Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Adjacency-Clustering and Its Application for Yield Prediction in Integrated Circuit Manufacturing

Dorit S. Hochbaum, Sheng Liu

To cite this article:
Dorit S. Hochbaum, Sheng Liu (2018) Adjacency-Clustering and Its Application for Yield Prediction in Integrated Circuit Manufacturing. Operations Research

Published online in Articles in Advance 28 Sep 2018

. https://doi.org/10.1287/opre.2018.1741

http://pubsonline.informs.org/journal/opre/

OPERATIONS RESEARCH
Articles in Advance, pp. 1–15
ISSN 0030-364X (print), ISSN 1526-5463 (online)

Adjacency-Clustering and Its Application for Yield Prediction in Integrated Circuit Manufacturing

Dorit S. Hochbaum,[a] **Sheng Liu**[a]

[a] Department of Industrial Engineering and Operations Research, University of California, Berkeley, Berkeley, California 94720

Contact: hochbaum@ieor.berkeley.edu, http://orcid.org/0000-0002-2498-0512 (DSH); lius10@berkeley.edu, http://orcid.org/0000-0003-2365-6013 (SL)

Received: July 19, 2016
Revised: April 18, 2017; November 28, 2017
Accepted: January 29, 2018
Published Online in Articles in Advance: September 28, 2018

Subject Classifications: manufacturing; prediction; combinatorial optimization
Area of Review: Operations and Supply Chains

https://doi.org/10.1287/opre.2018.1741

Abstract. Accurate yield prediction in integrated circuit manufacturing enables accurate estimation of production cost and early detection of processing problems. It is known that defects tend to be clustered and a chip is likely to be defective if its neighbors are defective. This *neighborhood effect* is not well captured in traditional yield modeling approaches. We propose a new yield prediction model, called *adjacency-clustering* which addresses, for the first time, the neighborhood effect, and delivers prediction results that are significantly better than state-of-the-art methods.

Adjacency-clustering (AC) model is a form of the Markov random field minimum energy model (MRF) that is primarily known in the context of image segmentation. AC model is a novel use of MRF for identifying defect patterns that enable diagnosis of failure causes in the manufacturing process. In this paper we utilize the defect patterns obtained by the AC model for yield prediction. We compare the performance of the AC model to that of leading yield prediction models including the Poisson, the negative binomial, the Poisson regression, and negative binomial regression models, on real data sets and on simulated data sets. The results demonstrate that the adjacency-clustering model captures the neighborhood effect and delivers superior prediction accuracy. Moreover, the concept and methodology of adjacency-clustering are not limited to integrated circuit manufacturing. Rather, it is applicable in any context where a neighborhood effect is present, such as disease risk mapping and energy consumption prediction.

Funding: The work of the first author was supported in part by the National Science Foundation (NSF) [Grant CMMI-1760102].
Supplemental Material: The e-companion is available at https://doi.org/10.1287/opre.2018.1741.

Keywords: Markov random field • yield prediction • semiconductor • adjacency-clustering

1. Introduction

We introduce here a clustering technique for identifying clusters in a set of objects, based on priors available on the objects, as well as on adjacency relationships, or "neighborhood effect." This incorporation of adjacency is critical in setups where the variables to be predicted are affected by their neighbors. The applicability and effectiveness of adjacency-clustering is demonstrated here in the context of yield prediction in integrated circuit manufacturing, where it provides significant improvement over statistical models that have been utilized to date.

Integrated circuit manufacturing is a highly complex, costly process that involves hundreds of chemical or physical processing steps (Yuan et al. 2011). The key processes include wafer fabrication, wafer probe, assembly or packaging, and final test. The degree of manufacturing success is measured by *yield*, which is defined as the average ratio of the number of usable devices that pass tests after completing processes to the number of potential usable devices before starting processes (Kim and Kuo 1999, Ferris-Prabhu 1992). Accurate yield prediction is critical for managers to estimate productivity, production cost, and to make scheduling decisions. Moreover, yield prediction helps to detect processing problems in an early production stage, which is crucial to quality improvement.

In semiconductor manufacturing, there are four components to the yield: wafer process yield, wafer probe yield, assembly yield, and final test yield (Milor 2013). Among these, wafer process yield (also known as line yield) and wafer probe yield (also known as die yield) are considered to be the major cost determining factors (Cunningham 1990). Wafer probe defects found in integrated circuits (also called chips) include shorts, opens, misalignment, photoresist splatters and flakes, and pinholes (Stapper et al. 1983). A chip containing at least one fatal defect is considered defective and "good" otherwise.

Yield prediction based on defect data from sampled wafers is to estimate the ratio of nondefective (good) chips to the total number of chips. Until now only statistical models have been utilized for this purpose. The classical yield model assumes that the number of defects on a chip follows Poisson distribution with density λ, taken to be the average number of defects on

a chip. It is assumed that the value of λ is uniform across all chips and wafers. The yield is then estimated as the probability that no defects occur on a chip. In later work researchers relax the assumption of the constant λ and assume λ itself follows a specific distribution. Two popular such models are Murphy's model and Seeds's model proposed in Murphy (1964) and Seeds (1968), respectively. In addition, negative binomial distribution and variants of Poisson distribution have been applied to improve yield prediction in recent years (see, e.g., Bae et al. 2007).

Existing models assume that the distribution of defects is identical for all chips on the wafer. Yet this is not the case in practice where defects are known to be clustered in contiguous groups (Bae et al. 2007, Hansen et al. 1997). Indeed, various mechanisms causing defects tend to only affect certain regions of the wafer (Hwang and Kuo 2007, Jeong et al. 2008, Stapper et al. 1983). Chips in close vicinity of each other are more likely to be affected by the same defect generating mechanism, and therefore the number of defects on a chip is correlated with the number of defects on its neighbors.

The adjacency-clustering approach introduced here is to partition the set of chips on the wafer to subsets, referred to as clusters, so that each cluster contains chips with similar defect level, which also tend to be adjacent to each other. This is attained by minimizing a combination of two objective functions, one that penalizes deviation from the priors (observed number of defects), and the second that penalizes the separation of adjacent chips to different clusters. The resulting clusters tend to contain chips with the same defect distribution since, because they reside in the same neighborhood, they are likely to be caused by the same mechanism. The wafer yield prediction is then attained from a combination of the individual cluster yields. This approach is in contrast to existing yield prediction methods that predict the wafer yield without differentiating among clusters. The performance of our approach is demonstrated via an empirical study on real data sets and on simulated data sets. The results show that the adjacency-clustering approach improves the prediction accuracy by a factor between 3 and 15 as compared to the use of Poisson and Poisson regression model for the real data sets. This superior performance of adjacency-clustering over the state-of-the-art methods is further validated on simulated data.

The success of the adjacency-clustering approach for yield prediction bodes well to its potential applicability to other contexts where the neighborhood effect is an important factor in clustering. This is the case for disease mapping where spatially correlated disease data are utilized to identify high-risk areas (clusters) and predict risk levels (see Charras-Garrido et al. 2012 for more details). Another case is that of energy consumption prediction for households where high consumption households tend to be in the vicinity (see, e.g., Baker and Rylatt 2008).

The remainder of this paper is organized as follows. Section 2 reviews related literature. Section 3 introduces notations and our approach. Section 4 describes the adjacency-clustering model and its solution method. In Section 5 we provide the analysis of the performance of our approach as compared to leading existing models for four real data sets. Section 6 includes a detailed performance comparison on simulated wafer maps. Conclusions and future directions are discussed in Section 7.

2. Related Literature

Relevant literature is reviewed here within three streams: (1) advanced statistical yield prediction models; (2) methods for measuring the extent of spatial aggregation of defects on wafers given defect counts observations; (3) identifying and classifying defect spatial patterns in a wafer.

2.1. Advanced Statistical Yield Prediction Models

Among statistical yield models the Poisson model is most widely used. A drawback of the Poisson model is that it is known to considerably underestimate the yield for wafers with defects that are aggregated nonuniformly (see, e.g., Stapper et al. 1983, Stapper 1989). To overcome this limitation, Stapper et al. (1983) derive a negative binomial model by assuming the probability that a defect occurs in a chip depends on the number of faults already on the chip, which is equivalent to assuming that λ follows a gamma distribution. Albin and Friedman (1989) introduce an alternative distribution, Neyman distribution, to fit the defect data. Koren et al. (1993) add a new parameter, block size, to the negative binomial model to account for the aggregation effects of defects. Although the negative binomial model and the Neyman model capture the defect aggregation, they fail to model the spatial information of chips and relationship between adjacent chips. For instance, these models ignore a common defect pattern where defects tend to be aggregated on the periphery of wafers, called *radial loss* (Ferris-Prabhu et al. 1987). To account for such spatial position effects, regression models (generalized linear models) are introduced: Bae et al. (2007) propose Poisson, negative binomial, and zero-inflated Poisson regression model. Yuan et al. (2011) introduce zero-inflated binomial negative model. Among these regression models, negative binomial regression model yields the lowest prediction error in general. More recently, Krueger and Montgomery (2014) introduce generalized linear mixed models for yield modeling to capture longitudinal correlation between and within batches of samples. However, they only explore the longitudinal correlation but ignore neighborhood effect within wafer.

Although these regression models improve yield prediction accuracy, estimation issues remain challenging. Two main estimation methods, Bayesian method and maximum likelihood method, are employed in the parameter estimation of regression models. Bayesian methods based on Markov chain Monte Carlo (MCMC) are time consuming and unstable for small-size samples, while maximum likelihood estimation methods may not provide tight interval estimate for parameters (Ghosh et al. 2006). In addition, for samples showing complicated spatial patterns, it is challenging to choose appropriate covariates and set up the linear relationship in regression models.

