A "Defect Level Versus Cost" System Tradeoff for Electronics Manufacturing

Michael Scheffler, *Member, IEEE*, Paul D. Franzon, *Senior Member, IEEE*, and Gerhard Tröster, *Senior Member, IEEE*

***Abstract*—Both cost and quality are important features when manufacturing today's high-performance electronics. Unfortunately, the two design goals (low) cost and (high) quality are somewhat mutually exclusive. High testing effort (and thus, quality) comes with a considerable cost, and lowering test activities has significant impact on the delivered quality. In this paper, we present a new structured search method to obtain the best combination of these two goals. It features a Petri-net oriented cost/quality modeling approach and uses a Pareto chart to visualize the results. The search for the Pareto-optimal points is done by means of a genetic algorithm. With our method, we optimize a manufacturing process for a global positioning system (GPS) front end. The optimized process clearly outperformed the standard fabrication process.**

***Index Terms*—Cost quality tradeoff, genetic algorithm, high-density packaging (HDP), Pareto chart.**

I. Introduction

It is a well-known fact that final shipment quality of an electronic product plays the same important role for customer satisfaction as the cost of this product. For the customer, the target is clear: Quality as high and cost as low as possible.

However, both targets are linked adversely: Exhaustive and expensive testing will lead to a very high quality level, but also to high final cost per shipped unit. On the other hand, neglecting a full functional test before shipping yields unacceptable delivery quality. Some rules-of-thumb exist to achieve a compromise. Fig. 1 outlines the QUALITATIVE relation of cost and quality, in this case describing testability versus area overhead [1]. Note that at a certain (system-specific) level, the cost–quality relation changes its behavior from quasi-linear rise to asymptotic approximation toward 100% quality. This means that beyond a certain level, even large investments result only in very small quality improvement. Whereas the cost–quality relation is closely monitored, e.g., during integrated circuit (IC) development to obtain an optimum, similar efforts on the system level are unknown.

However, cost and quality of a system are highly dependent on its components and testability concepts, which are usually defined during the product specification phase. *A priori*, no unique optimal concept is available, instead, various options can be found in an electronics manufacturing process, especially when high-density packaging (HDP) technologies are involved: For bare-die components, some companies offer different test levels from untested to known-good dies (KGDs); the fault coverage varies with the test vector length; during the assembly process, additional optical inspection or in-circuit test could be beneficial, etc. All these options translate into a range of applicable fault coverages and yield numbers.

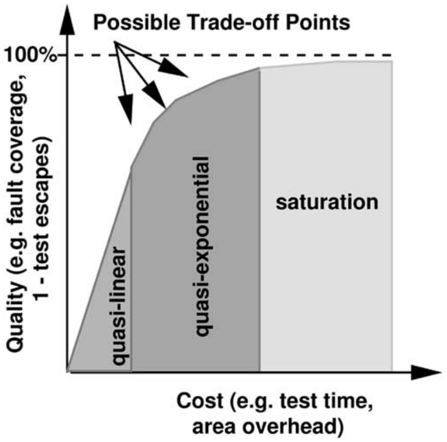


Fig. 1. Typical rules-of-thumb relationship between quality and test cost: after a quasi-linear start, cost changes pick up speed, and finally, quality improvements run into saturation.

Therefore, being faced with a specific design problem with its individual options, for a designer, it is vital to know the QUANTITATIVE cost–quality relation as early as possible in the design cycle to design a system with achievable cost/quality goals meeting its specifications.

Existing Solutions: To date, no cost modeling (CM) tool is able to perform such a concurrent optimization of quality and cost, simply because in most cases quality aspects are not included. Moreover, most of the CM tools are unable to map new technologies such as system-on-a-chip, integrated passives, etc. since they are tailored to particular processes. Also, none of them feature any optimization strategy, which has been done so far by trial and error.

Dislis [2] uses a spreadsheet to calculate cost and quality for a specific KGD fabrication setup. Adapting this method to other processes requires a rewrite of the entire sheet. Moreover, no tradeoff procedure is available.

Ungar [3] used a similar approach to assess the effectiveness of automatic versus built-in self-test (BIST) equipment, claiming that using BIST drives down the repair costs, so even with a higher number of defective delivered units one can be more cost effective. This calculation was based on various assumptions on field return costs and early time-to-market gains,

Manuscript received November 26, 2002.

M. Scheffler is with the Art of Technology, Badenerstrasse 734, 8048 Zurich, Switzerland (e-mail: scheffler@art-of-technology.ch

P. D. Franzon is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh NC 27695 USA.

G. Tröster is with the Electronics Lab, ETH Zurich, 8092 Zurich, Switzerland.

Digital Object Identifier 10.1109/TEPM.2004.830513

which are difficult to estimate accurately. Moreover, due to his spreadsheet calculation model, he sampled only a small part of all possible combinations.

On the other hand, some quality engineering (QE) approaches incorporate cost considerations: The total quality cost is considered to be the sum of all appraisal/evaluation costs, the failure costs, and the prevention costs [4].

Millman [5] made some analytical elaborations on yield versus fault coverage, giving strategies, when to invest in yield and when in fault coverage improvements. His model, however, incorporates only a general cost factor and is based on a very simple processing model.

Ng and Hui [6] present a general production-related method for setting up a quality improvement plan. Since they deal with "continuous" variables such as lengths, they assume the usual quadratic quality cost function, which is not appropriate for discrete events such as defective parts. In addition, this type of method requires an already defined quality goal and does not help determining it, as desired early in the design cycle.

Birolini [7] calculates, based on given yields and fault coverages, the remaining defect level after final test (the "field returns"), and imposes a cost penalty for every defective, delivered unit (failure cost). Changing some of the fault coverages and yields, thus increasing the prevention costs, he decreased the total failure costs, winding up with a more cost-effective solution. This approach has two draw backs: First, optimization in this simple case was done by "trial-and-error," which is not viable for larger production models. Second, it is based on an assumption of 100% field returns. Recent customer surveys of electronics contract manufacturers in Germany have revealed that a significant number of customers, rather than complaining and returning the product, instead change the supplier without notice, so field return rates are grossly underestimated [8]. Also, customer satisfaction, i.e., quality and its fulfillment, have a very high value, and it is very hard to win back a dissatisfied customer. Therefore, translating such dissatisfaction into money can be misleading.

Bukovjan [1] developed an algorithm for test point insertion during an IC synthesis phase. The concurrent criteria are testability and area overhead, which are optimized using branch-and-bound, rendering the generation of a tradeoff front rather computationally intensive. In addition to that, this algorithm focuses on IC-related tests only, neglecting any manufacturing issues.

Commercial test software allows for local test tradeoffs neglecting the synergistic optimization of design (i.e., yield) and test effects.

