A MULTI-OBJECTIVE TEST VS. COST OPTIMIZATION FOR ELECTRONIC PRODUCTS

Michael Scheffler, Gerhard Tröster
Electronics Lab, ETH Zurich, SWITZERLAND
Gloriastr. 35, 8092 Zurich, Ph: +41 1 63 25829, scheffler@ife.ee.ethz.ch

Biography

Michael Scheffler received his M.Sc. in 1996 in Electrical Engineering from the Technical University Berlin, Germany. Since then he is with the Electronics Lab of the ETH Zürich (Swiss Federal Institute of Technology), Switzerland. He is active in the field of cost modeling for electronic system manufacturing, with special emphasis on high-density packaging solutions.

Gerhard Tröster is a professor and head of the Electronics Laboratory at ETH Zürich. His research includes signal processing, mechatronics, MCMs, electronics packaging, and wearable systems. He received his M.Sc. from the Technical University Karlsruhe, Germany, and his Ph.D. from the Technical University Darmstadt, Germany, both in Electrical Engineering.

Abstract

Both cost and quality are important features when manufacturing today's high performance electronics. Unfortunately, usually 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 sparing the test 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 graphical 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 clearly outperforming a standard fabrication set-up.

Introduction

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

But unfortunately, both figures 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 in unacceptable delivery quality. To achieve a compromise, usually ranges for quality level and total unit cost are defined within the product specification phase.

The question is now how to obtain a reasonable figure for this compromise. To reach e.g. the first target of high quality, the natural conclusion would be to push every component's yield and the fault coverage of any test as close as possible to 100%. So in the beginning, significant quality improvements come at moderate cost penalty. But when approaching 100%, test and component cost have almost an exponential behavior increasing strongly the overall cost without benefiting too much in terms of quality. Thus, we need a trade off methodology. But to date, cost modeling tools available are not able to perform such a concurrent optimization of quality and cost. In fact, none of them feature any optimization strategies, which has been done so far purely by trial and error.

In this paper we present a test vs. cost optimization strategy giving information on the design space to set acceptable and manufacturable goals for quality and cost. To do so, we outline first the idea of quality enhanced cost modeling, then followed by the methodology for an automated search algorithm for trade-off

0-7803-6482-1/2000/$10.00 ©2000 IEEE

$$\begin{aligned}
Final\ Cost_{Shipped\,Unit} &= \Sigma Direct\ Cost_{Unit} + \frac{\Sigma_{all\ steps} Cost_{SCRAP} + \Sigma NRE}{No.\ shipped\ Units}. && (1)\\
Final\ Quality_{Shipped\,Unit} &= \frac{No.\ shipped\ Units\ containing\ no\ error}{No.\ shipped\ Units} && (2)\\
&= \frac{No.\ bad\ Units\ passing\ test\ X}{No.\ overall\ Units\ passing\ X}, && (3)\\
&= \frac{(1 - Yield) * TestTransparency}{Yield + (1 - Yield) * TestTransparency} && (4)\\
&= \frac{(1 - \prod Y_{process} * \prod Y_{comp}) * TT}{\prod Y_{process} * \prod Y_{comp} + (1 - \prod Y_{process} * \prod Y_{comp}) * TT}. && (5)
\end{aligned}$$

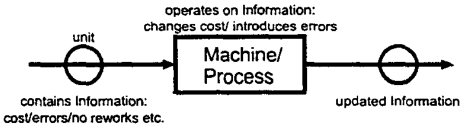

Figure 1: Concept of Information Packet Processing

points. The benefits of our approach are finally illustrated with a case study on manufacturing a GPS front end.

Quality Enhanced Cost Modeling

The main cost factors contributing to the final cost of an electronic product can be divided into three categories, **direct cost** such as components, process material, manpower cost, test and rework cost, the **non-recurring expenditure (NRE) or indirect cost** consisting of machine depreciation and overhead, e.g. management and design cost and the **reject cost or SCRAP** determined by quality factors as yield and fault coverage. All these costs have to be returned by a shipped unit leading to Equation 1 (see also [1]).