2.2. Measuring Spatial Clustering

Hansen et al. (1997) introduce a monitoring statistic to test the significance of spatial clustering based on Markov random field. Fellows et al. (2009) study the empirical performance of the Hansen et al. method on real data sets. Hansen et al. (1997) also propose the join-count statistics to measure the spatial randomness and the degree of clustering, where join is formed with two neighboring chips and join counts are measures of the adjacencies between different levels of a variable. Taam and Hamada (1993) utilize the join-count to propose the log odds ratio as a measure of spatial clustering. Jeong et al. (2008) further generalize join-count-based statistics with optimal weights, and they introduce the spatial correlogram to detect the presence of spatial autocorrelation. There are other statistics known that can be used as defect clustering indices, as reviewed in Tsai et al. (2008). All studies that measure spatial clustering are based on binary defect data, where chips are differentiated only in whether or not they contain defects. These methods highlight the importance of spatial clustering but do not apply for the yield prediction tasks addressed here.

2.3. Classifying Defect Patterns

Classifying defect patterns is important for the purpose of diagnosis of failure causes. Chen and Liu (2000) employ neural networks to recognize spatial defect patterns. Di Palma et al. (2005) test the approach of Chen and Liu on simulated and real data set. White et al. (2008) develop a procedure to detect different arrangements and shapes of defect aggregations (clusters). Recently, several recognition techniques based on support vector machines (SVM) have been tested on wafer defect data to identify different defect patterns (e.g., see Li and Huang 2009, Chao and Tong 2009, Yuan et al. 2010, and Wu et al. 2015). Ooi et al. (2013) develop an automatic defect pattern recognition system integrating feature extraction, selection, and classification techniques. These methods are helpful in diagnosis, but they do not explore how defect patterns can help yield prediction. To the best of our knowledge, our paper is the first attempt to utilize defect clustering pattern to improve yield prediction result.

3. Our Approach

The defect data available in semiconductor manufacturing is in the form of wafer maps. A wafer map example is given in Figure 1, where the number of defects on each chip is indicated at the chip position on the grid. Let the defect data for a wafer map on n chips be represented by the array $(d_1, d_2, \ldots, d_n)$, where d_i is the observed number of defects at location i or the i-th chip. The wafer map is formalized as a graph $G = (V, E)$ where each node in V represents a chip, and each pair, i, j, of neighboring chips is associated with an edge $[i, j] \in E$. There are several alternatives for neighborhood relationship, e.g., 4-neighbor system (rook-move neighborhood) and 8-neighbor system (king-move neighborhood). We select here the 4-neighbor system.

The goal of adjacency-clustering is to partition the set of chips into clusters, so that the chips that belong to the same cluster tend to have similar defect levels as well as to be adjacent to each other. These two goals are balanced by a parameter that weighs one goal versus the second. The clustering is represented by cluster label x_i assigned to chip i. One goal is to require that the chip label, x_i for chip i, deviates as little as possible from the observed value d_i, under a penalty called *deviation cost*. For the second goal there is a penalty associated with the difference in assigned labels for neighboring chips. This penalty, called *separation* penalty, is associated with each pair of adjacent chips, or nodes in $G = (V, E)$ that are linked with an edge of E that differ in their labels. The goal is to attain a solution that minimizes a combination of the two objectives of deviation and separation penalties. Let $f_i(x_i, d_i)$ be deviation functions associated with node $i \in V$ and $g_{ij}(x_i - x_j)$ be separation functions associated with every edge $[i, j] \in E$. Let X be a set of cluster label values. The adjacency-clustering model (AC) is formulated as a *deviation-separation* optimization problem as follows:

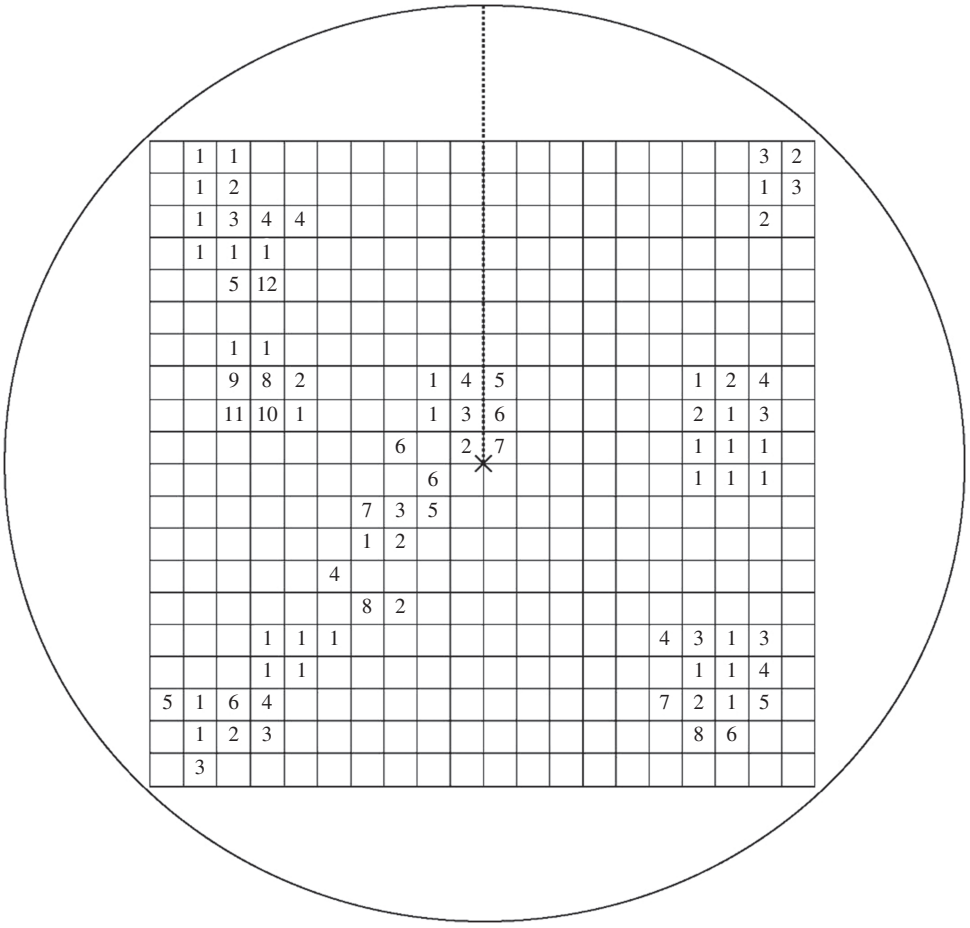
$$\min \left\{ \sum_{i \in V} f_i(x_i, d_i) + \sum_{[i,j] \in E} g_{ij}(x_i - x_j) \right\} \tag{1}$$

$$\text{s.t. } x_i \in X, \quad \forall\, i \in V. \tag{2}$$

This deviation-separation formulation arises in contexts such as computer vision and statistics, where it is referred to as Markov random field (MRF) (see, e.g., Blake and Zisserman 1987, Ishikawa and Geiger 1998, Hochbaum 2001).

All nodes that share the same label are considered to be a single cluster. The optimal cluster partition depends on the trade-off between the deviation and separation penalties. The larger the separation penalty

Figure 1. A Wafer Map Example Used by Bae et al. (2007)



functions, the more contiguous the resulting clusters. In contrast, relatively large deviation penalties render clusters that group together objects with similar or identical observation values regardless of their spatial positions.

The label of a cluster corresponds to the *yield level* of the cluster. We choose integer cluster labels in $\{0, 1, \ldots, k\}$ where a higher label value indicates a greater likelihood of having large number of defects and thus a lower yield level. For example, we may choose the labels $\{0, 1, 2\}$, with the interpretation that the model predicts no defects for chips in the cluster labeled 0, moderate number of defects for chips in the cluster labeled 1, and high number of defects for chips in the cluster labeled 2. Another example is for the binary labels $\{0, 1\}$ implying a distinction between a cluster that tends to contain chips with very small number of defects or is surrounded by such chips, and a cluster that tends to contain chips with large numbers of defects. Such clusters are interpreted as nondefective versus defective clusters.

Our yield prediction model works by first generating the adjacency-clustering. The output of AC is a partition of the wafer's set of chips into $\{V_0, V_1, \ldots, V_k\}$, where V_0 is the cluster of chips that are labeled nondefective and V_j for $j = 1, \ldots, k$ are clusters that for larger label values are increasingly likely to contain larger numbers of defects. In the second stage a yield model is applied to each cluster, and the weighted average of cluster yields ($\hat{y}_j$ for $j = 0, \ldots, k$) is the reported wafer yield prediction $\hat{y}$:

$$\hat{y} = \frac{\sum_{j=0}^{k} |V_j| \hat{y}_j}{\sum_{j=0}^{k} |V_j|}.$$

For this second stage we use Poisson model and negative binomial model, or a mixture of the two, as a yield model applied to each cluster. We use the notation AC-Poisson to indicate the use of adjacency-clustering in stage one followed by the Poisson model as a yield model in stage two. Similarly we use the notation AC-NB, AC-PNB, and AC-NBP to indicate the adjacency-clustering followed by negative binomial (NB) model, a combination of Poisson for the nondefective clusters and NB for defective ones, and a combination of NB for the nondefective clusters and Poisson for defective ones, respectively. Table 1 lists the nomenclature for the models, whether using AC, and the respective yield model.

