductor wafer factories include *lowering costs* through the streamlining of production and *increasing yield* through a rapid response to excursions (when process or equipment shift out of specifications) [7, 25, 26]. In particular, the problems which originate defects have to be detected in time and the related causes should be resolved as soon as possible to reduce the large amount of wafers discarded.

In addition to the in-line control for the detection of chip failures, some electrical tests are performed (circuit probe — CP) on the final product. The tests are performed on all devices manufactured; each die undergoes a predetermined ordered series of electrical measurements. As soon as a chip does not pass a particular test it is marked and, except in special cases, it does not undergo any additional testing. Test results can be analyzed at different levels: "die" (for each mold of the wafer the particular "bin" in charge of the event is shown), "wafer" (each wafer is associated with a vector containing, for each test, the number of failed devices), "lot" (an average of the wafer-level data).

A "human" analyzer can recognize as belonging to the same cluster defective slices: a) having the *same set* of failures (electrical conductivity, etc. detected after thermal, mechanical and electrical stresses); b) with the *same spatial arrangement* (e.g. spatially adjacent defects, with various shapes of random or repetitive type); c) in the *same wafer area* (this obviously does not apply well to random failures); d) with *different spatial extensions*.

An operation often performed to investigate the problems of a device is to run, on an adequate number of wafers, a bar graph representing the relative frequencies of electrical failures in descending order, where the bin relative to the functioning chip is removed from the graph. This graph is called "Pareto bin" (Figure 1) and it helps to understand which electrical failures weigh more and should be addressed first to try to understand their causes.

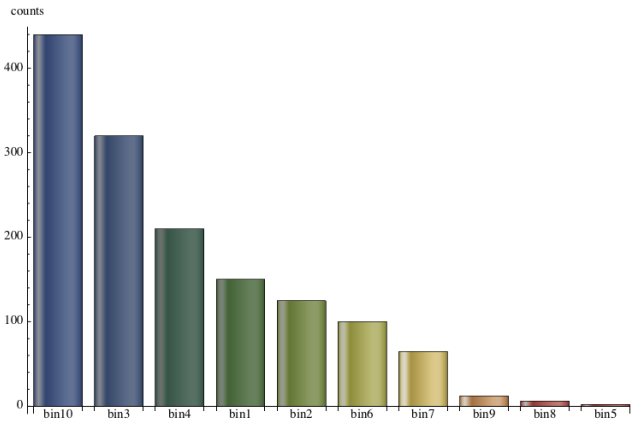


Figure 1: *Example of a Pareto bin diagram representing the relative frequencies of electrical failures.*

This diagram is a strong descriptive tool for random defects. In addition, an electrical failure rarely occurs alone, but very often in conjunction with other ones (e.g. physical corrosion, electrical leakage and shorts). So, in case of problems, a single bin (i.e. a single failure configuration) rarely changes its position in the diagram.

Moreover, *yield learning* is an activity that allows the identification of yield detractors; basically, it is a classification of defects affecting devices. The eventual result of this phase is the compilation of a yield Pareto diagram (see Figure 2), used by engineers to analyze and identify the causes of failure that may influence the yield by means of a detailed list of defects, sorted by the impact that they have on the final result. A considerable amount of operational data of electrical tests [27] forms the basis of the activity of yield learning.

In semiconductor manufacturing there is an important computing need for the fulfillment of an expert system capable of "clustering" [28] (i.e. putting together) wafers with similar sets of failures in similar zones, with similar shapes but with different variable spatial extensions. For

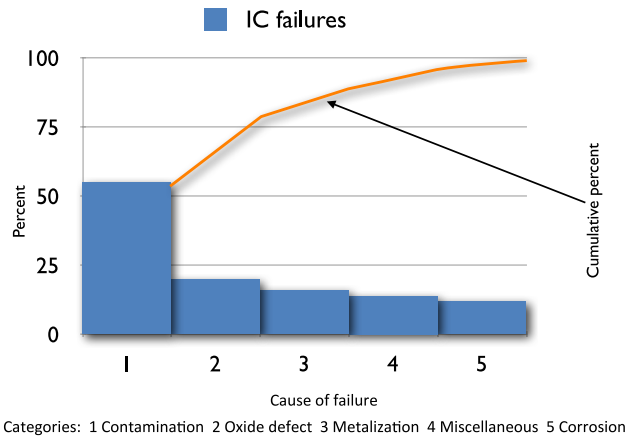


Figure 2: *Example of a yield Pareto diagram.*

example, a "half-moon" pattern can be very small or very large, but it retains the same shape and it is located in the same area of the wafer: this identifies wafers belonging to the same cluster.

These "identified problems" make up an important knowledge base (also known as *rules* or *defect patterns* [23]) useful for dealing with manufacturing problems. Then, in the production line, a wafer's defects can be compared with the set of known defect patterns [29] to verify the possible presence of already known "rules": if the defective wafer does not belong to any known feature (at least with an acceptable degree of confidence), a new defect pattern is submitted to the attention of the analyst who, if that pattern is internally consistent, formalises the recognition of a new yield issue (rule) and inserts it in the database of known issues.

2.2 Wafer Bin Maps

A largely used and useful tool for yield management are the Wafer Bin Maps (WBMs) [4, 13, 28, 30, 31]. Such maps present specific fault models and provide crucial information to keep track of process problems in semiconductor manufacturing. Production monitoring often relies on the "evaluation" of experts who visually explore map templates: such approach leads to a biased judgement, it is time consuming and it is also limited by the ability of human perception.

A Wafer Bin Map is the result of the "circuit probe" check of the dies on the wafer at the end of production. Figure 3 shows an example of a WBM, which is a spatial representation of the (thermal, mechanical and/or electrical) test results, in which "green" devices have passed all tests and other colors represent various types of failures. WBM patterns can provide very useful information to monitor the production process and products [31].

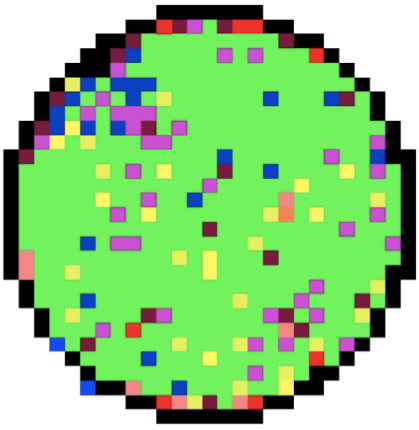


Figure 3: *An example of Wafer Bin Map: green color shows functioning regions, whereas the other colours represent various types of failures (source: https://en.wikipedia.org/wiki/Substrate_mapping).*

Since modern manufacturing facilities are equipped with C.I.M. (Computer Integrated Manufacturing) systems, data collection is no longer an important issue. Moreover, there are also very widespread off-line data analysis systems based on data warehouses [26, 32]. However, it remains the problem of "scouring" the relevant data from the huge mass arising from the analysis of production in order to obtain information that can help analysts in the timely resolution of problems and optimization of yield.

Wafer Bin Maps are multi-dimensional and are equipped with complex spatial structures, able to provide information essential to identify problems in the production process.

Figure 4 shows some examples of WBMs' patterns of defects [30] where different grey levels denote defective chips in different functional tests (dark = working chip, light = defective chip). WBM *defect patterns* can be classified into three main categories [5, 8, 22]:

Systematic defects: the positions of the defective chips in the wafer show the spatial correlation. For example, ring, board, checkerboard patterns. Figure 4 (1–10) shows some systematic patterns often found in factory, as defined by experts.