Next to this final cost, the second important Figure of Merit is the quality of this unit, the percentage of shipped units being error-free. Of course, this figure should be close to 100%. The final yield or quality results from yield figures of components and processes and the test effort (fault coverage) throughout the production process (see Eq.2).

Whereas the calculation of Equation (1) is straightforward, the computation of Equation (2) is more complicated. The final quality is the fraction of nonworking units passing the test (the so-called test escapes) and the total number of units passing this test (Equation 3). The test escapes themselves are determined by the test transparency TT or fault coverage FC of preceding test steps and the yield Y of the previous manufacturing steps (see [2]).

For a process with a single final test, the final quality turns into Equation (5), becoming even more complicated for several test stages checking only partial aspects of a unit[1].

Instead of using spreadsheet or list oriented calculations [3, 4], To hide this complexity of the underlying formulae from the design engineer and to bring the cost modeling to a more intuitive basis, we introduce a cost modeling tool based on Monte Carlo probability simulation and a graphic process representation, the Modular Optimization Environment MOE [5].

Using a packet oriented calculation, every unit under production has its own container comprising information on accumulated cost, errors, rework cycles undergone, etc. A process step ("a machine") operates on this information and changes it. The general concept is shown in Fig. 1, more detailed examples for the MOE algorithms are given below:

---

[1] X-ray testing after micro BGA soldering e.g. only checks for interconnect faults and cannot identify a non-working component.

Table I: Cost and Fault Coverage Data

	process yield	chip test cost		final test cost	
		99.2% w/o IddQ	99.7% IddQ	95% normal	97% extended
LPD chip	55-65%	8	9		
LVL2 chip	70-80%	8	9		
Final Test				10	15

Table II: Test Scenarios

<table>
<tr><th></th><th>LPD</th><th>LVL2</th><th>Final Test</th></tr>
<tr><td>A</td><td colspan="3">normal</td></tr>
<tr><td>B</td><td>IddQ</td><td colspan="2">normal</td></tr>
<tr><td>C</td><td colspan="2">IddQ</td><td>normal</td></tr>
<tr><td>D</td><td colspan="2">normal</td><td>extended</td></tr>
<tr><td>E</td><td>IddQ</td><td>normal</td><td>extended</td></tr>
<tr><td>F</td><td colspan="2">IddQ</td><td>extended</td></tr>
</table>

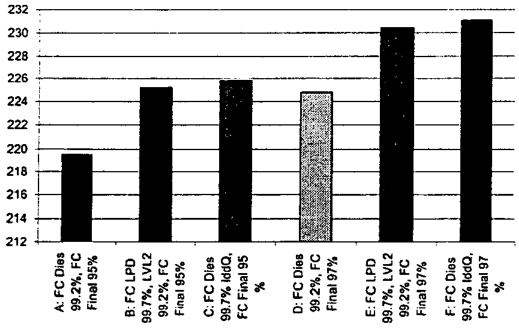

Figure 2: Result cost only: example for readout electronics

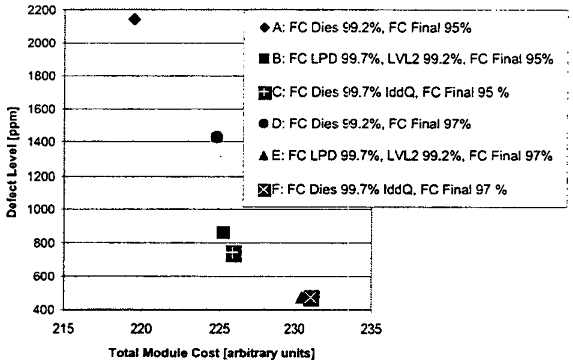

Figure 3: Result graph cost vs. quality: now a trade-off is possible

Examples of the MOE Computation Algorithms

RND() calculates a random number according to the binomial distribution. With a 90% yield, from every 100 produced units 90 are fault-free, and ten units contain an error. C denotes cost, E error, Y yield, NRE non-recurring expenditure.