In this paper, we present a test versus cost optimization strategy for manufacturing electronics, which overcomes the above listed drawbacks. This strategy provides test/yield recommendations for the design space to set acceptable and manufacturable goals for quality and cost. Presenting our approach, we outline first our concept of quality-enhanced cost modeling and a way to calculate the metrics cost and quality. This is followed by the methodology for an automated search algorithm for tradeoff points. Finally, we illustrate the benefits of our approach with a case study on manufacturing a global positioning system (GPS) front end.

II. Quality Enhanced Cost Modeling

The final cost of a shipped product is the sum of all costs required to manufacture it divided by the total number of shipped devices (1). The main factors contributing to this sum can be divided into three categories, **direct cost (DC)** such as components, operator cost, test and rework cost, **nonrecurring expenditure (NRE)** consisting of machine depreciation, development cost, overhead, and **reject cost (SCRAP)** caused by malfunctioning devices and determined by quality factors such as yield and fault coverage (2).

The quality of this unit (or 1—**Defect Level (DL)**) is the percentage of error-free shipped devices to the total number of devices (3). The final quality is driven by yield figures of components and processes y and the perfection of test effort (the so-called test transparency TT) throughout the production process [see (5)]. The test transparency TT depends on the fault coverage fc of preceding test steps and the yield y_{prev} of the previous manufacturing steps (see [9]). See (1)–(5) at the bottom of the page.

Instead of using spreadsheet-orientated calculations to calculate (2) and (5) and to simplify their adaptation to a specific process, we used the different approach of process-oriented cost–quality modeling.

The rationale of this approach is as follows: In general, every design/manufacturing process is described as a material or information flow, Fig. 2(a) shows the example of a mixed wire bond-SMD assembly. Material enters the flow top left and changes throughout the flow to an assembled module.

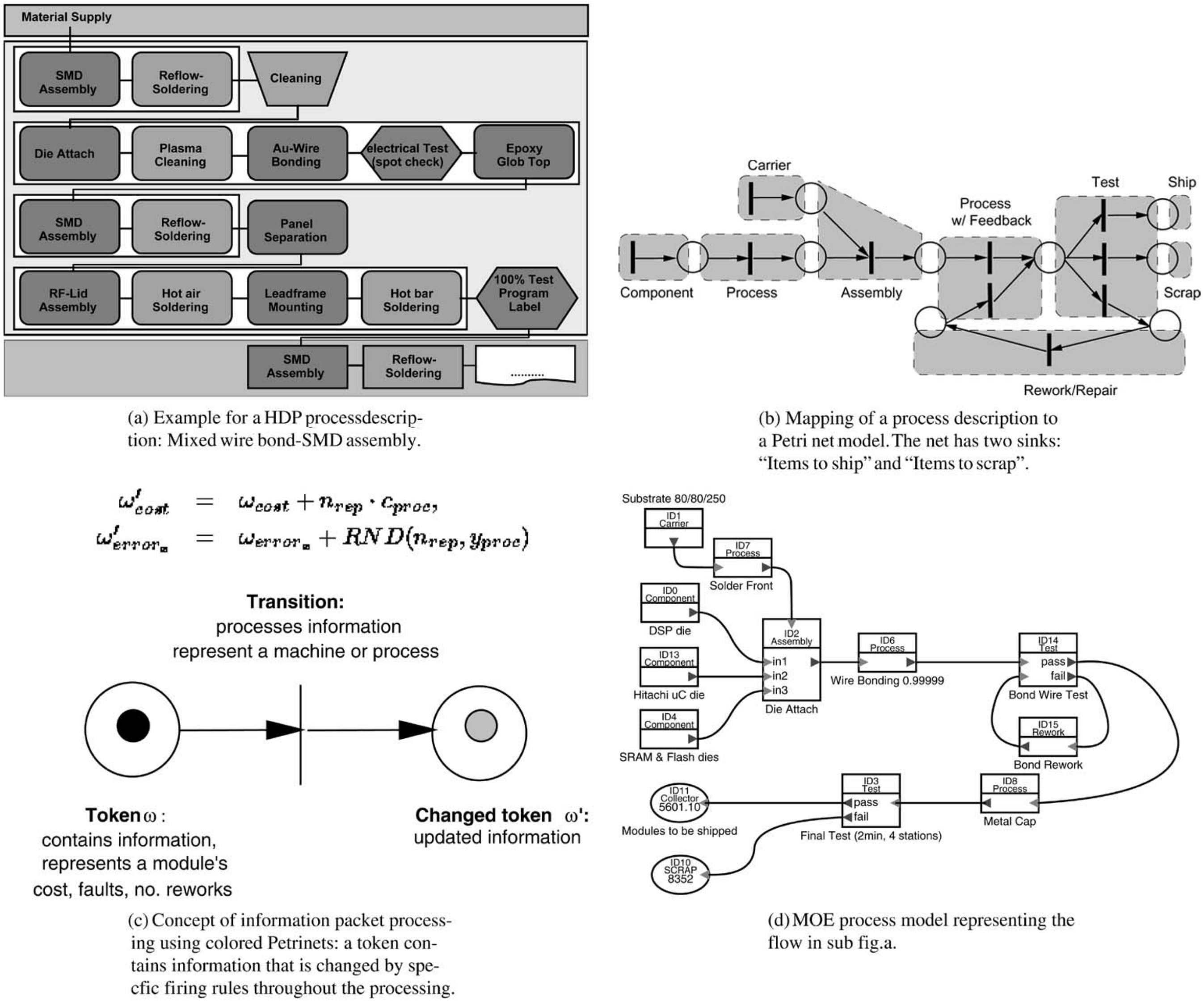
$$\text{Total Cost}_{\text{Unit}} = \frac{\text{Total Costs to manufacture given no. electronic systems}}{\text{Given no. of systems shipped}} \tag{1}$$

$$= \frac{\sum\limits_{\text{all units}} \text{DC}_{\text{Unit}} + \sum\limits_{\text{all processes}} \text{SCRAP} + \sum \text{NRE}}{\text{No. shipped Units}}. \tag{2}$$

$$\text{Quality Level}_{\text{Unit}} = \frac{\text{No. shipped Units containing no error}}{\text{No. shipped Units}} \tag{3}$$

$$= 1 - \text{DL}, \tag{4}$$

$$= 1 - \frac{(1-y)\cdot \text{TT}}{y + (1-y)\cdot \text{TT}}. \tag{5}$$



(a) Example for a HDP processdescription: Mixed wire bond-SMD assembly.

(b) Mapping of a process description to a Petri net model. The net has two sinks: "Items to ship" and "Items to scrap".

(c) Concept of information packet processing using colored Petrinets: a token contains information that is changed by specfic firing rules throughout the processing.

(d) MOE process model representing the flow in sub fig.a.

Fig. 2. Modeling an electronic manufacturing process using a Petri-net structure and the MOE graphical user interface.

Formally, a flow can be translated into a Petri net, mapping all possible routes through the net [Fig. 2(b)]. Single Petri transitions correspond to typical basic tasks in a manufacturing process, such as components entering, assembly, testing, rework/repair, etc. A manufacturing Petri net contains two types of sinks, "Items to ship," and "Items to scrap" (the latter having multiple instances).