Table 1. Model Names, Clusters Generated, if Any, and Yield Models

Model	Clusters generated	Yield model
AC-Poisson	$(V_0, V_1, \ldots, V_k)$	Poisson
AC-NB	$(V_0, V_1, \ldots, V_k)$	Negative binomial
AC-NBP	$(V_0, V_1, \ldots, V_k)$	V_0: Negative binomial $V_1, \ldots, V_k$: Poisson
AC-PNB	$(V_0, V_1, \ldots, V_k)$	V_0: Poisson $V_1, \ldots, V_k$: Negative binomial
Poisson	V	Poisson
Poisson regression	V	Poisson regression
Negative binomial	V	Negative binomial
Negative binomial regression	V	Negative binomial regression

4. The Adjacency-Clustering Model and Its Solution Technique

First we discuss the use of adjacency-clustering for yield prediction in the case of binary labels. Then we discuss the choice of penalty functions for multilabel instances and present the solution technique for the resulting multilabel adjacency-clustering model.

4.1. Binary Adjacency-Clustering Model

For binary labels $\{0,1\}$ the wafer is partitioned into only two clusters: one representing a set of defective chips that are labeled 1, and the second representing nondefective chips that are labeled 0. Let $V_0 = \{i \in V: d_i = 0\}$, $V_+ = \{i \in V: d_i > 0\}$. For chip $i \in V_0$ assigning $x_i = 0$ imposes no deviation cost, whereas assigning $x_i = 1$ imposes a penalty of $w_{i0} > 0$. Likewise, for chip $i \in V_+$ assigning $x_i = 1$ imposes no deviation cost, whereas assigning $x_i = 0$ incurs a penalty of $w_{i+} > 0$. With this notation, the total deviation cost (penalty) of assigning the x_i labels is,

$$\sum_{i\in V_+} w_{i+} + \sum_{i\in V_0} w_{i0}x_i - \sum_{i\in V_+} w_{i+}x_i. \tag{3}$$

Let the separation cost function be $g_{ij}(|x_i - x_j|)$, which equals to $u_{ij} > 0$ if $x_i \neq x_j$ and 0 otherwise. The optimization problem that minimizes the sum of these deviation and separation costs is (omitting the constant term $\sum_{i\in V_+} w_{i+}$),

$$\min \left\{ \sum_{i\in V_0} w_{i0}x_i - \sum_{i\in V_+} w_{i+}x_i + \sum_{[i,j]\in E} u_{ij}z_{ij} + \sum_{[i,j]\in E} u_{ij}z_{ji} \right\} \tag{4}$$

$$\text{s.t. } z_{ij} \geqslant x_i - x_j, \quad \forall [i,j] \in E, \tag{5}$$

$$z_{ji} \geqslant x_i - x_j, \quad \forall [i,j] \in E, \tag{6}$$

$$x_i \in \{0,1\}, \quad \forall i \in V, \quad z_{ij} \in \{0,1\}, \quad \forall [i,j] \in E. \tag{7}$$

Here the constraints ensure that $z_{ij} = 1$ for adjacent nodes i and j if $x_i \neq x_j$ and 0 otherwise.

The previous problem is the *minimum s-excess problem* (Hochbaum 2001), which is solved in polynomial time by applying a minimum-cut procedure on an associated graph. The optimal solution is a partition to two clusters, one of nodes of label 0, and the other of nodes of label 1. To allow for higher levels of differentiation between yield levels of clusters, we present next the multilabel version of AC.

4.2. Multilabel Adjacency-Clustering Model

In the multilabel case we let the set of labels be $\{0, 1, \ldots, k\}$, where k is a parameter specified by the user. The choice of k implies there are $(k + 1)$ potential labels that characterize the yield level of each chip. For instance, if $k = 2$, a wafer is partitioned into three types of clusters: nondefective ($x_i = 0$), medium defective ($x_i = 1$), and highly defective ($x_i = 2$).

4.2.1. Selecting Deviation and Separation Functions.

For nonbinary labels, the deviation and separation functions must be specified. For deviation functions we consider quadratic functions that correspond to Gaussian distribution in Bayesian estimation. Gaussian distribution is commonly used to approximate many distributions, and the corresponding quadratic deviation functions are widely applied in image segmentation and spatial statistics (Panjwani and Healey 1995, Held et al. 1997, Rue 2001). When the observation is not Gaussian, batching and averaging observations can lead to an approximately Gaussian sample (Law 2014). Since we impose no restrictions on the probabilistic relationship between the number of defects and yield level, such quadratic functions are suitable. It is noted that the common use of quadratic deviation functions in the literature is due in part to the existence of well-known algorithms, e.g., based on KKT conditions, that can be used for solving quadratic minimization problems. This is not the motivating reason in our case for choosing quadratic deviation functions.

In terms of the separation functions, absolute value penalty $u_{ij}|x_i - x_j|$ (ℓ_1 norm) is commonly used to penalize the difference of neighboring labels. Occasionally, in image segmentation context, the truncated form $g_{i,j}(x_i - x_j) = u_{i,j} \cdot \min\{|x_i - x_j|, M\}$ for a positive value M, is considered desirable (Veksler 2007, Szeliski et al. 2008). This form avoids the over-smoothness associated with the absolute value penalty, which occurs for large gaps in label values. However, these truncated

separation functions are nonlinear, and the respective problem is NP-hard and challenging even to approximate. Furthermore, in our setup the label values are small integers, and hence over-smoothness is not a concern.

We select here quadratic deviation functions and absolute value separation functions. For these functions, the adjacency-clustering formulation is

$$\text{(AC)} \quad \min \quad \sum_{i\in V}(x_i - d_i)^2 + \sum_{i\in V}\sum_{j:[i,j]\in E} u_{ij}|x_i - x_j| \quad (8)$$

$$\text{s.t.} \quad x_i \in \{0,1,\ldots,k\}, \quad \forall\, i \in V. \quad (9)$$

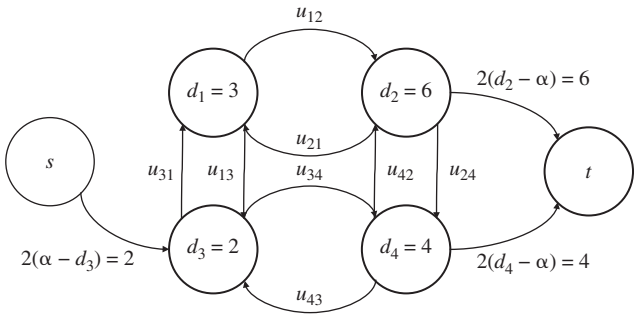
This (AC) problem is an MRF on convex separation and deviation functions. Such convex MRF is solved in polynomial time with the algorithm of Ahuja et al. (2003). For separation functions that are of the form $g_{i,j}(x_i - x_j) = u_{ij}|x_i - x_j|$ and convex deviation functions, the algorithm of Hochbaum (2001) is very efficient and provably fastest possible. The special structure of (AC), with quadratic deviation functions, is shown next to be solved with a yet more efficient algorithm.

4.3. An Efficient Solution Technique for the Multilabel Adjacency-Clustering Model

We now describe a particularly efficient algorithm for solving (AC) with quadratic deviation functions and absolute value separation functions. The key to the efficiency of the algorithm is the threshold theorem that links a minimum cut in an associated graph, G_α, with the optimal values of the variables.

For α a scalar in the range of the variables, the graph associated with $G = (V, E)$ is an s,t-graph G_α constructed as follows: We add to the graph $G = (V, E)$ a source node s and a sink node t; for each node $i \in V$ we add an arc from s of capacity $\max\{\partial f_i/\partial x_i(\alpha), 0\}$ and an arc to t of capacity $\max\{-\partial f_i/\partial x_i(\alpha), 0\}$, where $\partial f_i/\partial x_i(\alpha) = 2(\alpha - d_i)$. Note that at least one of these arcs must have capacity of 0; thus, a node can be connected either to source or to sink but not to both. Each edge $[i, j] \in E$ is replaced by a pair of arcs (i, j) and (j, i) both of capacity u_{ij}. An example of a G_α graph on 4 nodes and $\alpha = 3$ is illustrated in Figure 2. Note that node 1 (the top left node in the graph) has neither an arc from the source nor an arc to the sink since the respective derivative is $\partial f_1/\partial x_1(\alpha) = 2(\alpha - d_1) = 0$.

Figure 2. An Example of G_α on a 4-Node Graph and $\alpha = 3$



Because of the convexity of the functions $f_i(\,)$, the graph G_α has the property that the source adjacent capacities are monotone increasing (more generally, monotone nondecreasing) in α, and the sink adjacent capacities are monotone decreasing (more generally, monotone nonincreasing) in α. Such graphs are called parametric flow graphs. Let a minimum cut in G_α be $(S_\alpha \cup \{s\}, \bar{S}_\alpha \cup \{t\})$, where $S_\alpha \cup \{s\}$ is the *source set* of the minimum cut. If there are multiple minimum cuts, we select the one where $S_\alpha \cup \{s\}$ is *minimal* (contained in all source sets of minimum cuts). It is well known that the source sets of minimum cuts in parametric flow graphs are *nested*: For $\alpha_1 < \alpha_2$, $S_{\alpha_1} \subseteq S_{\alpha_2}$. The nestedness is also a corollary of the following threshold theorem, which states the relationship between the optimal solution to (AC), $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$, and the source set of a minimum cut in G_α (Hochbaum 2001):

Theorem 1 (Threshold Theorem)**.** *For S_α the minimal source set of a minimum cut in G_α, the optimal solution x^* to* (AC) *satisfies $x_i^* < \alpha\ \forall\, i \in S_\alpha$ and $x_i^* \geqslant \alpha\ \forall\, i \in \bar{S}_\alpha$.*

With the threshold theorem, the following algorithm is used to solve (AC): Call for a minimum cut procedure in the graphs G_α for $\alpha = 1, \ldots, k$ resulting in the sequence of nested source sets, $\{s\} = S_0 \subseteq S_1 \subseteq \cdots \subseteq S_k \subseteq S_{k+1} = V \cup \{s\}$. Let $\Delta_\alpha = S_\alpha \backslash S_{\alpha-1}$; then the optimal solution $\mathbf{x}^*$ is determined as follows:

$$\text{if } i \in \Delta_{\alpha+1} \text{ then } x_i^* = \alpha.$$

Let $T(n, m)$ be the complexity of a minimum cut algorithm on a graph with n nodes and m arcs; then this algorithm requires $O(kT(n, m))$ steps to solve (AC).