Process Step:

$$\begin{aligned} C_{packet} &\mathrel{+}= C_{Process} \\ E_{packet} &\mathrel{+}= RND(Y_{Process}) \\ NRE &\mathrel{+}= NRE_{Machine} \end{aligned}$$

Assembly Step:

$$\begin{aligned} C_{new\ packet} &= N_1 * C_{packet1} + \dots + C_{Assembly} \\ E_{new\ packet} &= N_1 * E_{packet1} + \dots + RND(Y_{Assembly}) \\ NRE &\mathrel{+}= NRE_{Machine} \end{aligned}$$

Test Step:

$$\begin{aligned} C_{packet} &\mathrel{+}= C_{Test} \\ NRE &\mathrel{+}= NRE_{Machine} \end{aligned}$$

```
if $(E_{packet} = 0)$ or $((E_{packet} > 0)$ and $(RND(TT) > 0))$
then packet OK, $RewCounter_{packet} := 0$
else packet FALSE
```

The advantage is the higher level of details available on direct cost, yield loss, and NRE, giving the opportunity for cost and quality optimization. One we have this data, the next task is how to put the various cost/quality results into a right perspective. This is explained and illustrated with a little example in the next section.

An Application Example

Among the components/subsystems that can tolerate only a very low number of defect parts shipped are experimental electronics for physics or space subsystems. Typically the defect level is in the range of parts per million (ppm) as these blocks are vital for system functionality or difficult and costly to replace.

For a readout electronic subsystem e.g., the question was whether the existing test concept would fulfill the customer quality specifications and how other concepts (IddQ, enhanced final test) would perform [6]. These additional test efforts come with a considerable cost overhead (and in case of IddQ also with a design time delay), numbers can be found in Table I. An overview of the scenarios under investigation is given in Table II.

In Fig. 2 the cost calculation for the different testing scenarios are shown. According to this graph there would be no benefit in changing the test strategy be-

0-7803-6482-1/2000/$10.00 ©2000 IEEE

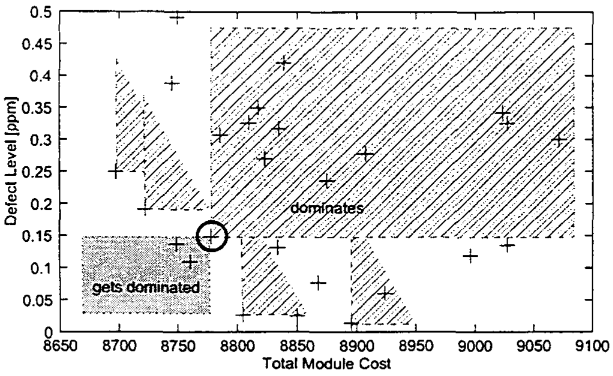

Figure 4: Dominated and Dominating Data Points

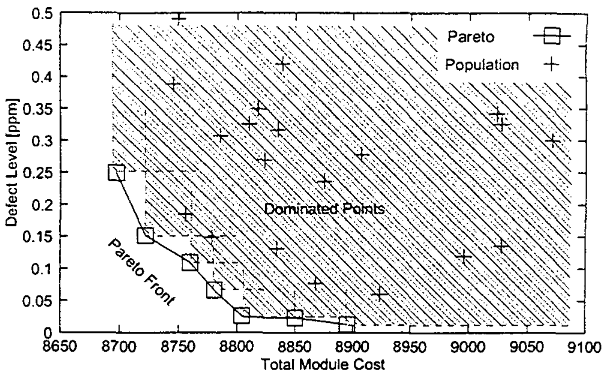

Figure 5: Example for Pareto Optimality

cause the total cost is only increasing. Since we want to monitor both quality and cost concurrently, we therefore need another result representation as this single figure of merit. Consequently, we use a two-dimensional graph plotting cost and test escape level simultaneously.

Fig. 3 presents the results for a given cost-test coverage relation, where scenarios A to E turn out to be so-called Pareto points. Solution A is the status-quo point with the actual IC wafer level test coverage and final test coverage. Solutions B to E represent the respective tradeoff for a dedicated defect level. Solution F can be dropped because E gives a better solution in terms of cost. Proceeding from B to E, a defect level reduction can be achieved while increasing the total module cost. The simplest approach (extending the final test time and therefore test coverage) already gives a remarkable defect reduction. Introducing IddQ for the lower-yield LPD chip even performs better at minimal additional cost. The use of IddQ for the LVL2 chip improves the defect level only minimal, so this action could be left away to spare engineering resources to redesign the chip.