Random defects: lack of spatial clustering or clearly identifiable patterns. Defective chips are randomly dis-

tributed in the two-dimensional map. Random defects are usually caused by environmental factors in the production process. Even in an almost sterile environment, polluting particles can not be removed completely. However, reducing the level of random defects may improve the overall productivity in the manufacture of wafers.

Mixed defects: they are made of random and systematic defects [21] (last pattern in Figure 4): most wafers have maps of this type. It is appropriate to separate random from systematic defects in WBMs, since systematic defect patterns can reveal a process problem [33].

In some cases, due to the particular spatial distribution of the defects on the surface of the wafer, it is also possible to formulate hypotheses as to which production step may have caused the fault. For example, the *checkerboard*-like (also called *repetitive*) faults are almost certainly due to a lithography process [13], so that the problem may be related to a particular pattern. Often, a view of wafer bin maps of an entire production lot is used, obtaining a noise reduction.

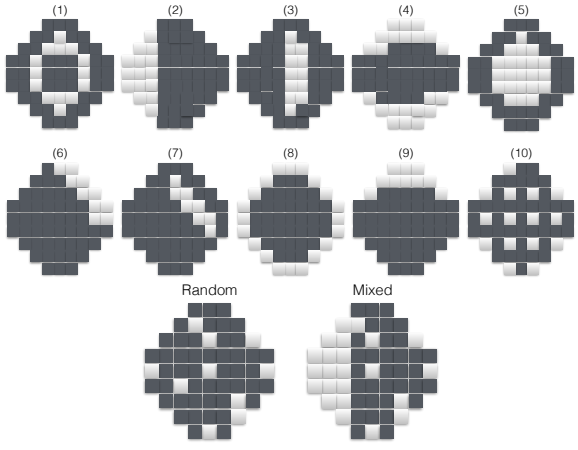


Figure 4: *Examples of patterns of systematic (1–10), random and mixed defects in wafer maps, where dark regions correspond to working chips and light regions to defective chips* (source: Hsu and Chien, 2007).

2.3 Our approach

In this work we present a data mining hybrid approach that integrates cluster analysis and spatial statistics [21–24, 30, 34] to quickly extract patterns from WBMs and associate them with manufacturing defects. The analysis of large amounts of data collected from production and the results can help engineers to make the right decision to classify failure patterns.

Following a networking approach, we have developed a "relaxed" clustering algorithm which allows the rapid extraction of failure patterns in order to associate them with defective manufacturing processes. A practical application of the clustering algorithm has been carried out on real production data for its validation, with encouraging results for a more complete development of an expert system aimed at the automated control of wafer production yield.

The yield data used for testing our algorithm have been acquired from 1817 wafers coming from 78 lots, indicating – for each chip – the relative bin value. Each wafer is divided into 577 chips for a total of 1,048,409 chips, each of which is accompanied by the following information: a) *Lot* Id; b) *Wafer* Id; c) *Die* X/Y coordinates (see Figure 5); d) *Bin* value (see the supplemental file in Section 4, which shows the dataset without the batch references).

Bin values are in the range 0–56, with 0 = working chip and 1–56 corresponding to 56 different failure configurations. Each bin (which was originally a pair of alphanumeric characters representative of a particular combination of chip malfunctions) was encoded as follows:

- value 0 has been assigned to the most frequent bin (which corresponds to the working chips);
- whole numbers 1 to 56 were progressively assigned to the successive bins in the relative frequency scale, so that the value 56 corresponds to the rarest bin (malfunction).

All bin values are sorted by progressive relative frequency; and each chip is assigned the bin value given by the most frequent one from all maps in the corresponding position.

The bi-dimensional coordinate system (Die$X \in [-9, 17]$, Die$Y \in [0, 26]$) was uniquely mapped — for simplicity — to a one-dimensional array (indexed between 1 and 577), while maintaining [24, 34] the concept of "adjacent chips" (see, for example, the two patterns shown in Figure 6) or same electrical failures with different patterns as shown in Figure 7. This *isomorphic* transformation of coordinate systems does not affect the search for patterns because the interrelationship between the single portions of the wafer map is preserved.

For the implementation of the clustering algorithm we run the *Mathematica* 10 and *Orange* data mining software on an Apple macOS server with dual Intel Quad-Xeon and 12GB of RAM. In particular, numerical and matrix manipulation functions, bi- and tri- dimensional graphic packages, graph management libraries and the NeuralNet-

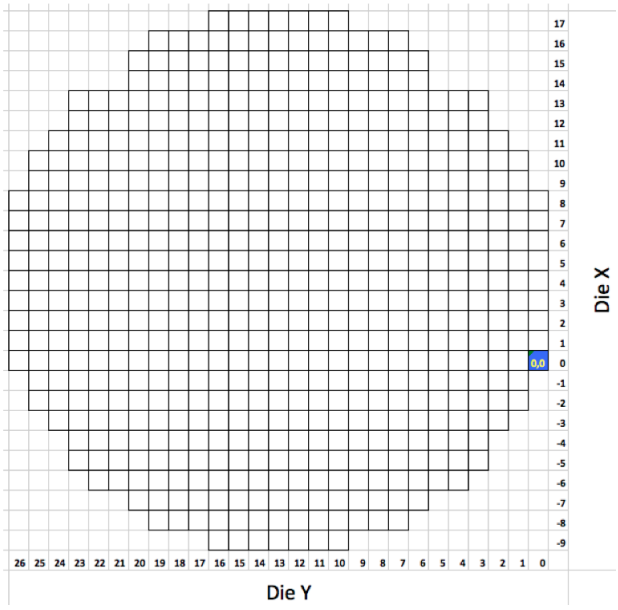


Figure 5: *The real wafer spatial coordinate system on which our model is based, where the four corners do not belong to the pattern.*

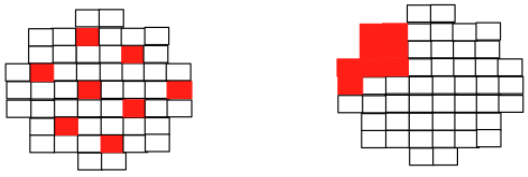


Figure 6: *Some examples of "adjacent" chips (defective chips are shown in red).*

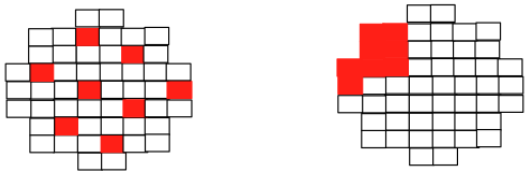


Figure 7: *Examples of electrical failures with different spatial patterns (defective chips are shown in red).*

works module have been used. When possible and necessary, calculations were performed in parallel (up to 8 kernels) using the native parallel processing capacity of *Mathematica*.