The material status within a flow can be described using colored tokens ω, representing information on components, comparable to tags or ID numbers on the factory floor [Fig. 2(c)]. A colored token is defined as a set of variables ("properties"). In a HDP production flow we define

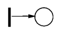
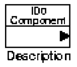
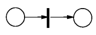
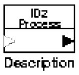
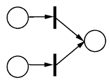
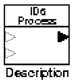
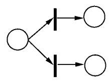
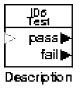
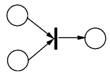
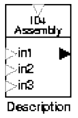
$$\omega = \{\text{cost},\ \text{error1}, \ldots, \text{error\_max},\ \text{rework\_counter}\}$$
$$\text{cost} \in \mathbb{R},\ \text{all others}\ \in \mathbb{N}. \tag{6}$$

Using specific firing rules, the tokens/components move through the flow, finally ending either as units shipped or scrapped. Evaluating the information stored within the tokens leads to cost and quality, transforming (2) and (5) to (7) and (8). Equation (7) sums up all the direct cost incurred from the units N that are routed to the sink P_{ship} (i.e., are shipped), adding the total loss cost from all scrapped units $N(P_{\text{ship}})$ and the overall NRE. Equation (8) can be seen as the ratio of the number of units to be shipped containing no errors and the total number of units in P_{ship}.

$$\begin{aligned}\text{Total Cost}_{\text{Unit}} = {} & \frac{1}{N(P_{\text{ship}})} \\ & \cdot \Biggl(\sum_{i=1}^{N(P_{\text{Ship}})} \omega_i(\text{cost}) \\ & + \sum_{\forall\ P_{\text{scrap}}} \sum_{i=1}^{N(P_{\text{scrap}})} \omega_i(\text{cost}) \\ & + \sum_{\text{all processes}} \text{NRE} \Biggr). \end{aligned} \tag{7}$$

$$\text{Quality Level}_{\text{Unit}} = \frac{1}{N(P_{\text{Ship}})} \cdot (N(P_{\text{Ship}})|_{\omega_i(\text{error})=0}). \tag{8}$$

TABLE I
ASSIGNMENT OF PETRI NET TRANSITIONS TO BASIC MANUFACTURING FLOW ITEMS

Petri Symbol	Transition type	Process element	MOE Symbol	Comment
	fire only	Component, Carrier		A fire-only transition can be interpreted as a component entering the manufacturing flow.
	simple	Process, Re-work/Repair		A simple transition is the model of a process, where a unit is processed from the input of a machine to the output. The process operates on the unit, i.e. on the information variables in the token.
	concurrent	Feed back Process		In a concurrency situation, two units compete on the utilization of a single machine, in our case with a feedback loop present. The transition has to guarantee that no deadlock and no endless looping occurs.
	conditional	Test		A conditional transition selects a transition to fire, mimicking a test station selecting the next process, depending on the device under test.
	synchronic	Assembly		Like an assembly station, this transition awaits the specified number of components (= tokens) to be sent out together. Only if the specified number of components is available, the process is completed.

To hide the Petri net complexity from the analyzing designer, the graphical user interface MOE has been developed [10]. MOE stands for Modular Optimization Environment, is a process-oriented manufacturing model, and uses the underlying colored Petri net to compute the equations above. The analogy of Petri transitions to process elements and their respective MOE symbol are shown in Table I, a more detailed description of the Petri process modeling, the firing rules, and MOE can be found in [10].

Using (7) and (8), we are able to calculate cost and quality for a manufacturing process using a single specific yield/test combinations. The next step is to compare the results for various combinations and to tradeoff the contradictory objectives.

III. Multiobjective Optimization, the Pareto Concept, and Automated Search

Generally, a cost–quality tradeoff can be considered as a multiobjective optimization problem.

Definition 1: In a general multiobjective optimization problem, for an n-dimensional parameter vector $\mathbf{x}$ and a target function vector $\mathbf{f}$ of dimension k, an objective vector $\mathbf{y}$ has to be found satisfying the following condition

$$\begin{aligned}\text{minimize}\,\mathbf{y} &= \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))\\ \text{where}\,\mathbf{x} &= (x_1, x_2, \ldots, x_n) \in \mathbf{X}\\ \mathbf{y} &= (y_1, y_2, \ldots, y_k) \in \mathbf{Y}.\end{aligned}$$

$\mathbf{X}$ is called the decision space. $\mathbf{Y}$ the objective space.

"True" Multiobjective or "Quasi-Single" Objective?: For multiobjective problems hardly any "true" multiobjective optimization technique had been available since recently [11]. What had been done so far was to aggregate multiobjective problems into single-objective target functions ("quasi-single"). One example was the use of weighting factors for the specific objectives [12]. Another way was the dedicated translation of all objectives into a single one, such as time-to-market to cost [13] or test escapes to cost [7].

The main advantage of a "quasi-single" optimization is that various solution techniques for single-objective problems do exist. Among them are analytical methods, numerical methods, e.g., linear programming techniques, or heuristic approaches. On the other hand, the drawbacks when solving a multiobjective problem using "quasi-single"-objective optimization are also numerous.

- While setting the weights, implying "decision making before search" [11], a dedicated knowledge of the search space is required to generate these weights.
- Aggregating implicitly into a single objective allows compensation of a specific underperforming objective, which is often not desired. Consider the case when for a product low quality comes with a very attractive cost. Measures preventing this result complicate the optimization procedure.
- To obtain more than one solution and to make a tradeoff decision, several independent runs are necessary where synergies cannot be exploited. Single results do not permit a design parameter space exploration to be conducted, which is one of the main goals of early design analysis.

Especially the last issue makes the case for a "real" multiobjective optimization in order to support tradeoff investigations. Doing so, we need a specific ranking mechanism for multiobjective optimizations in order to compare different solutions.

Pareto Optimality and Domination: Whereas in a single-objective optimization, two feasible parameter vectors, e.g., $\mathbf{a}$ and $\mathbf{b} \in \mathbf{X}$, can be ranked in a strict order according to a single objective function f, resulting in either $f(\mathbf{a}) \leq f(\mathbf{b})$ or $f(\mathbf{b}) \leq f(\mathbf{a})$, for multiobjective problems only partially ordering schemes are present, as introduced by Wilfredo Pareto (see [14]): a vector $\mathbf{a}$ that is at least as good as another vector $\mathbf{b}$ in all objectives and superior in at least one objective, i.e.,