Combining the strategies B and D to E gives again an improvement, reaching the minimum defect level border for the case. Now, setting a maximum acceptance value for cost and/or for test escapes, the final test setup can be chosen.

Pareto Optimality

What makes now solutions superior to others? Similar to Fig. 3 another two-dimensional plot is shown in Fig. 4. For ranking purposes of data points we refer to the domination principle: "For a minimization problem (minimum cost, minimum defect level) a point (indicated by the circle) dominates all the data lying in the hatched area right above it (these points represent higher cost AND higher defect level). It gets dominated by all the points left below it, in the shaded area (these points have lower cost AND lower defect level)". Points that do no get dominated by other ones are called "Pareto points". These points, indicated by the hatched triangles, are the ones located closest to the graph's axes [10].

When all these points are superposed, we obtain a hatched area (see Fig. 5) wherein all the points are dominated and on its edges a Pareto front giving the reachable borders of the optimization problem.

For the simple readout electronics case above with a countable number of choices, the Pareto search could be done semi-automatically, but one can easily imagine that for a higher number of alternatives a more structured search mechanism is required. This mechanism is described in the following section.

Automated Search for Pareto Points

In Fig. 6 the MOE manufacturing model for a GPS front end is depicted. It includes an RF die (upper left) and a correlator die (medium left). The RF ($yield1$) die undergoes a pre-screening with fault coverage $fc1$, the correlator ($yield2$) is rerouted for flip chip attach and afterwards optically inspected ($fc2$). In case of an error, this chip can be repaired once with the success rate $yield3$. Both dies are attached onto the thinfilm substrate (upper right corner with $yield4$), and then the entire system undergoes a functional test before shipping ($fc4$).

For every of the total 7 parameter, we arbitrarily define a range of 8 applicable values, leading to

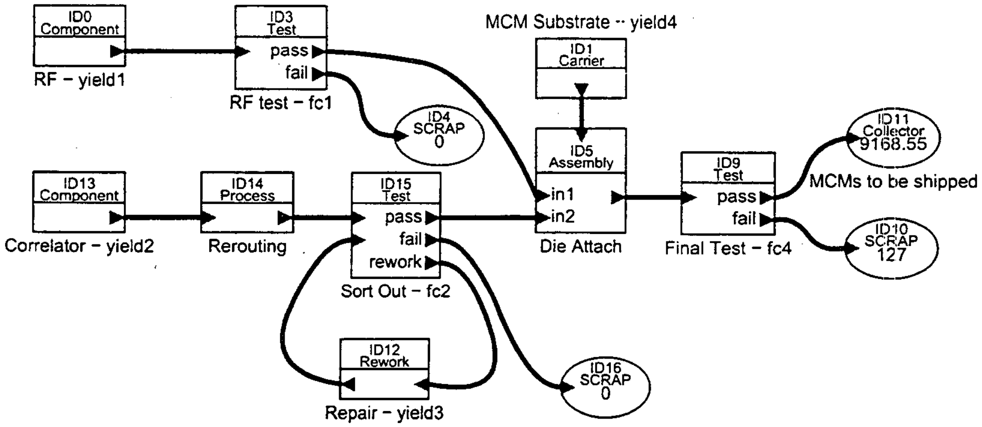

Figure 6: Graphical manufacturing model for a GPS RF front end

$8^7 = 2097152$ possible combinations. Additionally, when changing a yield or fault coverage parameter, dependent parameters (component cost or test cost) have to be changed as well. An overview of all parameters can be found in Table III. The number of possible alternatives forbids to make a brute force calculation of all combinations.

Due to the nature of the problem (discrete parameter space, no closed formula), no analytic solution is possible. Now, the usual procedure would be to use some standard heuristic methods, as e.g. simulated annealing, or hill climbing. But those procedures are suitable only for single objective optimization. Thus, we would have to use some weighting factors to combine multiple objectives to use these methods (see e.g. [7]). The problem with this approach is that

- bad performance with regard to some objectives can be compensated, which is often not desired;
- we have to define weighting factors in advance without knowing the search space;
- we only obtain a single solution without getting any additional information on the search space.