To improve on the complexity, we notice that because of the "nestedness" property, for $\alpha_1 < \alpha_2$, once the maximum flow in G_{α_1} is found, we can "shrink" the source set S_{α_1} with the source node s, as it is guaranteed that S_{α_1} is part of the source set of a minimum cut in G_{α_2}. Once the arcs adjacent to source and sink are adjusted for the new parameter value α_2, the previous maximum flow is feasible except possibly for the arcs adjacent to sink where their capacities have gone down. A parametric flow algorithm can warm-start from such a solution and solve the entire sequence of k parametric flows and cuts in the complexity of a single maximum flow (Gallo et al. 1989). The push-relabel algorithm (Goldberg and Tarjan 1988) or Hochbaum's pseudo-flow (HPF) algorithm (Hochbaum 2008) are both known to have this capability, and both run, for k parameter values, in $O(mn \log n^2/m + kn)$ steps on a graph with n nodes and m arcs. As a result, solving (AC), where m is $O(n)$ for the graph G, can be accomplished in $O(n^2 \log n + kn)$.

Theorem 2. *The time complexity of an algorithm solving* (AC) *with a parametric minimum cut HPF or push-relabel is $O(n^2 \log n + kn)$.*

5. Empirical Results on Real Data Sets

We analyze four real wafer maps in this section. The first one appears in Tyagi and Bayoumi (1994), and the other three are presented by Yuan et al. (2011). The first wafer map has $20 \times 20 = 400$ chips, and the other 3 have each contained 473 chips.

For all four wafer maps, we choose separation penalties $g_{ij}(x_i - x_j) = u \cdot |x_i - x_j|$, with u a factor that is common for all wafer maps and uniform for all pairs of chips. This is because there is no ex ante information to differentiate between different pairs. In case there is a reason to differentiate, or to stress the neighborhood effect in some areas of the wafer more than in others, one can select a nonuniform value of u. We test 3 different values of k $(= 1, 2, 3)$ combined with 26 different values of u $(= 0.5, 0.6, \ldots, 3)$. The selection of u is used to balance the separation versus the deviation penalties.

The experiments in this section and Section 6 are performed on a Lenovo X1 computer running the Windows 10 64-bit operating system with an Intel Core i5-5200U 2.20 GHz processor and 8.0 GB RAM. The adjacency-clustering problem for $k = 1$ is solved with Hochbaum's pseudo-flow (HPF) algorithm (available at http://riot.ieor.berkeley.edu/Applications/Pseudoflow/maxflow.html; see Hochbaum 2008, Chandran and Hochbaum 2009). The adjacency-clustering problem for $k \geqslant 2$ is solved with parametric maximum flow using parametric HPF (the source code used is available at http://riot.ieor.berkeley.edu/Applications/Pseudoflow/parametric.html).

The results are presented in the following sections: In Section 5.1 we illustrate the qualitative effect of changing the two parameter values, k and u, on the resulting clustering. Section 5.2 describes the application of the AC-Poisson model and the evaluation of the choice of the parameters in terms of the prediction error. The prediction error is measured by *relative absolute bias*, defined as

$$\frac{|\text{True yield} - \text{Estimated yield}|}{\text{True yield}}.$$

The lowest prediction errors lead to a choice of parameters for AC-Poisson, which is used afterward. In Section 5.3 the AC-Poisson model, with the specific selection of parameters, is compared to the Poisson model and the Poisson regression model in terms of the relative absolute bias. Finally, in Section 5.4, we test various yield models for the clusters generated by AC: First we test the negative binomial yield model, and then a combination of two different yield models (Poisson and negative binomial) for the no-defects cluster (of label 0), and the defective clusters, of positive label. These results are then compared with the negative binomial and negative binomial regression prediction models (that apply to the entire wafer).

5.1. Visual Clustering Results for Varying Parameters

We present first, visually, the clustering results for the first wafer map with different choices of the parameters. As shown in Figure 3, as the value of k increases (going down the rows of images), the clusters corresponding to positive values of the label are becoming more differentiated into small contiguous groups. As for the value of u, that increases for the columns of images from left to right, the effect is to create more contiguous clusters, since a higher value of u causes higher penalty for noncontiguity. Indeed, Figure 3(a) consists of many small contiguous groups while in Figure 3(d) there are only a few large groups. It should be noted that clusters generated by AC models are not necessarily contiguous. That is, there is a trade-off between contiguity, that implies contiguous chips should fall in the same cluster, and clustering chips with similar number (or density) of defects. For instance, adjacent chips tend to belong to the same cluster, unless there is a substantial gap between their numbers of defects. On the other hand, nonadjacent chips that have the same number of defects may well fall in the same cluster, resulting in a cluster by a collection of noncontiguous groups. We observe that for this first wafer map, the groups of positive labeled chips are positioned near the center and the four corners of the wafer, implying potential manufacturing problems.

The effects illustrated in Figure 3 on changing the values of u and k indicate a general trend. When u is small, the separation cost is low, and the adjacency effect is not playing a role. In contrast, for u that is very large, the separation cost dominates the objective function and the tendency is to group many of the chips in a single contiguous cluster, even if they differ substantially in their numbers of defects. In the extreme case where $u = +\infty$, the entire wafer forms a single cluster. Similarly for parameter k, when k is small, for example, the binary case, groups of high number of defects may be clustered together with groups of low number of defects. If k is large, then the clusters tend to have small number of objects, which may result in poor prediction performance. Next we study the effects of the parameters selection on the prediction error, for the AC-Poisson model.

5.2. Parameter Selection for AC-Poisson

AC-Poisson generates $k + 1$ clusters, and the yield of each is then computed with a Poisson yield model. The yield for each cluster j is estimated as $\hat{y}_j = \exp(-\lambda_j)$, where λ_j is the average number of defects for the cluster. These estimates are then used to predict the yield for the whole wafer.

To determine which parameters to select and how their choice affects the relative absolute bias error, we apply AC-Poisson to different combinations of u and k