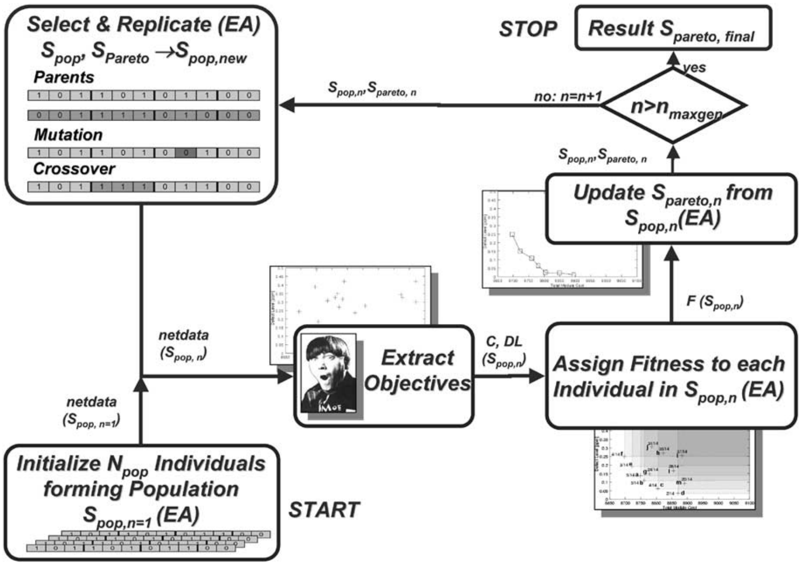

Select & Replicate (EA)
$S_{pop}, S_{Pareto} \rightarrow S_{pop,new}$
Parents
1 0 1 1 0 1 0 1 1 0 0
0 0 1 1 1 1 0 1 0 0 0
Mutation
1 0 1 1 0 1 0 0 1 0 0
Crossover
1 0 1 1 1 1 0 1 1 0 0
STOP
Result $S_{pareto, final}$
yes
$n>n_{maxgen}$
no: n=n+1
$S_{pop,n}, S_{pareto, n}$
$S_{pop,n}, S_{pareto, n}$
Update $S_{pareto,n}$ from $S_{pop,n}$(EA)
$F(S_{pop,n})$
netdata ($S_{pop, n}$)
Extract Objectives
MOE
C, DL ($S_{pop,n}$)
Assign Fitness to each Individual in $S_{pop,n}$ (EA)
netdata ($S_{pop, n=1}$)
Initialize N_{pop} Individuals forming Population $S_{pop,n=1}$ (EA)
START

Fig. 3. Interaction between the EA algorithm and the MOE tool. First, a model-specific parameter set is encoded and initialized into a population $S_{\mathrm{pop},n=1}$. Every individual is translated into a *netdata*-file to extract the objective values cost and defect level. Based on these values, fitness assignment is done and the Pareto set S_{Pareto} is updated. If the maximum number of generations has not been exceeded, a new generation is created by means of selection and replication, and the loop continues with the objective extraction.

$\mathbf{f}(\mathbf{a}) \geq \mathbf{f}(\mathbf{b})$, is said to be superior to or dominating $\mathbf{b}$. The "inferiority" condition would be $\mathbf{f}(\mathbf{b}) \geq \mathbf{f}(\mathbf{a})$. Howeverm if, e.g., $\mathbf{a}$ is only better in some objectives than $\mathbf{b}$ and worse in others, they cannot be ordered and are said to be indifferent. Vectors not being dominated by any other vector are called "Pareto vectors or points."

Automated Search: To avoid the drawbacks of a quasi-single objective optimization, we have chosen a genetic/evolutionary algorithm (GA/EA). Evolutionary algorithms (EA) are based on the Darwinist notion of a population development by means of variation and selection: Due to stochastic deviation of their composites (chromosomes or genes), individuals have a different degree of fitness with respect to their environment. When it comes to reproduction, usually the fitter individuals are preferred (selected). The interaction between the EA algorithm and the MOE tool is shown in Fig. 3.

Encoding: One of the core parts in EAs is to find a suitable encoding of the problem. Our optimization problem belongs to the class of real parameter optimizations [15], i.e., we have a given number of parameters (e.g., $y1, y2, y3, y4, fc1, fc2, fc4$) forming our decision vector according to Def. 1. This decision vector is also called individual $\mathbf{i}$ in the EA context, and each parameter can adopt either a value from a given range or from a set of values [see, e.g., (9)].

$$y1 = \{\mathrm{real\_value\_0}, \ldots, \mathrm{real\_value\_}n\} \overset{\mathrm{Example}}{=} \{0.50; 0.55; 0.61; 0.67; 0.72; 0.78; 0.84; 0.90\}. \quad (9)$$

Using (9), each parameter $\in \{y1, y2, y3, y4, fc1, fc2, fc4\}$ has been coded as a 3-bit gene, and all genes are lined up to a single bitstring ("individual"), as shown in Fig. 4.

Fitness Assignment: In a first step, a population's bitstrings are decoded to real values and fed into the MOE simulation engine to extract cost–quality data. To produce the next generation ("the offspring"), parents have to be selected in a tournament according to their fitness. The selection algorithm has been implemented adopting the Strength Pareto EA approach (SPEA) [11].

y1	y2	y3	y4	fc1	fc2	fc4

gene1	gene2	gene3	gene4	gene5	gene6	gene7

0 1 1	1 1 0	1 0 0	0 1 1	0 1 0	1 0 0	1 1 0

Fig. 4. Coding a parameter set as a one-chromosome individual; parameters are lined in an arbitrary sequence; each parameter is coded by a 3-bit gene selecting a value from its value list as in (9) (bit values in third line chosen as example).

SPEA uses the concept of elitism, where the best performing individuals (the actual Pareto points of every population) are stored in an external set S_{Pareto}. Now, aiming for fitness maximization, the fitness $F(\mathbf{i})$ of a Pareto individual $\mathbf{i}$ is the ratio of population individuals dominated by $\mathbf{i}$ to the total number of individuals. The fewer individuals there are in a certain niche, the better the fitness value. This strategy propels the search into less explored regions of the decision space.

On the other hand, the fitness $F(\mathbf{j})$ of a population individual is the sum of the fitnesses of all Pareto individuals $\mathbf{i}$ dominating it, plus one to make sure all ordinary individuals have a lower fitness than the Pareto individuals. A high number of Pareto individuals dominating this ordinary individual reduces its fitness value. The advantage of this approach is to better maintain the diversity of the population. An example for the fitness calculation is given in Fig. 5.