Therefore, a parallel search is considered to be the better way to avoid these obstacles [8]. A suitable method is the use of genetic/evolutionary algorithms (GA/EA)[2] The general scheme of this methodology is listed in pseudo-code below:

```
begin
initialize starting population
assess fitness of this population
repeat
   select parents
      according to their fitness
   replicate (using crossover
      and/or mutation)
   assess offspring
   generate new population
until max number of generations
   or stop criterion
print out results
end
```

Search for the Optimum

For the example above the 8 possible float numbers for each parameter have been coded as a 3-digit gene. All genes are lined up to a single bitstring ("individual") that, together with all other individuals, is fed into the evolutionary algorithm for optimization purposes. The resulting bitstrings ('the offspring") are decoded again to real values and fed into the MOE simulation engine. The cost/quality outcome is in turn fed into the EA algorithm to assess the fitness of the offspring. This loop continues for a given number of generations. The selection algorithm has been implemented adopting the Strength Pareto approach [10].

---

[2] For reasons of simplifications, we will not distinguish between those two concepts. For some considerations see [9].

0-7803-6482-1/2000/$10.00 ©2000 IEEE

Table III: Values for the parameters used in Fig. 6

	low				medium			high
				main variables				
yield1	0.5	0.557143	0.614286	0.671429	0.728572	0.785715	0.842858	0.900001
yield2	0.5	0.568572	0.637143	0.705715	0.774286	0.842858	0.91143	0.980001
yield3	0.687023	0.730644	0.774264	0.817885	0.861505	0.905126	0.948747	0.992367
yield4	0.687023	0.730644	0.774264	0.817885	0.861505	0.905126	0.948747	0.992367
fc1	0.5	0.557143	0.614286	0.671429	0.728572	0.785715	0.842858	0.900001
fc2	0.5	0.568572	0.637143	0.705715	0.774286	0.842858	0.91143	0.980001
fc4	0.687023	0.730644	0.774264	0.817885	0.861505	0.905126	0.948747	0.992367
				dependent variables				
yield1_cost	50	55.7143	61.4286	67.1429	72.8572	78.5715	84.2858	90.0001
yield2_cost	25	28.4286	31.8572	35.2857	38.7143	42.1429	45.5715	49.0001
yield3_cost	137.405	146.129	154.853	163.577	172.301	181.025	189.749	198.473
yield4_cost	343.511	365.322	387.132	408.942	430.753	452.563	474.373	496.184
fc1_cost	50	55.7143	61.4286	67.1429	72.8572	78.5715	84.2858	90.0001
fc2_cost	100	113.714	127.429	141.143	154.857	168.572	182.286	196
fc4_cost	687.023	730.644	774.264	817.885	861.505	905.126	948.747	992.367

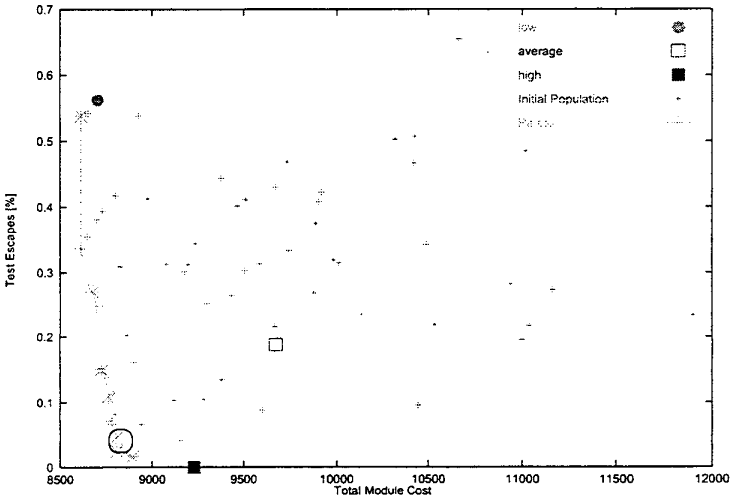

Figure 7: Comparison of an EA run with high-medium-low parameter combinations; the Pareto front clearly outperforms the average parameter combination

Fig. 7 shows the result for a run with the EA framework (50 Generations/50 Individuals). In addition, we calculated three typical parameter combinations:

low: minimum values from Table III,

medium: a medium value from every parameter range,

high: maximum values from Table III.