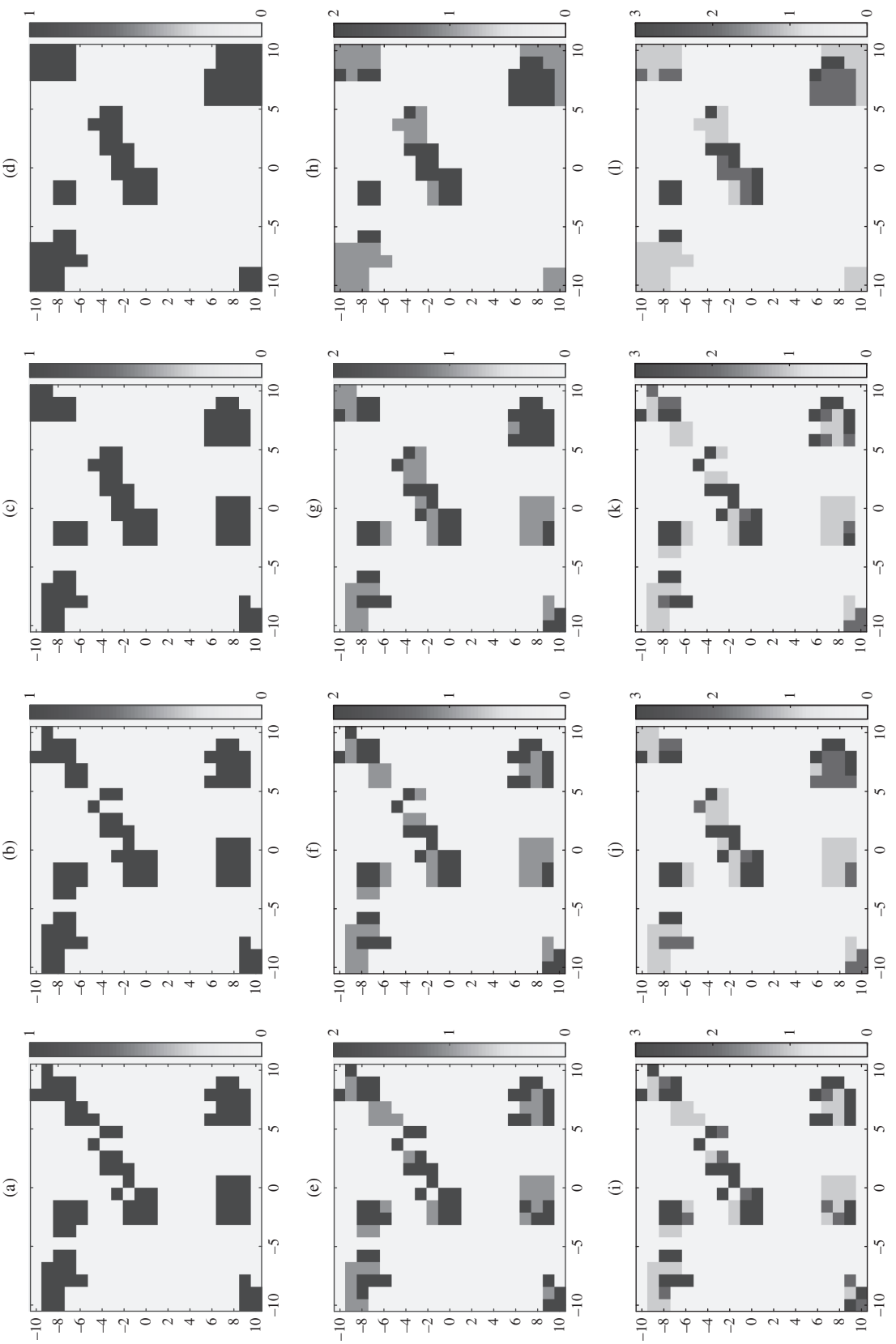
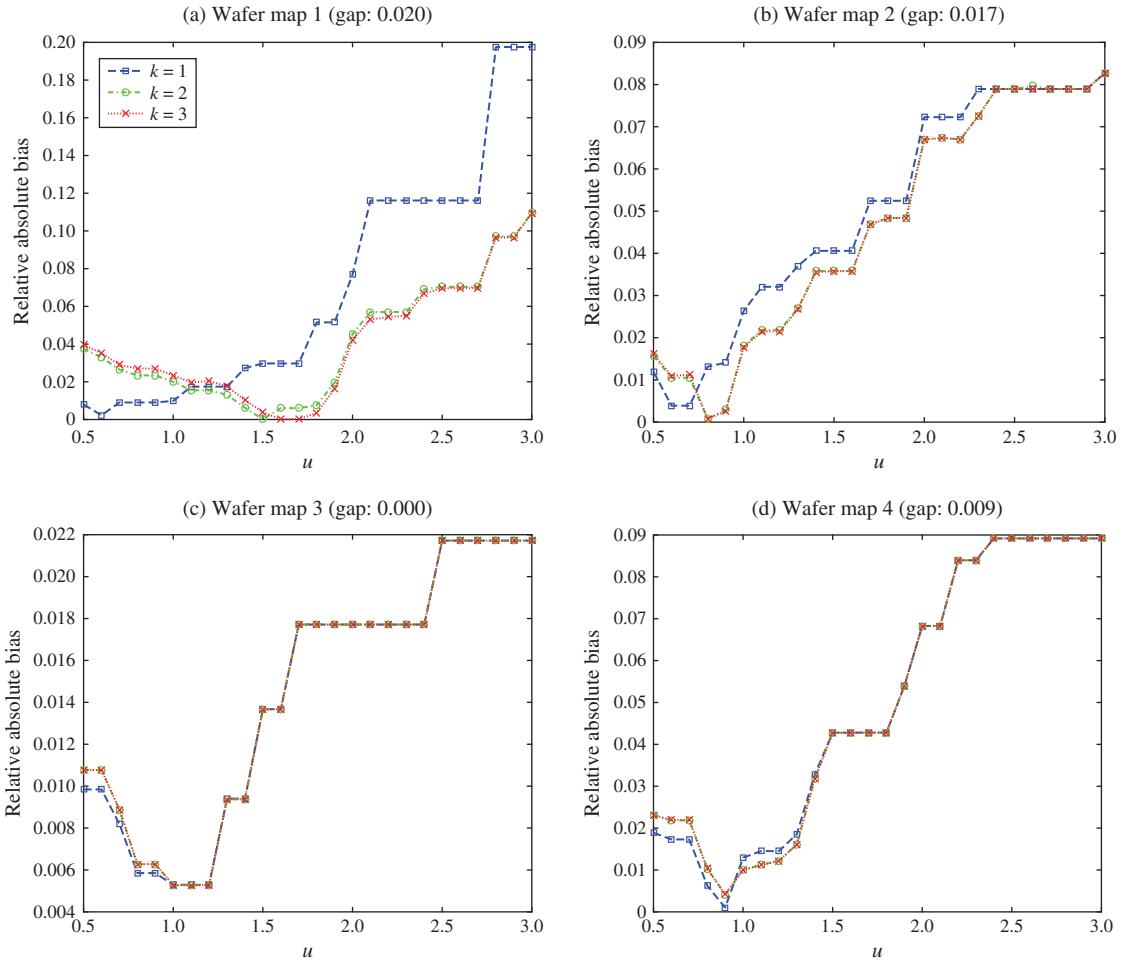


Figure 3. Adjacency-Clustering Results for Sample Wafer Map 1 Where the First Row Presents the Results for $k = 1$, the Second Row Is for $k = 2$, and the Third Row Is for $k = 3$; From Left to Right, the Four Columns Correspond to Increasing Values of $u = 0.1, 0.5, 1, 2$

Note. Different clusters are differentiated based on the colors.

Figure 4. (Color online) Relative Absolute Bias Error of AC-Poisson Model for u in $\{0.5, 0.6, \ldots, 3\}$ and k in $\{1, 2, 3\}$ on Four Real Data Sets



Note. The gap value is the difference between the minimum error across combinations attained and the error for $u = 1$ and $k = 2$.

for the four data sets. Figure 4 presents the relative absolute bias of AC-Poisson model for each data set with the choice of values of u in $\{0.5, 0.6, \ldots, 3\}$ and values of k in $\{1, 2, 3\}$. The results indicate that the choice of the combination of $u = 1$ and $k = 2$ is very close to the best combination of the two parameters. Indeed, as will be shown, in all our experiments this combination is close to the best combination. Therefore, we refer to it as the *default setting*. Here, the gaps between the minimum error across all combinations and the error attained for the default setting of $u = 1$ and $k = 2$, are 0.020, 0.019, 0.000, 0.009 for wafer maps 1, 2, 3, 4, respectively.

A known statistical method of selecting a value such as k is the Bayesian information criterion (BIC). BIC is a trade-off between an increase in the number of parameters and an increase in the likelihood of all observations that results from finer distributions with larger number of parameters. In the context of the adjacency-clustering model, the number of clusters increases with k, and for each cluster the yield estimation requires the estimation of the cluster's distribution parameters. For instance, using Poisson model for each cluster, the total number of estimated parameters is $k + 1$. Therefore, the number of parameters, denoted by h_k, grows here linearly with k. For $\mathcal{L}$ the likelihood of all observations on the wafer, the BIC score is defined as

$$\text{BIC} = h_k \ln n - 2 \ln \mathcal{L}.$$

The lower the BIC score the better. Setting $u = 1$ and the AC-Poisson model, we compute the BIC scores for $k = \{1, 2, 3, 4\}$ on the four real wafer maps, as shown in Table 2. For each wafer map, the value of k corresponding to the lowest BIC is assigned a rank of 1 and the second lowest is assigned a rank of 2, and so forth. The average rank across the four samples is presented in the last column of Table 2. From Table 2, $k = 2$ has the lowest average rank, which is one of the reasons why we select this value of k in our default setting.

We discuss further issues concerning the choice of u and k in the section on simulated data, Section 6.

Table 2. BIC of AC-Poisson

Wafer map	1	2	3	4	Average rank
$k=1$	478.73	498.42	348.55	568.08	2.75
$k=2$	384.19	476.97	354.63	564.92	1.75
$k=3$	355.14	480.33	360.79	571.08	2.5
$k=4$	351.51	486.35	366.95	577.24	3

For the comparison with other prediction models we are selecting the default setting of $u=1$ and $k=2$ as the one for AC-Poisson.

5.3. Performance Comparison of AC-Poisson with Poisson and Poisson Regression Models

Table 3 provides the comparison of the relative absolute bias for AC-Poisson model ($u=1$ and $k=2$), Poisson model, and Poisson regression model. In Poisson regression model, the covariate vector is selected as $\{r, \cos\phi, \sin\phi, r\cos\phi, r\sin\phi\}$, as suggested by Bae et al. (2007). (Using the center of the wafer as the reference point, r and ϕ denote the radial coordinate and angular coordinate in the polar coordinate system.) We estimate the corresponding coefficients by maximum likelihood method with *glm*( ) function in R (see https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html for details). The adjacency-clustering prediction results of AC-Poisson improve by a factor that varies between 3 to 15 on the Poisson regression model, and by a larger factor on the Poisson model. In particular, the improvement for wafer map is remarkably large. This may be the case, since the defects in wafer 1 are heavily clustered and exhibit a clear pattern (as shown in Figure 3), which can be easily captured by AC-Poisson.

It is noted that even though the Poisson regression model is intended to incorporate the spatial positioning that is absent in the Poisson model, our results indicate that it is only minimally effective. In addition, our results show that AC-Poisson model dominates the Poisson regression for any choice of $u \leqslant 2.3$ and $k \in \{1,2,3\}$. The results reported in Table 3 provide evidence that strongly supports the capability of adjacency-clustering to introduce major improvements in yield prediction.

In terms of running time, solving the adjacency-clustering model (with parametric maximum flow using HPF algorithm) requires 0.07 seconds for sample 1 and an average of 1.11 seconds for samples 2, 3, 4.