Summary: We are now able to extract cost/quality values from a given manufacturing specification and to rank different results with regard to these two objectives. In general, our method can be used for two different purposes: "tuning" an existing manufacturing environment and evaluating possible setups for a future fabrication process. The first one is based on more fine-grained yield/test data, and an example is given in

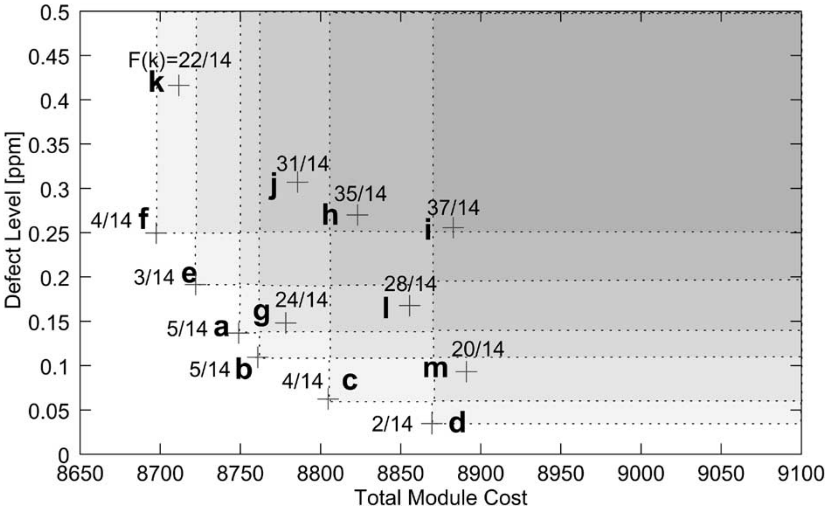
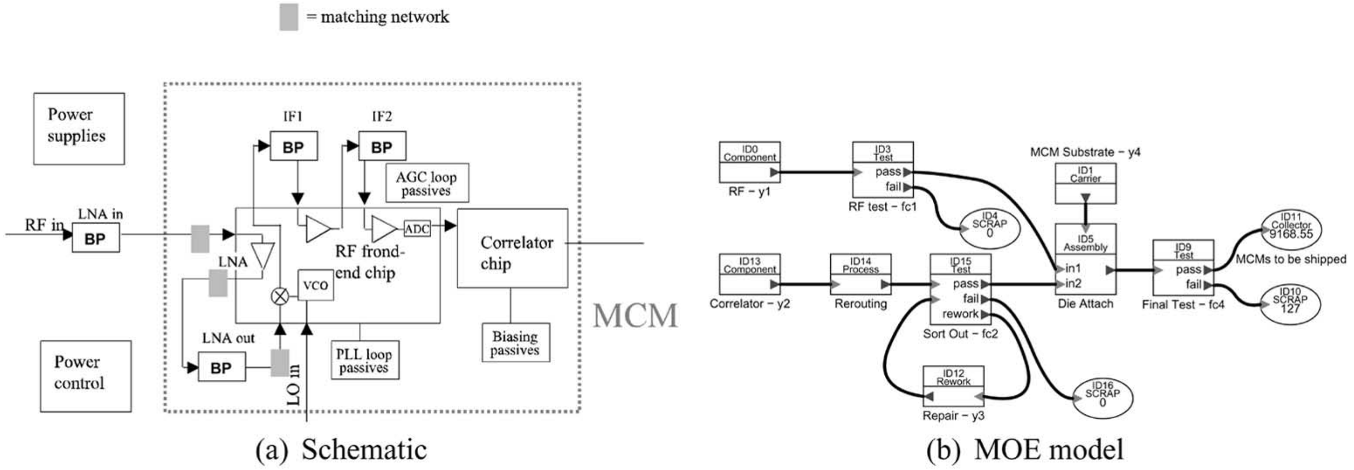


Fig. 5. Fitness calculation of SPEA: the lower the value, the better the fitness $F(\cdot)$. Pareto points dominating more population individuals, e.g., a, b get a worse fitness value than those covering a less explored region, such as d, f. From the normal population, individuals (k, m) close to those marginal points are promoted in order to support them to "boldly go where no individual has gone before."



(a) Schematic

(b) MOE model

Fig. 6. Manufacturing of a GPS RF front end.

Section IV. The second application area would explore wider ranges covering different strategies and equipments. Due to space restrictions, no example of the second application area is shown here, the interested reader may refer to [10].

IV. A Case Study: Manufacturing a GPS Module

A GPS front end converts the incoming signal after external filtering via intermediate frequencies (IF) to the base band. After A/D conversion, the signal undergoes the selection in the correlator and the subsequent stages. A schematic can be found in Fig. 6(a), a more detailed description of the module is given in [16]. The existing cost/quality values are unsatisfactory, and improvement is to be sought.

A first analysis of the manufacturing flow identified seven main parameters to be investigated, incorporating the yield figures of the IC components, the substrate, a repair process, and the fault coverages of all present tests. In Fig. 6(b), the MOE manufacturing model for the GPS front end is depicted. It includes an RF die and a correlator die. The RF $(y1)$ die undergoes a prescreening with fault coverage $fc1$, the correlator $(y2)$ is rerouted for flip chip attach and afterwards optically inspected $(fc2)$. In case of an error, this chip can be replaced once with the success rate $y3$. Both dies are attached onto the thinfilm substrate (upper right corner with $y4$), and then the entire system undergoes a functional test before shipping $(fc4)$.

Situation: For simplicity's sake, we set arbitrary maximum and minimum values for the parameters $\{y1, y2, y3, y4, fc1, fc2, fc4\}$ and calculated numerically intermediate values. The values for the parameters and their respective cost are shown in Table II. A more elaborate (and extensive) example showing more realistic yield/fault coverage values from a technical point of view can be found in [10]. Technically, higher yield and higher fault coverages can be explained with higher quality components and longer test times. Since an existing fabrication environment is to be changed, again for simplicity's sake in this case study we ruled out additional investment affecting the NRE cost.

We calculated cost and defect level and found the result of 11 624.90 cents or arbitrary cost units (a.u.) and the defect level of 0.1114 too high. The idea is now to identify key and marginal parameters by using our approach, to leverage cost/quality by

TABLE II
PARAMETER VALUES USED FOR THE GPS FRONT END; FOR EACH MAIN PARAMETER THERE IS A DEPENDENT COST VARIABLE CHANGING WITH YIELD AND FAULT COVERAGE

	low				existing			high
	main variables (yield y and fault coverage fc)							
y1	0.500	0.557	0.614	0.671	0.729	0.786	0.843	0.900
y2	0.500	0.569	0.637	0.706	0.774	0.843	0.911	0.980
y3	0.687	0.731	0.774	0.818	0.862	0.905	0.949	0.992
y4	0.687	0.731	0.774	0.818	0.862	0.905	0.949	0.992
fc1	0.500	0.557	0.614	0.671	0.729	0.786	0.843	0.900
fc2	0.500	0.569	0.637	0.706	0.774	0.843	0.911	0.980
fc4	0.687	0.731	0.774	0.818	0.862	0.905	0.949	0.992
	dependent variables ($y\_cost = f(y)$ in a.u., $fc\_cost = f(fc)$)							
y1_cost	50	55	61	67	72	78	84	90
y2_cost	25	28	31	35	38	42	45	49
y3_cost	137	146	154	163	172	181	189	198
y4_cost	343	365	387	408	430	452	474	496
fc1_cost	50	55	61	67	72	78	84	90
fc2_cost	100	113	127	141	154	168	182	196
fc4_cost	687	730	774	817	861	905	948	992

low: minimum values from Table II,
existing: the established yield/fault coverage values,
high: maximum values from Table II.

improving key parameters while turning down marginal ones in order to reduce cost. To benchmark the EA search approach to "simple solutions," we also included the results of typical parameter combinations, marked in Table II

low: minimum values from Table II;
existing: the established yield/fault coverage values;
high: maximum values from Table II.