3 The Relaxed Clustering Algorithm

The **clustering algorithm steps** are:

1. Read the input bi-dimensional array X made of n rows, one row per wafer, and m columns, one for each chip of the wafer. The array X is obtained from each wafer bin map of the dataset, putting in its (577) columns the bin values corresponding to the wafer chips, according to the system coordinates in Fig. 5 (see also the attached supplemental file, which shows the actual 1817 wafers of the dataset, each with its 577 bin values.)
2. The $n \times n$ correlation matrix C is defined on all wafer (m–dimensional vector) pairs (x_k, x_l), $k, l = 1 \ldots n$, whose elements are computed using the Spearman correlation coefficient ρ, defined as the Pearson correlation coefficient between the ranked variables [35]. The m raw scores for each vector x are converted to ranks r and the correlation coefficient between x_k and x_l is computed as
$$c_{k,l} = 1 - \frac{6 \sum_{j=1}^{m} d_j^2}{(n^2 - 1)n} \in [-1, 1]$$
where $d_j = x_k(j) - x_l(j)$. The coefficient $c_{k,l}$ is discarded if it does not pass the statistical significance test on the random variable $z = \sqrt{\rho(N-1)}$ (the test is applicable [35, 36] since the sample size is greater than 20). Note that the correlation coefficient is applied in absolute value. In fact, in the next step of the algorithm two nodes k and l are connected only if the absolute value of their correlation coefficient $c_{k,l}$ is equal or higher than the threshold θ_1.
3. Let θ_1 be a fixed threshold defining an open interval of symmetrical comparison $(-\theta_1, \theta_1)$ and let us build the adjacency matrix A between wafers whose elements $a_{k,l}$, $k, l = 1 \ldots n$, are defined as follows:
$$a_{k,l} = \begin{cases} 0 & \text{if } c_{k,l} \in (-\theta_1, \theta_1) \\ 1 & \text{otherwise} \end{cases}$$
4. An undirected (correlation) graph G is built from the adjacency matrix, with edge weights equal to the correlation coefficients $c_{k,l}$. This graph has one node for each selected wafer; links between nodes correspond to the presence of a correlation between them (see Figures A5 and A6).
5. A clustering coefficient k_i is associated to each node i belonging to the correlation graph:
$$k_i = \frac{2\alpha}{\gamma(\gamma - 1)} \in [0, 1]$$
where α is the number of triangles (a triangle is a cluster of three directly and fully connected nodes) having node i as one vertex and γ is the number of nodes directly linked to i.

6. Le us fix $\theta_2 \in [0, 1]$ as a clustering threshold. From the graph G we induce a subgraph H made of all nodes $i \in G$ (and related edges e) with *relaxed* clustering coefficient greater than or equal to θ_2: $H = \{\text{nodes } i \in G \text{ and related edges } e : k_i \geq \theta_2\}$. The *relaxed* clustering coefficient of a node i is given by the maximum clustering coefficient among k_i and those of the nodes directly connected to it up to a predetermined maximum distance δ: this gives a "dragging" effect by the surrounding nodes, better taking into account the complex structure of the relations between the graph nodes (wafers) than the simple clustering coefficient of each node.
7. From this subgraph we extract all connected components (made of at least two linked nodes) representing the clusters corresponding to the values chosen for thresholds θ_1 and θ_2.

The above illustrated algorithm is based on correlation coefficients, combining a network analysis technique with the classical clustering for the study of wafer failures. It uses two parameters (*correlation* threshold θ_1 and *clustering* threshold θ_2, as specified in the steps 3 and 6 of the algorithm), resulting in a flexible and effective ability to analyze the sensitivity of the results in response to the particular configuration chosen.

On the contrary, the k–means clustering algorithm [24] requires the *a priori* definition of the number k of clusters to be searched; this is a method that gradually defines clusters by partitioning observations around an average distance between them (centroid). This method is different and not comparable to our algorithm, which in the first phase eliminates the correlations below threshold θ_1 (possibly taking into account only the significant coefficients according to a proper statistical test). Then, in a second phase, our method computes — for each remaining node of the correlation graph — its "relaxed" clustering coefficient, removing those below the threshold θ_2.

In this second phase, our algorithm introduces a variation with respect to the classic method of performing clustering, a variation from which the algorithm takes its name. In fact, the single clustering coefficients are "relaxed" (i.e. made equal) to the value of the maximum clustering coefficient of all the nodes connected — at a prefixed distance δ, usually 1 — to the node under examination. In this way, the "relaxed" clustering algorithm takes into account all nodes connected to a given node up to distance δ; if any of the δ–near nodes has a clustering coefficient greater or equal to the threshold θ_2 the given node will not be eliminated. In essence, there is a *dragging* effect of the nodes with a low clustering coefficient by those with a higher one and directly connected to it, as shown in the examples provided in Section 5.

Our algorithm (which has been also applied, in its simple double threshold form, in a biological context with positive results [37, 38]) is based on the concept of curvature [20, 39] applied to the network (aka *correlation graph*, see Figures A5 and A6 in Section 5) of related wafers, where nodes represent devices and edges represent correlations between them. Typically the Spearman rank correlation coefficient ρ is used because it is not limited to a linear correlation [10] and it is less sensitive to possible high outliers in the tails of both samples. On the contrary, if a normal behavior is observed in data, then it would be preferable to use Pearson's correlation coefficient because it is simpler and less computationally expensive. In our case not all distributions of bin values were normal, so we chose to use Spearman's rank correlation coefficient.

Whichever correlation method is used, it has to be followed by computing each coefficient, and a statistical significance test has to be passed as well. Possible tests for assessing correlations [35, 36] are z–, t– or F–test according to the correlation method chosen. We adopted the z–test (which is based on the random variable $z = \sqrt{\rho(N-1)}$) for normal distributions, and the t–test (which is based on the random variable $t = \rho/\sqrt{(1-\rho^2)/(N-2)}$) for other distributions. These tests are simpler than F–test but still effective for a large sample size ($N >> 20$), which is the case in this type of applications.

The main objective of the algorithm is to verify "in real time" a batch of wafers coming from a specific device of the production line, to test the malfunctioning of that device and intervene on the production as soon as possible. Therefore, the algorithm does not address "big data", but instead it wants to identify with sufficient speed and precision the most important patterns of production defects (clusters) and present them to the production line manager to correct the malfunctions of the devices involved. Nevertheless, the characteristics of the algorithm (subsequent use of the two thresholds, which allow to quickly and progressively discard a high number of nodes not relevant in the correlation graph) make it suitable also for the processing of a high number (hundreds of thousands) of wafers. In particular, in the first phase the (theoretical) computational complexity of the algorithm is $O(n^2)$, having to calculate a (square) correlation matrix of n elements, with n being the number of wafers to be processed. In practice, the correlation matrix is computed and stored in a "sparse" way, which significantly reduces the calculation time and memory occupation. Moreover, after the application of the first threshold, the initial correlation graph is considerably reduced, and for the remaining nodes (which may be at

most n but are normally many less) the order of the clustering index calculation is $O(k \cdot n)$ with $k = \overline{\alpha}/\overline{\gamma}^2$, where $\overline{\alpha}$ is the average number of "triangles" connected to each node and $\overline{\gamma}$ is the average number of nodes directly linked to the node itself.

The wafer bin clusters are the densest regions of the correlation graph, which correspond to defect patterns. It should be noted that the clustering coefficient, introduced in step 5 of our method, is typically extremely low in random graphs that have a small average degree compared to the number of nodes [39]. Clusters of correlation graphs with high clustering coefficients are thus highly non-random structures which are difficult to extract when the nodes are randomly distributed, and this point can be a limitation of our method.

An application of the algorithm is illustrated in Section 5.