5.4. Testing AC with Different Yield Models: Comparison of AC-NB, AC-NBP, and AC-PNB

In addition to the Poisson yield model, the negative binomial model is also widely used in yield prediction. Compared with Poisson model, it is less likely to underestimate the yield (Kim 2011). Following negative binomial model, the yield for cluster j is given by $\hat{y}_j = (1+\lambda_j/\gamma_j)^{-\gamma_j}$, where γ_j is called the cluster parameter. There are multiple ways of determining γ_j (see Cunningham 1990 for details), and we adopt the method of moments as

$$\gamma_j = \frac{\lambda_j^2}{\sigma_j^2 - \lambda_j}. \tag{10}$$

Here σ_j^2 is the variance of the number of defects per chip for the cluster, which is estimated by the sample variance. Three different prediction models combined with adjacency-clustering are used here: (1) AC-NB model—negative binomial yield model is fitted to each cluster; (2) AC-NBP model—negative binomial yield model is fitted to nondefective clusters (cluster with "0"s), while Poisson yield model is fitted to defective clusters (of label > 1); (3) AC-PNB model—Poisson yield model is fitted to nondefective clusters, while negative binomial yield model is applied to defective clusters. These three models are tested on the four wafers for different combinations of u and k. We select the parameter values that yield the lowest prediction errors (AC-NB: $u=0.7$, $k=3$; AC-NBP: $u=0.6$, $k=1$; AC-PNB: $u=0.7$, $k=3$). Experimental results for the choice of these parameter values are provided in the e-companion. It should be noted that the choice of $u=1$ and $k=2$ achieves similar results to the previous parameters with average gaps of 0.0036, 0.0019, and 0.0083 for AC-NB, AC-NBP, and AC-PNB, respectively.

We compare the prediction results of these three AC models with negative binomial model and negative binomial regression model. In negative binomial regression model, we choose the same covariates as in Poisson regression model, and the coefficients are estimated using maximum likelihood method, which is implemented in *glm.nb*( ) in R (see https://stat.ethz.ch/R-manual/R-devel/library/MASS/html/glm.nb.html for details).

The results of comparing the performance of AC-NB, AC-PNB, AC-NBP, negative binomial, and negative binomial regression models are given in Table 4. The results indicate that AC-NB is the best model to use uniformly. Specifically, AC-NB model outperforms

Table 3. Yield Prediction Comparison Results: AC-Poisson with Poisson Model and Poisson Regression Model

Wafer map	1 (%)	2 (%)	3 (%)	4 (%)
True yield	79.50	84.36	89.85	79.28
Poisson model	52.33	74.85	87.90	72.21
Relative absolute bias	34.18	11.27	2.17	8.92
Poisson regression model	55.12	76.13	88.16	72.34
Relative absolute bias	30.67	9.76	1.88	8.75
AC-Poisson model ($u=1, k=2$)	81.09	82.84	89.38	78.49
Relative absolute bias	**2.00**	**1.80**	**0.52**	**1.00**

Note. The best results are given in boldface.

Table 4. Yield Prediction Comparison Between AC-NB, AC-PNB, AC-NBP, Negative Binomial, and Negative Binomial Regression Models

Wafer map	1 (%)	2 (%)	3 (%)	4 (%)
True yield	79.50	84.36	89.85	79.28
AC-NB ($u = 0.7$, $k = 3$)	79.82	84.12	89.94	79.19
Relative absolute bias	0.40	0.27	**0.10**	**0.12**
AC-NBP ($u = 0.6$, $k = 1$)	79.33	84.47	90.54	80.06
Relative absolute bias	**0.22**	0.14	0.76	0.98
AC-PNB ($u = 0.7$, $k = 3$)	79.83	84.34	90.18	79.78
Relative absolute bias	0.41	**0.02**	0.37	0.63
Negative binomial model	76.31	83.09	89.71	78.07
Relative absolute bias	4.01	1.51	0.16	1.53
Negative binomial regression model	79.28	84.12	89.76	79.07
Relative absolute bias	0.28	0.28	**0.10**	0.26

Note. The best results are given in boldface.

other models for wafer 3 and wafer 4, AC-NBP model yields the best result for wafer 1, and AC-PNB model gives the best result for wafer 2. Compared with the negative binomial model, AC-NB model improves the prediction result by a factor between 2 and 14. Compared with negative binomial regression model, the error of AC-NB model is lower on wafers 2 and 4, about the same for wafer 3, and a bit worse for wafer 1.

In some of the cases AC-PNB and AC-NBP perform better than AC-NB because improved yield prediction can be achieved by fitting different yield models to different clusters. Combining different yield models works better in cases of unstable manufacturing processes that render different defect behaviors in different areas on the wafer. Still, AC-NB model is uniformly the most robust; therefore, it is our recommended choice.

6. Simulated Data Study

To further compare the AC model with existing models, we generate simulated wafer maps with different degrees of clustering and radial loss. Wafer maps have been simulated using scattering scheme or superposition of different defect patterns (see, e.g., Yuan et al. 2011, Bae et al. 2007, Hansen et al. 1997). We adopt here the generalized linear mixed model (GLMM) to generate simulated samples because it easily captures both the inhomogeneity and spatial dependence of defects, which is difficult to model using either scattering or superposition method. GLMM has been applied in the analysis of spatial correlated count data, as shown in Christensen and Waagepetersen (2002), Park and Lord (2007) and Chib and Winkelmann (2012).

In the simulation, the number of defects on chip i follows Poisson distribution with density λ_i, which is related to covariates with the canonical logarithmic function. As radial loss is significant in semiconductor manufacturing, the radial distance, r_i, is used as a covariate, while no other spatial variables are considered:

$$\log(\lambda_i) = \beta_0 + \beta_1 r_i + s_i. \tag{11}$$

Here $\mathbf{s} = (s_1, \ldots, s_n)$ is a random vector that follows a Gaussian distribution with a specified covariance matrix Σ: $\mathbf{s} \sim N(0, \Sigma)$. To model the spatial dependency between chips and thus generate defect clusters, Σ is designed with the conditional autoregressive model (CAR), resulting in a type of Gaussian Markov random field (GMRF). Under the four-neighborhood system, the inverse of covariance matrix $Q = \Sigma^{-1}$ has the following structure:

$$Q_{ij} = \begin{cases} 4p & i = j, \\ -p & j \in N(i), \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

where p is called precision parameter. Smaller p indicates greater neighborhood effects and thus generates more clustered defects. Additional details about GMRF and CAR can be found in Lichstein et al. (2002) and Rue and Held (2005). Our simulation test consists of two parts. The first part simulates wafer maps with different degrees of clustering but with no radial loss, i.e., $\beta_1 = 0$, while the second part simulates samples with radial loss, i.e., $\beta_1 > 0$. The values of β_0 and β_1 are selected such that the simulated wafer maps have similar number of defects to real wafer maps.

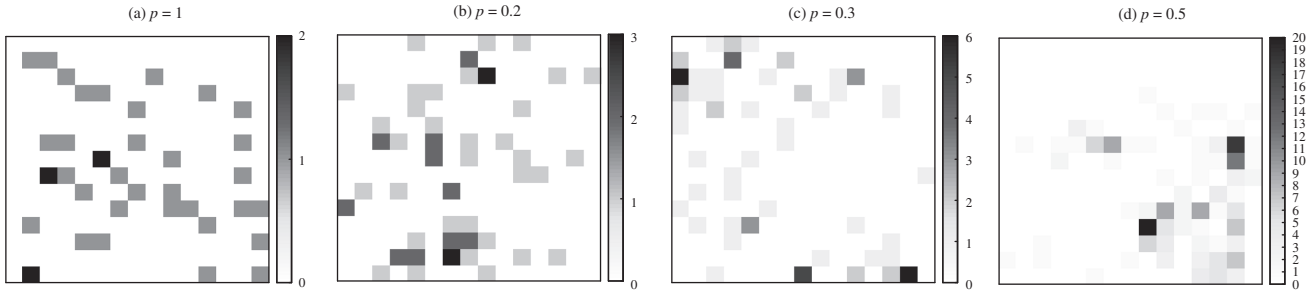
Each simulated wafer contains $15 \times 15 = 225$ chips, and parameters are chosen to be $(\beta_0, \beta_1) = (-2, 0)$ for cases with radial loss and $(\beta_0, \beta_1) = (-2, 0.1)$ for cases without radial loss. Figure 5 demonstrates four wafer maps simulated by GLMM with different values of p, in which smaller p implies greater spatial correlation and thus higher degree of clustering. For each p, we generate 100 simulated wafer maps.

It is noted that our adjacency-clustering algorithm is scalable for large wafer maps. For instance, for a wafer map containing $100 \times 100 = 10{,}000$ chips, the AC model is solved to optimality within 3.36 seconds. For the purpose of generating insights from the simulation, it is sufficient to use simulated wafers containing small number of chips, for example, $15 \times 15 = 225$ chips.

6.1. Parameter Selection.

The AC model requires the setting of the two parameters, u and k. Similar to other data analytics methods and machine learning techniques, the combination of u and k can be selected through training on given training data sets (samples produced in the initial manufacturing stage or representative historical samples). Specifically, we identify the best combination of u and k that minimizes the training error. To mimic the practical prediction tasks in integrated circuit manufacturing, we perform two different training procedures to

Figure 5. Simulated Wafer Maps Without Radial Loss for Varying Values of p



select u and k on simulated wafer maps. (1) *One-sample training*—among 100 simulated wafer maps, we pick as the training set one wafer map at a time and the other 99 maps as the test set. The AC model is fitted to the training set to find the best combination of u and k, which is then applied to the test set and the mean absolute percentage error (MAPE) is calculated. We report as the error the average MAPE across these 100 runs. (2) *Twofold training*—we select a random subset of size 50 out of the 100 simulated wafer maps to serve as training set, and the complement set serves as test set. Then the MAPE is evaluated on the test set. Next the roles of the same two sets are reversed, with the second one serving as training and the first as test set. The average of these two MAPE values is then reported. These two training procedures are two different types of cross-validation designed to measure the actual prediction performance of AC models. It is noted that the reported errors are test errors instead of training errors.