The "low" combination corresponds to low testing efforts and low-cost low-yield components, giving the highest overall defect level, possibly at lowest cost. On the other hand, the "high" value represents the opposite with maximum test effort and high-quality components, yielding the lowest defect level. The "low" and "high" combinations are two often undesirable extremes and correspond with the beginning and the end of the graph in Fig. 1. In addition, we calculated the cost–quality of 30 000 pure random parameter combinations following Table II, the same number of solutions as used with the EA (three runs containing 100 iterations with 100 individuals each), to benchmark the EA effectiveness.

Results: Fig. 7 shows the joint results from three runs. As one can see, the high-existing-low results are situated almost on a straight line moving toward the process optimum. The "low"-value (filled circle) is dominated by the existing setup from Table II (outline square), and both are surpassed by the "high" parameters (filled square). Although the "high" combination gives the lowest defect level (while the "low" combination gives the highest one), lower cost is obtainable at the price of a small increase in the defect level.

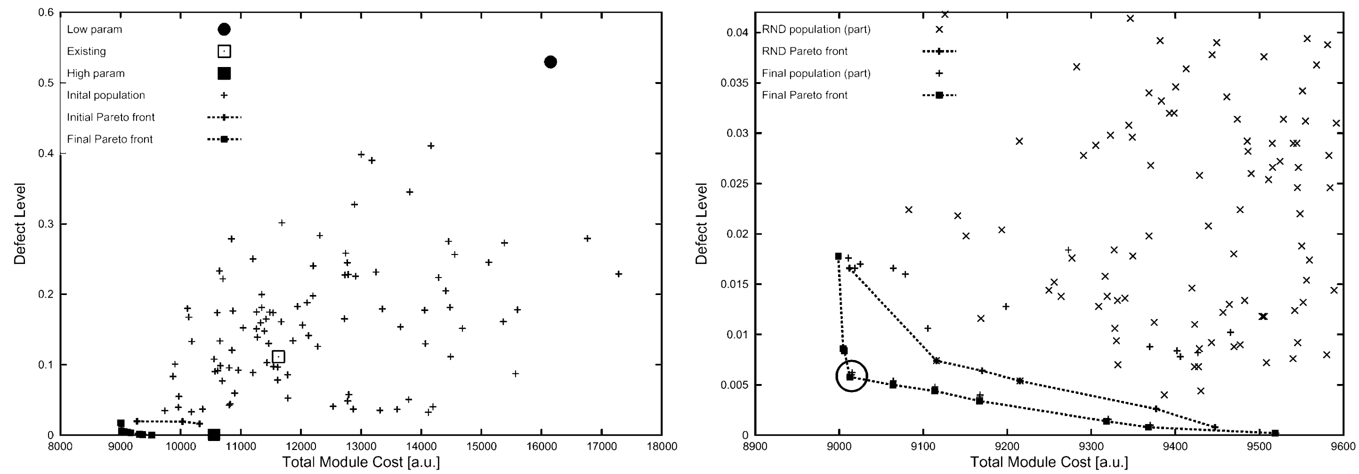
The Pareto front (small filled squares) is located in the lower left area of Fig. 7(a), and a zoom of this area is shown in Fig. 7(b). We choose the point $\{9012\ \mathrm{a.u.}; 5800\ \mathrm{ppm}\}$ as tradeoff point, marking it with a circle. Thus, compared to the "high" point with a cost of 10'500 a.u. and a defect level $<1000\ \mathrm{ppm}$, we achieve a 16% cost reduction, let alone the improvement to the previous established setup. Comparing the tradeoff point to the random run results with similar cost–quality ranking, we have achieved either a 186% defect level improvement at the same cost or a cost advantage of 2.5% choosing the same defect level.

Table III shows the decoded parameters of the Pareto front. From Table III we can see that high yield values for the RF die $(y1)$ and the correlator IC $(y2)$ are mandatory. When looking back on Table II and comparing the cost penalty for yield and fault coverage increase, the leverage is best for these two components. The influence of the rework step $(y3)$ is gradually turned down while the final fault coverage $fc4$ increases, although the cost penalty is four times higher when increasing $fc4$. Obviously, the high component yield together with good test renders it superfluous to do some repair and to invest in testing the repaired component again. Also high substrate yield $(y4)$ is imperative. The final test efficiency increases permanently, driving the defect level down the Pareto front. An interesting point is that in the higher defect level regions the low $fc4$ value is first compensated by increasing the preliminary test $fc2$ due to its better cost/performance ratio, then the final test comes into play. Thus for some applications, where only minor defect level improvements are required, a preliminary test might already suffice to move to an acceptable defect level.

V. DISCUSSION

In Section IV, we have shown a typical application example for our proposed methodology, enabling the designer to effectively screen a wide range of parameter values. In this section, we discuss some other findings.

Analysis of the Pareto Front and Its Parameters: Based on the simulations, the designer also gets a general understanding of the problem nature: Notice the two different Pareto shapes with respect to the population in Figs. 7(a) and 8. In Fig. 7(a), the min-medium-max vectors are situated on a straight line leading toward the area of the global optimum, indicating that in general



(a) The Pareto front (small filled squares) clearly outperforms the existing parameter combination.

(b) Zoom of a); trade-off point in the EA Pareto front (dashed line/small filled squares) marked with a circle. Benchmark results (dashed line/plus sign) for 30'000 individual pure-random runs are added.

Fig. 7. Results for the GPS manufacturing: comparison of the Pareto front to high-existing-low parameter combinations; the defect level varies between 0 and 1 (EA parameters are population size $N_{\rm Pop} = 100$, Pareto set size $N_{\rm Pareto} = 10$, $n_{\rm maxgen} = 100$ iterations, crossover probability $p_{\rm cross} = 0.8$, and mutation probability $p_{\rm mut} = 0.1$).

TABLE III
DECODED PARAMETERS FOR THE PARETO FRONT; TRADEOFF POINT IS MARKED

y1	y2	y3	y4	fc1	fc2	fc4	cost c [a.u.]	defect level DL
0.90	0.98	0.82	0.99	0.90	0.50	0.69	8998.86	0.0178
0.90	0.98	0.73	0.99	0.90	0.91	0.69	9004.68	0.0086
0.90	0.98	0.69	0.99	0.90	0.91	0.69	9005.93	0.0084
0.90	0.98	0.69	0.99	0.90	0.98	0.69	9012.74	0.0058
0.90	0.98	0.69	0.99	0.90	0.98	0.73	9064.02	0.005
0.90	0.98	0.69	0.99	0.90	0.98	0.77	9113.58	0.0044
0.90	0.98	0.69	0.99	0.90	0.98	0.82	9166.78	0.0034
0.90	0.98	0.69	0.99	0.90	0.98	0.95	9318.07	0.0014
0.90	0.98	0.69	0.99	0.90	0.98	0.99	9367.91	0.0008
0.84	0.98	0.69	0.99	0.90	0.98	0.99	9518.82	0.0002

all types of yield/test improvement will pay off for both cost and quality. Only a small tradeoff area exists. For the second case (Fig. 8) however, the min-medium-max line connects the undesired extremes "low cost/low quality" and "high cost/high quality." This means that a continuous tradeoff is required, since every quality improvement will result in higher total cost.