The algorithm's "metaphor"

To better understand the operation of our relaxed double-threshold clustering algorithm we can think of the initial correlation graph as a seabed conformation corresponding to a particular θ_1 value. That is, the correlation graph G depends on the chosen θ_1 value and it depicts the correlation structure between nodes (wafers) extracted from the correlation matrix that we can metaphorically associate to a particular seabed conformation.

Threshold θ_2 corresponds figuratively to sea level and clusters are emerging islands. Varying the correlation threshold θ_1 results in changing the seabed landscape (as in the example shown in Figure 8), while the clustering threshold θ_2 moves only the sea level and, consequently, it changes the existence and relative size of clusters.

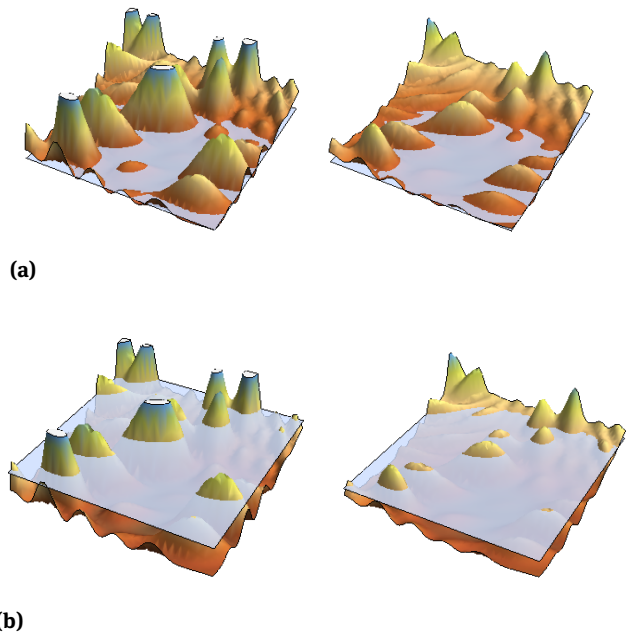


Figure 8: *The "seabed" metaphor. The two landscapes in (a) correspond by analogy to two correlation structures (where peaks represent all possible clusters) obtained with two different θ_1 values; (b) shows the remaining clusters (peaks emerging above sea level) after applying a particular θ_2 value.*

4 Results and Discussion

The two thresholds used in our algorithm are intended to progressively refine the correlation graph (as illustrated in Section 3), gradually eliminating the less closely related wafers. In particular, the first threshold aims to define the initial correlation graph containing the clusters of wafers with a correlation coefficient at least equal to the threshold itself. Once the initial correlation graph is obtained, which contains one or more connected components (i.e. clusters), each of these components is further thinned by eliminating the nodes (wafers) that do not have a clustering index at least equal to the second threshold.

The clustering index in practice represents the capacity of a graph node to be an "aggregation" hub for the other nodes directly connected to it, as it is normally calculated in classic correlation graphs: in our algorithm, instead, the clustering index of a node (wafer) is equal to the maximum clustering index of the nodes "close" to it up to a prefixed maximum distance δ (as shown in the examples in Section 5). This characterizes the concept of "relaxation", allowing to keep nodes (wafers) which — in a traditional clustering algorithm — would be discarded. Moreover, in the present work, unlike the initial biological implementation, the distance δ has been used with a value of at least one, which has increased the "dragging" effect of the neighbor nodes allowing to heuristically calibrate the behavior of the algorithm according to the data under examination.

In particular, the thresholds θ_1 and θ_2 were set empirically; for θ_1 the value 0.6 was used in order to keep the wafers with a Spearman correlation coefficient of at least 60% in the same cluster. Subsequently, a progressive series of values of the threshold θ_2 was applied to the initially obtained clusters (related components of the correlation graph). Further, the nodes with a clustering index greater than the chosen threshold were removed. Eventually, the value of θ_2 (in our case, 0.3) corresponding to the

best overall average silhouette index of the corresponding clustering was considered.

In the clustering process we first aimed to "discover" the spatial distribution of the bins for each wafer, in order to have an empirical proof of the electrical failures distribution. In our specific case, this was accomplished by plotting the bin values of the 27×27 matrix corresponding to a single wafer, where the white squares are the working chips (bin value=0), whereas a higher bin value corresponds to a more marked color for different failure configurations. Figure 9 illustrates an example map of some bin values for all "overlapped" wafers in relation to the position of the chip.

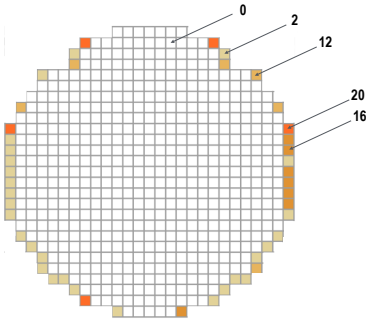


Figure 9: *Some example bin values of the synthetic map derived from a subset of (ideally overlapping) wafer bin maps. White squares (bin value=0) represent the working chips; color squares (bin values 2, 12, 16 and 20) correspond to an example of different failure configurations. The color of a single chip in a particular position is given by the most frequent bin of all "overlapped" wafer bin maps.*

The results of the wafer bin maps obtained for the first 9 bin values are illustrated in Figure 10. In this picture, for each chip the highest frequency of the relative bin value on all the sample's wafer maps is highlighted: this can be considered firstly as overlapping all the wafer maps of the sample with only a particular bin marked, and then observing the overlapped maps from the top.

Figure 11 shows the complete three-dimensional histogram of the frequencies of the bin values for each one of the 577 chips of the reference coordinate system shown in Figure 5.

The clustering performed on the 1817 wafers according to the algorithm illustrated in the previous section (with $\theta_1 = 0.60$ and $\theta_2 = 0.30$) shows the "spatial" regularity of some defective patterns, i.e. homologous sets of electrical failures that occur in same areas of the wafers. The wafer maps shown in Figure 12 were obtained by overlapping the maps of all wafers belonging to the cluster, in order to visually expose the patterns found by the clustering algorithm.

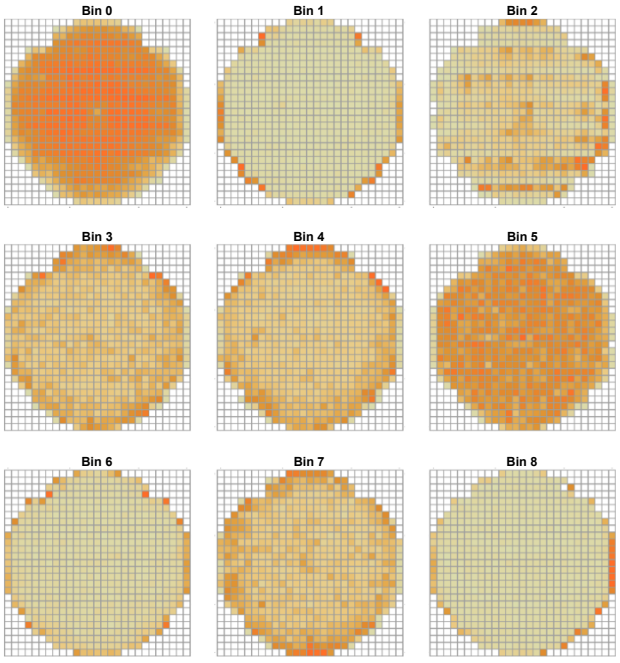


Figure 10: *The wafer bin maps corresponding to the first 9 bin values of the initial dataset from the manufacturing line (the four corners do not belong to the maps).*