In the next two sections we evaluate the performance of the AC models with the two training procedures compared to the default setting of $u = 1$ and $k = 2$ on the simulated data. AC with these three parameter selection procedures are compared to Poisson model, Poisson regression and negative binomial models. In Section 6.2 we analyze AC-Poisson with parameters selected by training and the default setting, and compare their performance with Poisson and Poisson regression model. In Section 6.3 we test the performance of AC with other yield models.

6.2. Testing AC-Poisson on Simulated Data

In this section we compare AC-Poisson to Poisson on simulated maps. We are not testing the regression model here because the regression is on the radial distance and in the simulated data here $\beta_1 = 0$, which means that there is no radial effect.

The results given in Table 5 are in terms of MAPE. The two training procedures are performed, and the corresponding cross-validation errors are presented for AC-Poisson. We also report the prediction error of the default setting. As shown in Table 5, AC-Poisson model provides significantly better prediction results than the Poisson model. As expected, the gap between the two narrows as the clustering becomes less pronounced in the simulated data, which is measured by the increasing value of the precision parameter p. For highly clustered wafer maps ($p = 0.2$), our model reduces the prediction error by a factor of 50, compared with Poisson model. In addition, twofold training provides lower errors than one-sample training, which is explained by the larger size of the training data set. Compared with the twofold training, the default setting gives almost the same prediction results, which provides additional evidence to support the use of this combination in the context of integrated circuit manufacturing.

Next we consider simulated data with radial loss, for instance, $\beta_1 = 0.1$. Here we compare AC-Poisson with both Poisson and Poisson regression models, where the only covariate for Poisson regression model is $\{r\}$, the radial distance of a chip (and no angle-dependent variables). Table 6 displays the comparison results, in terms of the error measure MAPE, of AC-Poisson and Poisson and Poisson regression models.

As seen in Table 6, AC-Poisson outperforms Poisson and Poisson regression model with the two training procedures and the default setting. As expected, Poisson regression model provides better prediction accuracy than Poisson model, which can be explained by the fact that Poisson model does not relate the defect density to radial distance. Both models, however, are significantly inferior to AC-Poisson, in terms of the error. Overall, twofold training leads to the best prediction results, and the default setting gives very close results to the two-fold training. This validates the

Table 5. Mean Absolute Percentage Error Comparison Results Between AC-Poisson and Poisson Model for Simulated Wafer Maps ($\beta_0 = -2$, $\beta_1 = 0$)

Precision (p)	0.2 (%)	0.3 (%)	0.5 (%)	1 (%)
AC-Poisson model (one-sample training)	1.81	1.64	1.22	0.99
AC-Poisson model (twofold)	**0.81**	**0.98**	**0.84**	**0.68**
AC-Poisson model ($u = 1$, $k = 2$)	**0.81**	**0.98**	0.85	**0.68**
Poisson model	40.42	19.27	6.93	1.81

Note. The best results are given in boldface.

Table 6. Mean Absolute Percentage Errors Comparison Results Between AC-Poisson, Poisson, and Poisson Regression Model for Simulated Wafer Maps with Radial Loss ($\beta_0 = -2$, $\beta_1 = 0.1$)

Precision (p)	0.2 (%)	0.3 (%)	0.5 (%)	1 (%)
AC-Poisson model (one-sample training)	1.97	2.39	2.49	1.70
AC-Poisson model (twofold)	1.27	**1.29**	**1.43**	**1.18**
AC-Poisson model ($u = 1$, $k = 2$)	**1.23**	1.41	1.60	1.36
Poisson model	62.21	41.87	18.90	5.77
Poisson regression model	45.82	31.53	14.88	4.75

Note. The best results are given in boldface.

setting of $u = 1$ and $k = 2$ as a good choice in the absence of training data.

6.3. Testing AC-NB, AC-NBP, and AC-PNB on Simulated Data

In this section, we extend our discussion to AC models with the negative binomial yield model. The MAPE of AC-NB, AC-NBP, and AC-PNB as well as the negative binomial model for data sets with and without radial loss are presented in Tables 7 and 8, respectively. Similarly, we consider three parameter settings for AC models: two with our training procedures and one with the default setting. It should be mentioned that for the simulated wafer maps with radial effect, we do not construct a negative binomial regression model as the iteratively reweighted least squares (IRLS) algorithm fails to converge for many simulated maps. The lack of convergence has been noted previously in the literature (for more details see Marschner 2011). This phenomenon worsens as the neighborhood effect becomes more prominent in our simulation.

Table 7. Mean Absolute Percentage Errors Comparison Results Between AC-NB, AC-NBP, AC-PNB, and Negative Binomial Model for Simulated Wafer Maps ($\beta_0 = -2$, $\beta_1 = 0$)

Precision (p)	0.2 (%)	0.3 (%)	0.5 (%)	1 (%)
AC-NB (one-sample training)	1.50	0.87	0.45	0.22
AC-NB (twofold)	0.92	**0.37**	**0.23**	**0.18**
AC-NB ($u = 1, k = 2$)	1.64	0.58	0.28	0.19
AC-NBP (one-sample training)	1.69	1.53	0.81	0.44
AC-NBP (twofold)	0.74	0.69	0.49	0.21
AC-NBP ($u = 1, k = 2$)	**0.69**	0.71	0.60	0.42
AC-PNB (one-sample training)	2.42	1.83	1.02	0.78
AC-PNB (twofold)	1.26	0.74	0.58	0.68
AC-PNB ($u = 1, k = 2$)	1.85	0.98	0.77	0.63
Negative binomial model	8.42	4.93	1.33	0.28

Note. The best results are given in boldface.

Table 8. Mean Absolute Percentage Errors Comparison Results Between AC-NB, AC-NBP, AC-PNB, and Negative Binomial Model for Simulated Wafer Maps ($\beta_0 = -2$, $\beta_1 = 0.1$)

Precision (p)	0.2 (%)	0.3 (%)	0.5 (%)	1 (%)
AC-NB (one-sample training)	2.86	1.38	1.01	0.51
AC-NB (twofold)	2.10	**0.94**	**0.48**	**0.32**
AC-NB ($u = 1, k = 2$)	3.74	1.91	0.73	0.34
AC-NBP (one-sample training)	1.78	1.93	1.97	1.01
AC-NBP (twofold)	1.16	1.02	1.03	0.63
AC-NBP ($u = 1, k = 2$)	**1.03**	1.23	1.22	0.96
AC-PNB (one-sample training)	3.55	2.40	1.96	0.92
AC-PNB (twofold)	2.19	1.15	1.04	0.90
AC-PNB ($u = 1, k = 2$)	3.66	1.98	1.19	0.92
Negative binomial model	16.58	11.93	4.45	1.31

Note. The best results are given in boldface.

From the comparison results we conclude that both AC-NB and AC-NBP models outperform the AC-PNB model and negative binomial model for the simulated data set without radial effects. For the simulated wafer maps with radial effects, AC-NB, AC-NBP, and AC-PNB all provide smaller prediction errors than the negative binomial model. On both simulated data sets, AC-NB model is the leading model in terms of the lowest errors for most cases. For these AC models, the default setting of $u = 1$ and $k = 2$ has similar prediction results to the twofold training results. The default setting provides better results than the one-sample training in most simulations, which implies that it is the combination to select unless sufficient data is available for training.

7. Conclusions

We introduce the adjacency-clustering (AC) model for yield prediction that takes into account a neighborhood effect. We demonstrate that this model delivers significant improvements in prediction accuracy compared to state-of-the-art statistical approaches. The empirical evidence is based on runs for real data sets and simulated data sets. The AC model is parametrized by the selection of two parameters that could be tuned for specific purposes. Nevertheless, we show that even making a default selection of values $u = 1$ and $k = 2$ still delivers high quality prediction results that substantially improve on existing techniques. The AC model applies a polynomial time algorithm to obtain clusters efficiently and thus can be available for online monitoring and other practical uses. Although it fits classical yield model for each cluster, the yield prediction result of adjacency-clustering model exhibits significant improvement in the accuracy compared with classical models that do not differentiate between clusters. We also observe that the scheme of fitting different yield models to clusters with different yield levels can further increase the accuracy. This observation implies that different clusters in a wafer may have different types of mechanisms of generating defects. In practice, historical data can be used as the training set to select the two parameters, u and k. Our simulation results show that AC models work well even with very small training set. Also, through evaluation on both real and simulated data sets, we find out that the combination of $u = 1$ and $k = 2$ leads to superior prediction performance.

Apart from yield prediction, the adjacency-clustering model can be used to evaluate the extent of clustering of defects on wafer maps, where larger objective values correspond to higher segregation, or separation, of clustering of defects. This may be helpful in the quality control of a manufacturing process.

Compared with existing regression models in the literature, our model presents not only improved prediction accuracy but also other advantages: First, our model is free from coefficient estimation, which remains challenging for regression models based on complicated distributions or discrete hidden Markov models, especially when handling large-scale data. Second, our model is highly flexible and can be applied to wafers with various spatial patterns, since the spatial pattern is naturally captured by the solution to the adjacency-clustering model. In contrast, regression models necessarily require covariate selection, and this selection is increasingly difficult as wafers exhibit more complicated spatial patterns. The success of the technique of adjacency-clustering presented in this paper bodes well for its applications to other contexts where a neighborhood effect is manifested, for example, energy consumption prediction and disease mapping.

Acknowledgments

The authors are grateful to Chung-Piaw Teo (the department editor), the associate editor, and two anonymous referees for their very valuable suggestions. The authors also thank Kaibo Wang of Tsinghua University and the attendees of the 2015 INFORMS Annual Meeting (Philadelphia) for insightful discussions.