The analysis of the Pareto front provides such information. Whereas some parameters are mandatory for a globally optimal result (e.g., KGD-like components), others drive the tradeoff point down the Pareto front influencing the optimum locally (Fig. 8).

Stability: Another concern is the stability of the tradeoff point (marked with a circle in Fig. 7). We define stability in this context as variation in the object space $\{\Delta c; \Delta \mathrm{DL}\}$ as a result of parameter variations in the decision space $\{y1 \pm \Delta y1, y2 \pm \Delta y2, \ldots, fc4 \pm \Delta fc4\}$. In real-world optimization problems using statistical parameters, it is very probable that some tradeoff points are more stable than others. To quantify the susceptibility to this parameter variation, we perform a stability analysis, allowing every parameter independently to assume the adjacent lower and higher value, thus creating a subpopulation.

In a second step, we introduce a rectangle enveloping this subpopulation as the stability criterion; the larger the size of this rectangle, the less stable the point is. For a three-valued objective space, this criterion would be extended to a box containing all subpopulation members. The analysis is exemplified using the tradeoff point from Fig. 7(b).

The stability envelope compared to the previous search space is shown in Fig. 9(a), a closeup [Fig. 9(b)] shows the relation to the Pareto front. This envelope has been compared to one of the neighboring points on the Pareto front (see Table III), which could have been used as tradeoff candidates. The stability envelope of the originally chosen tradeoff point covers the smallest area, thus giving the highest stability of all candidates.

Scalability and Effort: One of the method's advantages is its applicability to a wide range of fabrication problems including standard SMD processes as well as more "exotic" processes. Prerequisite is that the manufacturing description can be broken down to a chain of basic processes, described by their specific direct cost, NRE, and yield/fault coverages. Since the computational effort[1] is somewhere between 2 h (the GPS case presented

[1] With the actual nonoptimized code.

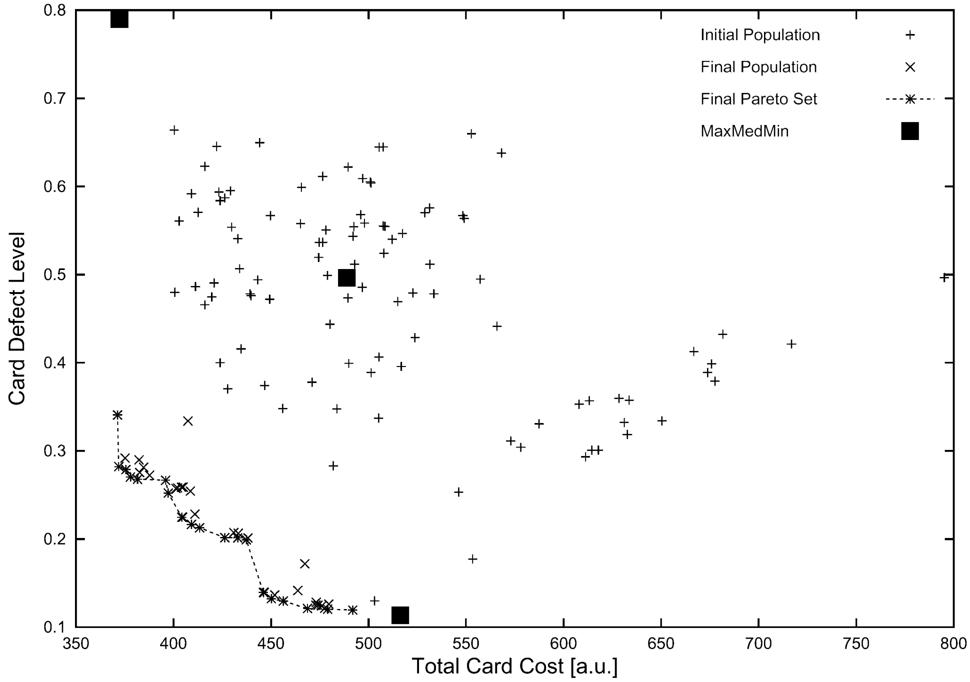
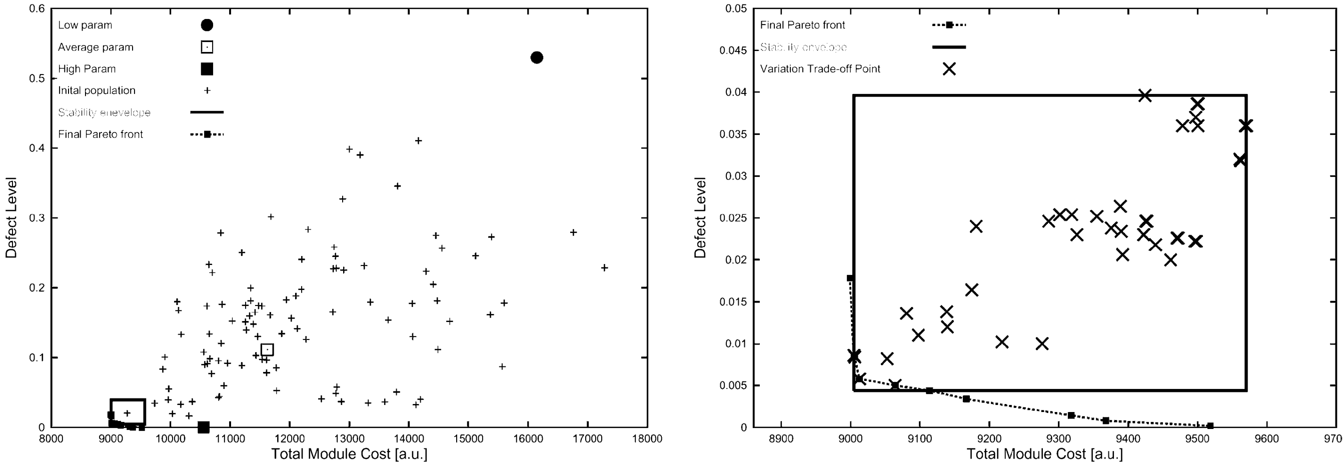


Fig. 8. Example for a second type of Pareto front.



(a) Stability region $\{\Delta c; \Delta DL\}$ for the chosen trade-off point

(b) Close up of the stability region

Fig. 9. Stability of tradeoff points: even taking parameter variations into account, the entire stability region is still a significant improvement to the existing setup (left). A close-up of the stability region (right) proved that the chosen tradeoff point has a smaller $\{\Delta c; \Delta \mathrm{DL}\}$ than its neighboring points and is thus more robust.

here) to overnight calculations (a more elaborate study on producing a smart card), depending on the number of parameters, the main analysis work remains in the gathering of accurate data.