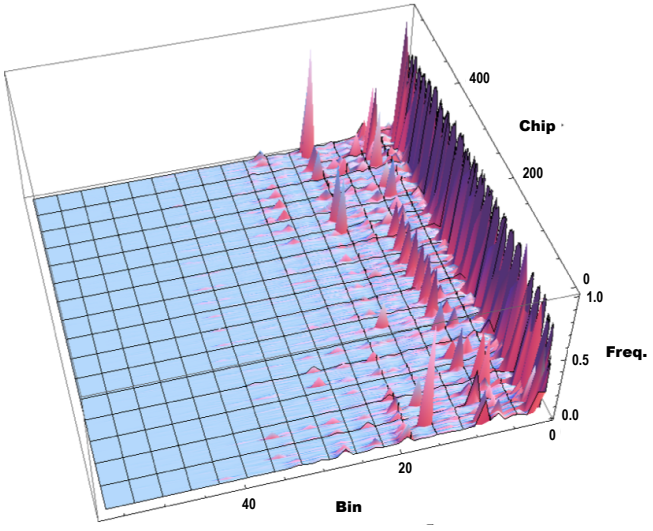


Figure 11: *The frequency distribution of the 57 bin values for each one of the 577 chips in the wafer maps of the initial dataset from the manufacturing line.*

In particular, the maps of the large number of the examined wafers highlight some different configurations of electrical failures or *rules* corresponding to systematic or mixed defects:

– Rule 1 "**ring**": clusters #1 and #4 of Figure 12.

– Rule 2 "**sector**": clusters #2 and #3 of Figure 12.
– Rule 3 "**stripe**": clusters #5 (shaded) and #6 (strong) of Figure 12.

These early findings have been examined and confirmed by the engineers of the factory where the tests have been carried out, and coincided with the results of the visual inspections made by the experts. The above results justify the viability of the "clustering" approach in classifying defect patterns of WBMs.

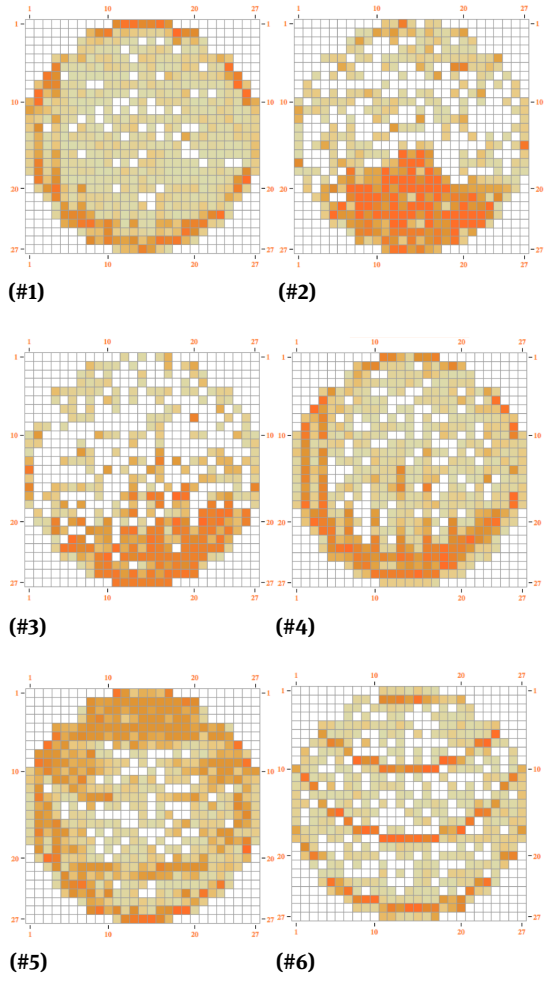


Figure 12: *Figure shows the first six clusters obtained by our relaxed clustering algorithm reported in Section 3 (the four corners do not belong to the patterns). The aggregation function used to obtain the six cluster maps is the "mode"; in fact each chip is colored according to the most frequent bin in its position for all the overlapped wafer bin maps constituting the cluster.*

5 Conclusion

This investigation presents a hybrid clustering algorithm for automatic extraction of failure patterns in the wafer bin maps resulting from the production of wafer semiconductors. The proposed method is able to identify the maps with spatial correlation and it provides, in a short time, useful information to support decisions that improve the production yield.

Compared to the initial version used for the biological analysis of gene correlation networks [37, 38], the proposed algorithm has been applied to the industrial field to control the yield of semiconductor wafers by extending the "double threshold" clustering approach. This method has been extended through the "relaxed" computation of the clustering index, considering a distance of at least 1 "neighbor" nodes (wafers), thus allowing to verify the generalizability of the proposed clustering algorithm.

The results obtained both from the visual analysis of the wafer maps and our "relaxed" clustering algorithm confirm the effectiveness of the applied method. This reinforces the applied methodology, and encourages to carry on a further series of experiments by properly fine-tuning the clustering thresholds in order to optimally calibrate the model.

The future research objective will be to start from the identification of a set of "synthetic" parameters (clusters of defects' patterns), in order to detect similarities among processed wafers by using the results obtained in the direction of the implementation of one (or more) automatic classifiers (neural networks or similar machine learning algorithms). Such an issue is aimed at the recognition of existing "defect rules" in the manufactured wafers or at the discovery of new rules to increase the knowledge base.

It could be then worth exploring the possibility to discover such new configurations (spatial distributions) of Wafer Bin Map failures by processing a large number of maps with a big-data approach, for developing an automated expert system able to identify manufacturing defects in order to facilitate yield management.

References

[1] Chien C.F., Hsu C.Y., Chang K.H., Overall Wafer Effectiveness (OWE): A novel industry standard for semiconductor ecosystem as a whole, Computers & Industrial Engineering, 65, 2013, 117–127, 10.1016.j.cie.2011.11.024
[2] Gardner R., Bieker J., Elwell S., Solving tough semiconductor manufacturing problems using data mining., IEEE/SEMI Advanced Semiconductor Manufacturing Conference 2000., 2000