References

Ahuja RK, Hochbaum DS, Orlin JB (2003) Solving the convex cost integer dual network flow problem. *Management Sci.* 49(7): 950–964.

Albin SL, Friedman DJ (1989) The impact of clustered defect distributions in IC fabrication. *Management Sci.* 35(9):1066–1078.

Bae SJ, Hwang JY, Kuo W (2007) Yield prediction via spatial modeling of clustered defect counts across a wafer map. *IIE Trans.* 39(12):1073–1083.

Baker KJ, Rylatt RM (2008) Improving the prediction of UK domestic energy-demand using annual consumption-data. *Appl. Energy* 85(6):475–482.

Blake A, Zisserman A (1987) *Visual Reconstruction* (MIT Press, Cambridge, MA).

Chandran BG, Hochbaum DS (2009) A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Oper. Res.* 57(2):358–376.

Chao LC, Tong LI (2009) Wafer defect pattern recognition by multiclass support vector machines by using a novel defect cluster index. *Expert Systems Appl.* 36(6):10158–10167.

Charras-Garrido M, Abrial D, De Goër J, Dachian S, Peyrard N (2012) Classification method for disease risk mapping based on discrete hidden Markov random fields. *Biostatistics* 13(2):241–255.

Chen FL, Liu SF (2000) A neural-network approach to recognize defect spatial pattern in semiconductor fabrication. *IIEEE Trans. Semiconductor Manufacturing* 13(3):366–373.

Chib S, Winkelmann R (2012) Markov chain Monte Carlo analysis of correlated count data. *J. Bus. Econom. Statist.* 19(4):428–435.

Christensen OF, Waagepetersen R (2002) Bayesian prediction of spatial count data using generalized linear mixed models. *Biometrics* 58(2):280–286.

Cunningham JA (1990) The use and evaluation of yield models in integrated circuit manufacturing. *IIEEE Trans. Semiconductor Manufacturing* 3(2):60–71.

Di Palma F, De Nicolao G, Miraglia G, Pasquinetti E, Piccinini F (2005) Unsupervised spatial pattern classification of electrical-wafer-sorting maps in semiconductor manufacturing. *Pattern Recognition Lett.* 26(12):1857–1865.

Fellows HH, Mastrangelo CM, White KP Jr (2009) An empirical comparison of spatial randomness models for yield analysis. *Electronics Packaging Manufacturing, IEEE Trans.* 32(2):115–120.

Ferris-Prabhu AV (1992) *Introduction to Semiconductor Device Yield Modeling* (Artech House, Boston).

Ferris-Prabhu AV, Smith LD, Bonges HA, Paulsen JK (1987) Radial yield variations in semiconductor wafers. *Circuits Devices Magazine, IEEE* 3(2):42–47.

Gallo G, Grigoriadis MD, Tarjan RE (1989) A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.* 18(1): 30–55.

Ghosh SK, Mukhopadhyay P, Lu JCJ (2006) Bayesian analysis of zero-inflated regression models. *J. Statist. Planning Inference* 136(4):1360–1375.

Goldberg AV, Tarjan RE (1988) A new approach to the maximum-flow problem. *J. ACM* (*JACM*) 35(4):921–940.

Hansen MH, Nair VN, Friedman DJ (1997) Monitoring wafer map data from integrated circuit fabrication processes for spatially clustered defects. *Technometrics* 39(3):241–253.

Held K, Kops ER, Krause BJ, Wells WM, Kikinis R, Muller-Gartner HW (1997) Markov random field segmentation of brain MR images. *IEEE Trans. Medical Imaging* 16(6):878–886.

Hochbaum DS (2001) An efficient algorithm for image segmentation, markov random fields and related problems. *J. ACM* (*JACM*) 48(4):686–701.

Hochbaum DS (2008) The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Oper. Res.* 56(4):992–1009.

Hwang JY, Kuo W (2007) Model-based clustering for integrated circuit yield enhancement. *Eur. J. Oper. Res.* 178(1):143–153.

Ishikawa H, Geiger D (1998) Segmentation by grouping junctions. Spencer R, ed. *Comput. Vision Pattern Recognition, 1998 Proc.*, 1998 *IEEE Comput. Soc. Conf.*, (IEEE, Los Alamitos, CA), 125–131.

Jeong YS, Kim SJ, Jeong MK (2008) Automatic identification of defect patterns in semiconductor wafer maps using spatial correlogram and dynamic time warping. *IEEE Trans. Semiconductor Manufacturing* 21(4):625–637.

Kim KO (2011) Burn-in considering yield loss and reliability gain for integrated circuits. *Eur. J. Oper. Res.* 212(2):337–344.

Kim T, Kuo W (1999) Modeling manufacturing yield and reliability. *IEEE Trans. Semiconductor Manufacturing* 12(4):485–492.

Koren I, Koren Z, Stepper C (1993) A unified negative-binomial distribution for yield analysis of defect-tolerant circuits. *IEEE Trans. Comput.* 42(6):724–734.

Krueger D, Montgomery D (2014) Modeling and analyzing semiconductor yield with generalized linear mixed models. *Appl. Stochastic Models Bus. Indust.* 30(6):691–707.

Law AM (2014) *Simulation Modeling and Analysis* (McGraw-Hill, New York).

Li TS, Huang CL (2009) Defect spatial pattern recognition using a hybrid SOM–SVM approach in semiconductor manufacturing. *Expert Systems Appl.* 36(1):374–385.

Lichstein JW, Simons TR, Shriner SA, Franzreb KE (2002) Spatial autocorrelation and autoregressive models in ecology. *Ecological Monographs* 72(3):445–463.

Marschner IC (2011) glm2: Fitting generalized linear models with convergence problems. *R J.* 3(2):12–15.

Milor L (2013) A survey of yield modeling and yield enhancement methods. *IEEE Trans. Semiconductor Manufacturing* 26(2): 196–213.

Murphy BT (1964) Cost-size optima of monolithic integrated circuits. *Proc. IEEE* 52(12):1537–1545.

Ooi MPL, Sok HK, Kuang YC, Demidenko S, Chan C (2013) Defect cluster recognition system for fabricated semiconductor wafers. *Engrg. Appl. Artificial Intelligence* 26(3):1029–1043.

Panjwani DK, Healey G (1995) Markov random field models for unsupervised segmentation of textured color images. *IEEE Trans. Pattern Anal. Machine Intelligence* 17(10):939–954.

Park E, Lord D (2007) Multivariate Poisson-lognormal models for jointly modeling crash frequency by severity. *Transportation Res. Record: J. Transportation Res. Board* 2019:1–6.

Rue H (2001) Fast sampling of Gaussian Markov random fields. *J. Roy. Statist. Soc. Ser. B* (*Statist. Methodology*) 63(2):325–338.

Rue H, Held L (2005) *Gaussian Markov Random Fields: Theory and Applications* (CRC Press, Boca Raton, FL).

Seeds R (1968) Yield and cost analysis of bipolar LSI. *IEEE Trans. Electron Devices* 15(6):409.

Stapper CH (1989) Large-area fault clusters and fault tolerance in VLSI circuits. *IBM J. Res. Development* 33(2):162–173.

Stapper CH, Armstrong FM, Saji K (1983) Integrated circuit yield statistics. *Proc. IEEE* 71(4):453–470.

Szeliski R, Zabih R, Scharstein D, Veksler O, Kolmogorov V, Agarwala A, Tappen M, et al. (2008) A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Machine Intelligence* 30(6):1068–1080.

Taam W, Hamada M (1993) Detecting spatial effects from factorial experiments: An application from integrated-circuit manufacturing. *Technometrics* 35(2):149–160.

Tsai WJ, Tong LI, Wang CH (2008) Developing a new defect cluster index. *J. Chinese Inst. Indust. Engineers* 25(1):18–30.

Tyagi A, Bayoumi MA (1994) The nature of defect patterns on integrated-circuit wafer maps. *IEEE Trans. Rel.* 43(1):22–29.

Veksler O (2007) Graph cut based optimization for MRFs with truncated convex priors. Lucey S, Chen T, eds. *Comput. Vision and Pattern Recognition,* 2007, *CVPR'07, IEEE Conf.* (IEEE, Minneapolis), 1–8.

White KP Jr, Kundu B, Mastrangelo CM (2008) Classification of defect clusters on semiconductor wafers via the hough transformation. *IEEE Trans. Semiconductor Manufacturing* 21(2):272–278.

Wu MJ, Jang JSR, Chen JL (2015) Wafer map failure pattern recognition and similarity ranking for large-scale data sets. *IEEE Trans. Semiconductor Manufacturing* 28(1):1–12.

Yuan T, Bae SJ, Park JI (2010) Bayesian spatial defect pattern recognition in semiconductor fabrication using support vector clustering. *Internat. J. Adv. Manufacturing Tech.* 51(5-8):671–683.

Yuan T, Ramadan SZ, Bae SJ (2011) Yield prediction for integrated circuits manufacturing through hierarchical Bayesian modeling of spatial defects. *IEEE Trans. Rel.* 60(4):729–741.

Dorit S. Hochbaum is a Chancellor Full Professor in the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. Her research interests are in the areas of discrete optimization, network flow techniques, data mining, image segmentation, supply chain management, and efficient utilization of resources. In 2004 she received an honorary doctorate of sciences from the University of Copenhagen for her work on approximation algorithms. She is an INFORMS fellow and a SIAM fellow.

Sheng Liu is a doctoral candidate in the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. His research interests include data-driven optimization, supply chain management, and sharing economy.