Outlook: Future work could focus on integration of technical design options as presented in [17] to the EA algorithm. First conceptual thoughts have been made about concurrent evaluation of system-on-a-chip versus system-in-a-package manufacturing options [10]. The solution could be a built-in model switch in a chromosome as shown in Fig. 10.

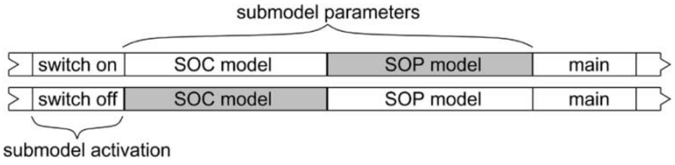


Fig. 10. Part of a one-chromosome individual similar to Fig. 4: depending on the actual switch value, a specific submodel would be selected using either an SOP or an SOC setup (selected branch marked gray).

VI. Conclusion

In this paper, we have presented a quality versus cost tradeoff method for electronic systems. The EA-based search clearly outperforms average parameter combinations and random searches in finding tradeoff options. It allows effective screening of a wide range of improvement options, and the analysis of the overall Pareto type (continuous tradeoff versus driving toward global minimum) provides valuable insight into the cost/quality relation of a specific manufacturing problem.

The methodology has been demonstrated with one case study. The process-oriented approach is flexible and can be adapted to a wide range of manufacturing processes. A stability analysis has been proposed to investigate the sensitivity of such a tradeoff point to parameter variations. With our methodology, it is now possible to identify a process setup and a component selection to achieve an optimal cost/quality combination for a HDP product.

References

[1] P. Bukovjan, "Allocation for testability in high-level synthesis," Ph.D. dissertation, Institut National Polytechnique Grenoble/Technical University Brno, Grenoble, France, 2000.

[2] C. Dislis and J. Jalowiecki, "Economics modeling and optimization of MCM test strategies," in *Proc. Europ. Design and Test Conf. 1996*, Los Alamitos, CA, 1996, p. 615.

[3] L. Ungar, "Built-in (self) test economics in a manufacturing process," in *Conf. Proc. NEPCON West*, vol. 1, 1996, pp. 218–230.

[4] B. S. Dhillon, *Quality Control, Reliability, and Engineering Design*. New York: (Industrial Engineering) Marcel Dekker, 1995.

[5] S. D. Millman, "Improving quality: Yield vs. test coverage," *J. Electron. Testing*, pp. 253–261, 1994.

[6] W. C. Ng and Y. V. Hui, "Economic design of a complete inspection plan with interactive quality improvement," *Europ. J. Operat. Res.*, pp. 122–129, 1996.

[7] A. Birolini, *Quality and Reliability of Technical Systems*, 2nd ed. Berlin, Germany: Springer-Verlag, 1997.

[8] A. Syska and F. Meckmann, "Der dienstleister und sein umfeld," *Elektronik Produktion und Prüftechnik (EPP)*, pp. 18–22, Sept. 2000.

[9] V. Agrawal, S. Seth, and P. Agrawal, "Fault coverage requirements in production testing of LSI circuits," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 57–61, Feb. 1982.

[10] M. Scheffler, "Cost vs. quality trade-off for high-density packaging of electronic systems," Ph.D. dissertation, Swiss Federal Inst. Technol., ETH Zurich, Switzerland, 2001.

[11] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, pp. 257–271, 1999.

[12] Y. Tokat, O. Palusinski, and F. Szidarovszky, "Multicriteria decision making in design of oriented wire boards," *IEEE Trans. Comp., Packag., Manufact. Technol. B*, vol. 21, pp. 73–78, 1998.

[13] J. M. Miranda, "A BIST and boundary-scan economics framework," *IEEE Des. Test Comput.*, vol. 14, no. 3, pp. 17–23, 1997.

[14] M. Blaug, Ed., "Vilfredo Pareto (1848–1923). no. 35 in Pioneers in Economiques," Edward Elgar, Aldershot, 1992.

[15] A. H. Wright, *Foundations of Genetic Algorithms*. San Matteo, CA: Morgan Kaufman, 1991, ch. Genetic Algorithms for Real Parameter Optimization, pp. 205–218.

[16] M. Scheffler, G. Tröster, J. L. Contreras, J. Hartung, and M. Menard, "Assessing the cost effectiveness of integrated passives," *Microelectron. Int.*, vol. 17, no. 3, pp. 11–15, 2000.

[17] E. Hirt, M. Scheffler, and G. Tröster, "Virtual prototyping for high-density packaging systems," *IEEE Trans. Adv. Packag.*, vol. 24, pp. 392–400, 2001.

Michael Scheffler (M'04) received the Dipl. Ing. (M.Sc.) in electrical engineering with honors from the Technical University of Berlin, Germany, in 1996 and the Ph.D. degree from the Electronics Lab, ETH Zurich, Switzerland, in 2001.

He has been a research assistant with ETH Zurich, where his work focused on cost modeling, production optimization, and technology evaluation in the field of high-density packaging. He was involved in the implementation and preparation of the EU projects SUMMIT, LAP, EASIT, and LIPS. Since 2001, he has been responsible for production at the Art of Technology, Zurich, an SME specialized on miniaturization of electronics.

Paul D. Franzon (S'85–M'88–SM'99) received the Ph.D. degree from the University of Adelaide, Adelaide, Australia, in 1988.

He is currently an Alumni Distinguished Professor with North Carolina State University. He has also been with AT&T Bell Laboratories, DSTO Australia, Australia Telecom and Communica, Ltd. His current interests center on the technology and design of complex systems incorporating VLSI, MEMS, advanced packaging, and molecular computing. Application areas currently being explored include novel advanced packaging structures, Network Processors, SOI baseband radio circuit design for deep space, on-chip inductor and inductance issues, RF MEMS, and moleware circuits and characterization. He has led several major efforts and published more than 100 papers in these areas.

Dr. Franzon received an NSF Young Investigators Award in 1993. In 2001, he was selected to join the NCSU Academy of Outstanding Teachers.

Gerhard Tröster (SM'93) received the M.S. degree from the Technical University of Karlsruhe, Germany, in 1978, and the Ph.D. degree from the Technical University of Darmstadt, Germany, in 1984, both in electrical engineering.

He is a Professor and head of the Electronics Laboratory, ETH Zürich, Switzerland. During the eight years he spent at Telefunken Corporation, Germany, he was responsible for various national and international research projects focused on key components for ISDN and digital mobile phones. His field of research includes wearable computing, reconfigurable systems, signal processing, mechatronics, and electronic packaging. He authored and coauthored more than 100 articles and holds five patents. In 1997, he cofounded the spinoff u-blox ag.