[3] Soenjaya J., Hsu W., Lee M.L., Lee T., Mining wafer fabrication: framework and challenges, Next Generation of Data-Mining Application, John Wiley & Sons, New York, 2005, 17–40
[4] Chien C.F., Hsu S.C., Chen Y.J., A system for online detection and classification of wafer bin map defect patterns for manufacturing intelligence, International Journal of Production Research, 51(8), 2013, 2324–2338, http://dx.doi.org/10.1080/00207543.2012.737943
[5] Taam W., Hamada M., Detecting spatial effects from factorial experiments: an application from integrated-circuit manufacturing., Technometrics, 35(2), 1993, 149–160
[6] Kang S., Cho S., An D., Rim J., Using wafer map features to better predict die-level failures in final test, IEEE Transactions on Semiconductor Manufacturing, 28(3), 2015, 431–437
[7] Leachman R., Hodges D., Benchmarking semiconductor manufacturing., IEEE Transactions on Semiconductor Manufacturing, 9(2), 1996, 158–169
[8] Stapper C., LSI yield modeling and process monitoring., IBM Journal of Research and Development, 44(2), 2000, 112–118
[9] Montgomery D.C., Introduction to statistical quality control, John Wiley & Sons, 2007
[10] Myers J., Well A., Research Design and Statistical Analysis (2nd ed.), Lawrence Erlbaum, 2003
[11] Arnold N.N.A., Wafer defect prediction with statistical machine learning, Ph.D. thesis, Massachusetts Institute of Technology, 2016
[12] Jang S.J., Lee J.H., Kim T.W., Kim J.S., Lee H.J., Lee J.B., A wafer map yield model based on deep learning for wafer productivity enhancement, in SEMI Advanced Semiconductor Manufacturing Conference (ASMC), 2018 29th Annual, IEEE, 2018, 29–34
[13] Palma F.D., Nicolao G.D., Miraglia G., Donzelli O.M., Process diagnosis via electrical-wafer-sorting maps classification., in Proceedings of the Fifth IEEE International Conference on Data Mining., 2005
[14] Gan G., Ma C., Wu J., Data clustering: theory, algorithms, and applications, volume 20, Siam, 2007
[15] Praveen P., Rama B., An empirical comparison of Clustering using hierarchical methods and K-means, in 2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), IEEE, 2016, 445–449
[16] Dabhi D.P., Patel M.R., Extensive survey on hierarchical clustering methods in data mining, International Research Journal of Engineering and Technology (IRJET), 3, 2016, 659–665
[17] Xu R., Wunsch D.C., II, Clustering. Hoboken, NJ: Wiley/IEEE Press, 6, 2009, 583–617
[18] Yin H., Benson A.R., Leskovec J., Gleich D.F., Local higher-order graph clustering, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, 555–564
[19] Thinsungnoena T., Kaoungkub N., Durongdumronchaib P., Kerdprasopb K., Kerdprasopb N., The clustering validity with silhouette and sum of squared errors, learning, 3(7), 2015
[20] Eckmann J., Moses E., Curvature of co-links uncovers hidden thematic layers in the world wide web., in Proc NatL Acad Sci USA, volume 99, 2002, 5825–5829
[21] Kim J., Lee Y., Kim H., Detection and clustering of mixed-type defect patterns in wafer bin maps, IISE Transactions, 50(2), 2018, 99–111
[22] Hsu C.Y., Clustering ensemble for identifying defective wafer bin map in semiconductor manufacturing, Mathematical Problems in Engineering, 2015, 2015
[23] Dong H., Chen N., Wang K., Wafer yield prediction using derived spatial variables, Quality and Reliability Engineering International, 33(8), 2017, 2327–2342
[24] Liukkonen M., Hiltunen Y., Recognition of Systematic Spatial Patterns in Silicon Wafers Based on SOM and K-means, IFAC-PapersOnLine, 51(2), 2018, 439–444
[25] Leachman R.C., Ding S., Excursion yield loss and cycle time reduction in semiconductor manufacturing, IEEE Transactions on Automation science and engineering, 8(1), 2011, 112–117
[26] Peng C., Chien C., Data value development to enhance yield and maintain competitive advantage for semiconductor manufacturing., International Journal of Service Technology and Management, 4(6), 2003, 365–383
[27] Cunningham S., Spanos C., Voros K., Semiconductor yield improvement: Results and best practices., IEEE Transactions on Semiconductor Manufacturing, 8(2), 1995, 103–109
[28] Wang C.H., Separation of composite defect patterns on wafer bin map using support vector clustering, Expert Systems with Applications, 36, 2009, 2554–2561, 10.1016/j.eswa.2008.01.057
[29] Wang C.H., Recognition of semiconductor defect patterns using spatial filtering and spectral clustering, Expert Systems with Applications, 34, 2008, 1914–1923, 10.1016/j.eswa.2007.02.014
[30] Hsu S.C., Chien C.F., Hybrid data mining approach for pattern extraction from wafer bin map to improve yield in semiconductor manufacturing, Int. J. Production Economics, 107, 2007, 88–103, 10.1016/j.ijpe.2006.05.015
[31] Liu C.W., Chien C.F., An intelligent system for wafer bin map defect diagnosis: An empirical study for semiconductor manufacturing, Engineering Applications of Artificial Intelligence, 26, 2013, 1479–1486, 10.1016/j.engappai.2012.11.009
[32] Chien C., Wang W., Cheng J., Data mining for yield enhancement in semiconductor manufacturing and an empirical study., Expert Systems with Applications, 33(1), 2007, 1–7
[33] Friedman D., Hansen M., Nair V., James D., Model-free estimation of defect clustering in integrated circuit fabrication., IEEE Transactions on Semiconductor Manufacturing, 10(3), 1997, 344–359
[34] Li T.S., Huang C.L., Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing, Expert Systems with Applications, 36, 2009, 374–385, 10.1016/j.eswa.2007.09.023
[35] Zar J.H., Significance testing of the Spearman rank correlation coefficient, Journal of the American Statistical Association, 67(339), 1972, 578–580
[36] Student, An experimental determination of the probable error of Dr Spearman's correlation coefficients, Biometrika, 1921, 263–282
[37] Di Salle P., Colantuono C., Gallo C., Traini A., Frusciante L., Chiusano M., Modeling molecular pathways based on gene expression and social network analyses: an example from Arabidopsis Thaliana., in 56th Annual Congress Società Italiana di Genetica Agraria, Perugia, 2012
[38] Gallo C., Capozzi V., Clustering techniques for revealing gene expression patterns, in Encyclopedia of Information Science and Technology, Third Edition, IGI Global, 2015, 438–447
[39] Watts D., Strogatz S., Collective dynamics of 'small-world' networks., Nature, 393, 1998, 440–442

Appendix – Application examples

1 Clustering performance analysis

Our algorithm is comparable with the classical clustering methods described at the end of Section 1, both from a theoretical (how clusters are obtained) and practical (the obtained clusters) point of view. The classical methods (hierarchical, k–Means) are not generally applicable to the domain of the problem with relevant results, because they have — first of all — to define a priori the number of expected clusters (k–Means) or to obtain the clusters by subsequent agglomeration (hierarchical); therefore, these two classical methods involve a considerable computing effort compared to our method. Further, the classical clustering methods suffer from the inability to "refine" progressively the clusters obtained, since the results are based exclusively on the matrix of distances initially calculated.

The performance of our algorithm was compared with the above mentioned classical methods by computing (with the software tools *Mathematica*[1] and *Orange*[2]) the silhouette coefficients of the clusters obtained from the dataset (see Figures A1-A4).

Figure A2 shows the silhouette plot for the hierarchical clustering obtained with the Ward distance metric and a dendrogram threshold limited to 6 clusters. The related maximum silhouette coefficient is less than 0.2. Figure A3 shows the results from the k–Means clustering with a prefixed number of 6 clusters. The related maximum silhouette coefficient is slightly greater than the one obtained from the hierarchical clustering. In Figure A4 the results of the application of our method are illustrated; in particular, the top 6 clusters were selected, with an overall silhouette index greater than 0.4 (about twice the classical methods).

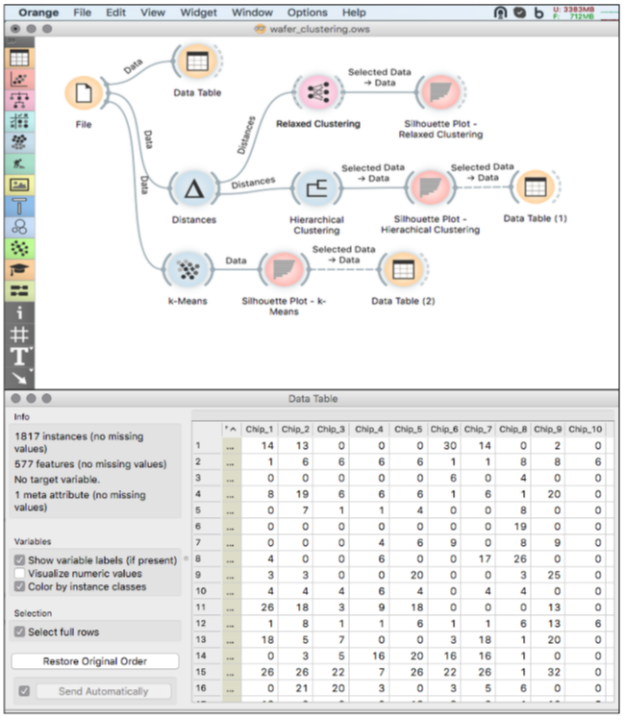


Figure A1: *The Orange workflow for computing the silhouette indexes of the clustering methods (a description and an initial part of the dataset is also shown).*

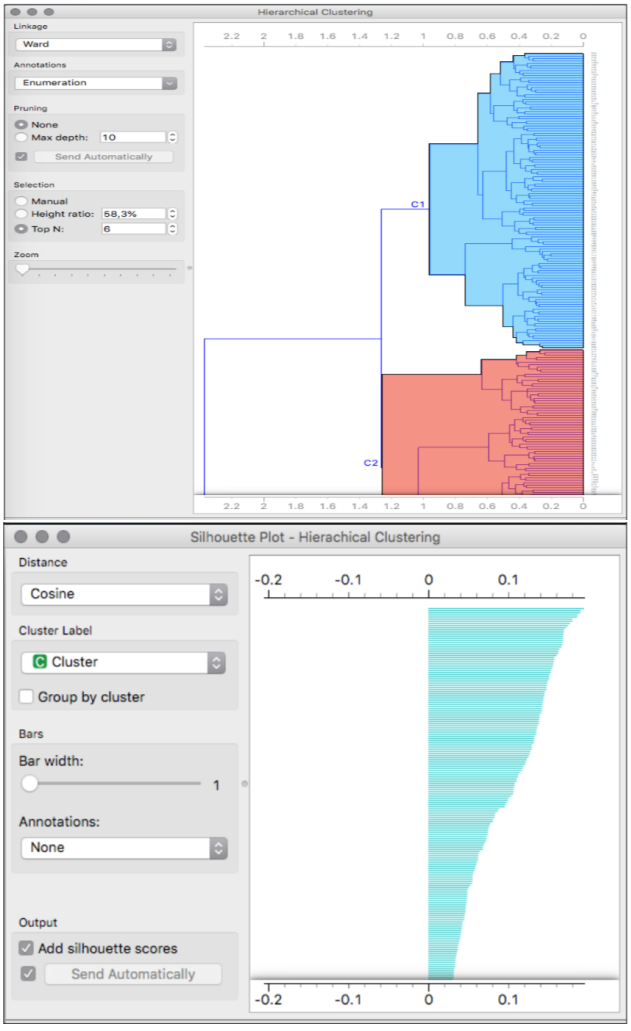


Figure A2: *The silhouette plot of the hierarchical clustering method (linkage method: Ward; forced to 6 clusters; maximum silhouette coefficient less than 0.2).*

1 https://www.wolfram.com/mathematica/
2 https://orange.biolab.si

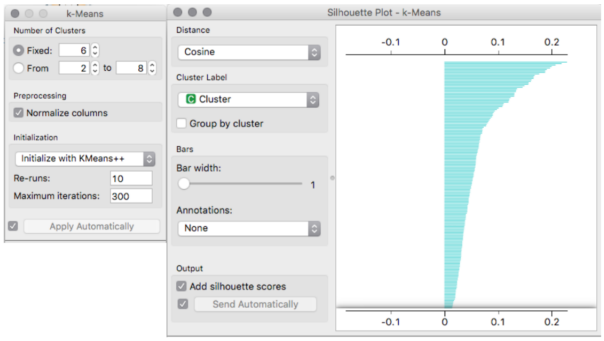


Figure A3: *The silhouette plot of the k-means clustering method (fixed to 6 clusters; maximum coefficient* ≥ 0.2*).*

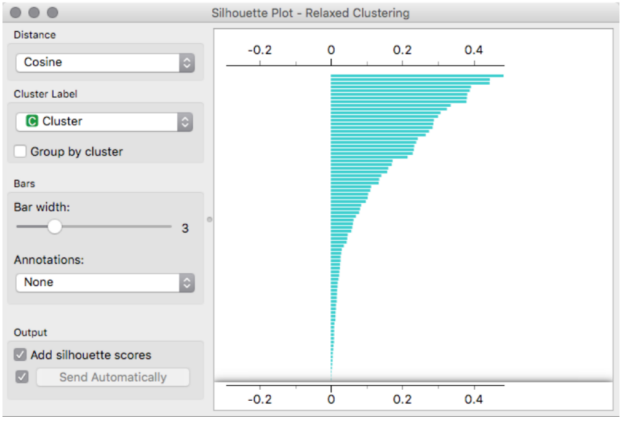


Figure A4: *The silhouette plot of the "relaxed" clustering method (top 6 clusters; maximum coefficient* ≥ 0.4*).*

2 Example 1: A simple example of the "relaxed" clustering algorithm.

Suppose you want to identify the similarity clusters among six given wafer bin maps a–f (which correspond to the nodes in Figure A5), whose bin value sequences make the array X of step 1 of our algorithm from Section 3. From array X we obtain, for example, the correlation matrix C (see step 2 of our algorithm) shown in Table A1.

Table A1: The example correlation matrix C between nodes a–f reported in Figure A5.

	a	b	c	d	e	f
a	–	−0.80	0.35	0.40	0.82	0.74
b	−0.80	–	0.91	−0.92	0.81	0.68
c	0.35	0.91	–	0.77	0.83	0.55
d	0.40	−0.92	0.77	–	0.94	0.18
e	0.82	0.81	0.83	0.94	–	−0.33
f	0.74	0.68	0.55	0.18	−0.33	–

From this correlation matrix in the first phase of our algorithm reported in Section 3 we derive the initial correlation graph shown in Figure A5, where the only (absolute) correlation values above a predefined correlation threshold $\theta_1 = 0.70$ are shown and the initial single cluster identified consists of all six nodes.

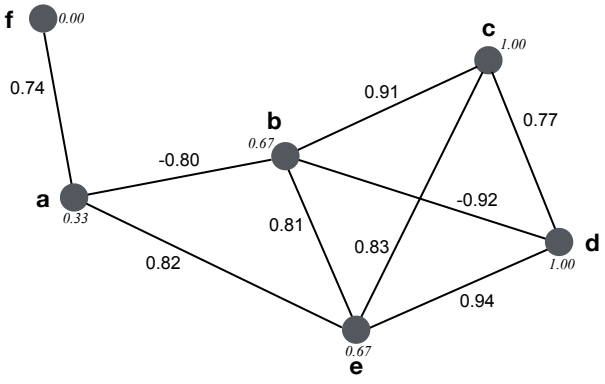


Figure A5: *Phase 1 — The initial correlation graph coming from Table A1. Each node is associated (in italics) with the corresponding clustering coefficient, while each edge shows the correlation coefficient between the two connected nodes.*

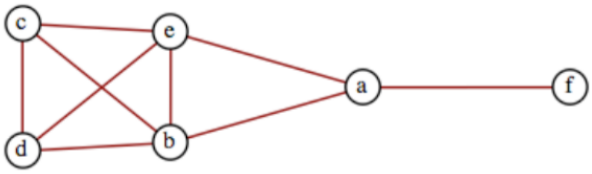
In the second phase, for each node i its clustering coefficient k_i is computed, given by the number of "triangles" (clusters of three directly and fully connected nodes comprising i) as shown in step 5 of our algorithm. In this example, the six clustering coefficients (shown in italics in Figure A5) are: $k_a = 0.33$, $k_b = k_e = 0.67$, $k_c = k_d = 1.00$, $k_f = 0.00$. Then a minimum threshold θ_2 is established, for the elimination of nodes having clustering coefficients lower than this fixed threshold.

When removing a node i with $k_i < \theta_2$, our *relaxed* algorithm also takes into account the nodes close to it (up to a predetermined distance δ). If any of the δ–neighboring nodes has a clustering coefficient ≥ θ_2 then node i will be maintained. In essence, there is a *dragging* effect of a node with a clustering coefficient lower than the threshold θ_2 by the nodes with a higher clustering coefficient and directly connected to it up to a distance δ.

Now, we report some examples of relaxed clustering for different values of θ_2 applied to the initial correlation graph, with the assumption $\delta = 1$.

$\theta_2 = 0.30$

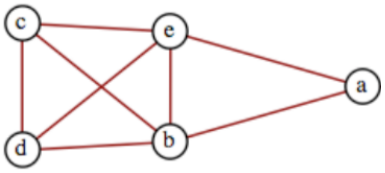
For the clustering threshold $\theta_2 = 0.30$ there are no nodes to be removed, and the corresponding final clustering graph is identical to the initial one:



Note that node f is maintained (dragged by node a at distance $\delta = 1$), whereas it should have to be removed because its clustering coefficient $k_f = 0.00$ is less than $\theta_2 = 0.30$.

$\theta_2 = 0.50$

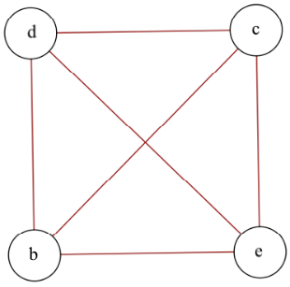
For $\theta_2 = 0.50$ there is one node out of six to be removed (f), and the corresponding final clustering graph becomes:



In this case we note that node a is retained (dragged by nodes b and e at distance $\delta = 1$), whereas it should have to be removed because its clustering coefficient $k_a = 0.33$ is less than $\theta_2 = 0.50$.

$\theta_2 = 0.70$

Now, there are two out of six nodes to be removed (a, f), and the corresponding final cluster is:



Note that the nodes b and e are kept (dragged by nodes c and d), whereas they should have to be removed, due to their clustering coefficient being $k_b = k_e = 0.66 < \theta_2 = 0.70$.

$\theta_2 = 1.00$

Eventually, for $\theta_2 = 1.00$ there are the same two out of six nodes to be removed (a, f) from the previous case, with the same final cluster. Also in this case nodes b and e are maintained (dragged by nodes c and d), whereas they should have to be removed.

It is worth noting that both the initial and final correlation graphs show only one cluster, given the simplicity of our example. In the next example two clusters are obtained due to the different conformation of the connected components in the starting graph and the corresponding correlation and clustering coefficients.

3 Example 2: A more complete application of the "relaxed" clustering algorithm.

Suppose you want now to identify the similarity clusters among eight given wafer bin maps (nodes) a–h (making the analogous array X as in the above example) from which we obtain the correlation matrix C shown in Table A2.

From this correlation matrix in the first phase of our algorithm reported in Section 3 we derive the initial correlation graph shown in Figure A6, where only the (absolute) correlation values above or equal to a predefined correlation threshold $\theta_1 = 0.40$ are shown and the initial single cluster identified consists of all eight nodes.

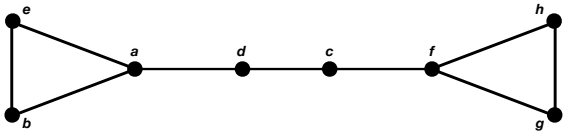
Figure A6: *Phase 1 — The initial correlation graph coming from Table A2. The corresponding clustering coefficient (in italics) is associated to each node, whereas each edge shows the correlation coefficient between the two connected nodes.*

In the second phase, for each node i its clustering coefficient k_i is computed (shown in italics in Figure A6): $k_c = k_d = 0.00$, $k_a = k_f = 0.33$, $k_b = k_e = k_g = k_h = 1.00$. Then a minimum threshold θ_2 is established, for the elimination of nodes with a lower clustering coefficient. As already shown, our algorithm takes also into account — when removing a node — the nodes close to it (up to a predetermined distance δ).

Now we report some examples of "relaxed" clustering for different values of θ_2 applied to the initial correlation graph of Figure A6, with the assumption $\delta = 1$.

$\theta_2 = 0.30$

For the clustering threshold $\theta_2 = 0.30$ there are no nodes to be removed, and the corresponding final clustering graph is identical to the initial one:



Note that nodes c and d are retained ("dragged" by nodes f and a at distance $\delta = 1$), whereas they should

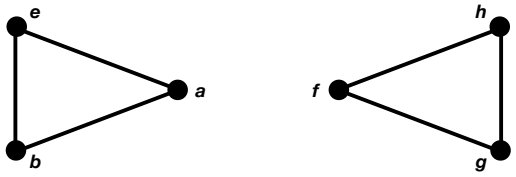
Table A2: The example correlation matrix C between nodes a–h reported in Figure A6.

	a	b	c	d	e	f	g	h
a	–	−0.80	0.35	0.40	0.82	0.24	0.00	0.00
b	−0.80	–	0.11	−0.11	0.81	0.38	0.00	0.00
c	0.35	0.11	–	0.77	0.13	0.55	0.00	0.00
d	0.40	−0.11	0.77	–	0.14	0.18	0.00	0.00
e	0.82	0.81	0.13	0.14	–	−0.33	0.00	0.00
f	0.24	0.38	0.55	0.18	−0.33	–	0.86	0.78
g	0.00	0.00	0.00	0.00	0.00	0.86	–	0.88
h	0.00	0.00	0.00	0.00	0.00	0.78	0.88	–

have to be removed because their clustering coefficient $k_c = k_d = 0.00 < \theta_2 = 0.30$.

$\theta_2 = 0.50$

For $\theta_2 = 0.50$ the two nodes c and d are removed (in fact, their clustering coefficients are $k_c = k_d = 0.00$ and their "relaxed" clustering coefficients are $0.33 < \theta_2 = 0.50$), and the corresponding final clustering graph defines two clusters:



In this case let's note that nodes a and f are retained (dragged by nodes b and e and g and h at distance $\delta = 1$ with clustering coefficient equal to 1), whereas they should have to be removed because their clustering coefficient $k_a = k_f = 0.33 < \theta_2 = 0.50$.

4 Supplemental file

The (anonymized) dataset is available under this link: https://www.crescenziogallo.it/unifg/pub/ic-yield_supplemental_file.xlsx