As one can clearly see from Fig. 7, the outline square (medium values in Table III) is surpassed by the filled square (high values) and the filled circle (low values) parameters either in terms of cost or in terms of quality. The Pareto front (stars) gives the connection between the high/low values, clearly outperforming the average solution and even the low-value solution. Thus, starting in the upper left corner of the Pareto front one can see that with small cost investments significant gains in quality can be made. When the test escapes have come down to approx. 0.03%, a turning point (indicated by the circle) is reached and further im-

0-7803-6482-1/2000/$10.00 ©2000 IEEE

Table IV: Decoded Parameters for the Pareto Front

yield1	yield2	yield3	yield4	fc1	fc2	fc4	cost	defect level [%]
0.9	0.98	0.77	0.77	0.9	0.5	0.69	8697.04	0.2491
0.9	0.98	0.69	0.91	0.5	0.5	0.77	8721.87	0.1510
0.9	0.98	0.77	0.91	0.5	0.5	0.69	8724.66	0.1499
0.9	0.98	0.69	0.91	0.61	0.5	0.69	8740.03	0.1441
0.9	0.98	0.69	0.91	0.9	0.5	0.77	8759.8	0.1090
0.9	0.98	0.91	0.91	0.9	0.5	0.69	8762.84	0.1058
0.9	0.98	0.77	0.99	0.5	0.5	0.77	8765.81	0.0700
0.9	0.98	0.91	0.99	0.61	0.5	0.77	8780.42	0.0660
0.9	0.98	0.77	0.99	0.79	0.5	0.69	8801.19	0.0450
0.9	0.98	0.99	0.99	0.9	0.5	0.77	8804.06	0.0260
0.9	0.98	0.69	0.99	0.9	0.5	0.69	8806.17	0.0242
0.9	0.98	0.69	0.99	0.9	0.64	0.99	8850.05	0.0230
0.9	0.98	0.69	0.99	0.9	0.84	0.77	8894.06	0.0168
0.9	0.98	0.77	0.99	0.9	0.98	0.77	8895.04	0.0150
0.9	0.98	0.69	0.99	0.9	0.98	0.77	8895.38	0.0124

provement in quality only comes at a considerable cost. This turning point would be the best state to chose for fabrication set-up.

When looking at the decoded parameter results for the Pareto front (Table IV), one can see that from one Pareto point to another, about only 30% of the parameters change their values. This gives raise to the hope that the Pareto points are fairly stable, but this matter requires additional investigations. Moreover, for the $yield1$- and $yield2$- parameters, always maximum yield are used due to its low cost penalty. Thus, for the given set up it is crucial to maintain high yield for all Pareto points.

Conclusions

In this paper we presented a quality vs. cost tradeoff for electronic systems. In contrast to existing approaches, which would have provided only a cost perspective of the results, our method offers a second important view on the result space. Only from the cost point of view, none of the manufacturing changes (IddQ, extended test) in the readout application example would have made sense, as no significant yield loss reduction could have been achieved. But when the module cost is not the only driving force, we see that there is a benefit in terms of quality when changing the testing setup.

Using the automated search, the designer can also make a risk assessment testing the case when manufacturing key specs are not met (e.g. die supplier does not meet yield, test engineer cannot deliver fault coverage promised). Visualizing the population helps extrapolating the ranges within cost and quality can float for a given parameter space. Subsequent analysis can than focus on the region of interest.

Finally, the resulting Pareto front gives the opportunity to chose the optimal process setup to meet given cost and quality specifications.

Future work will include enhanced stability of the Pareto points.

References

[1] P. Sandborn and H. Moreno, *Conceptual Design of Multi-Chip Modules and Systems*. Kluwer Academic Publishers, 1 ed., 1994.

[2] V. Agrawal, S. Seth, and P. Agrawal, "Fault Coverage Requirements in Production Testing of LSI Circuits," *IEEE J. Solid-State Circuits*, vol. SC-17, February 1982.

[3] C. Dislis and J. Jalowiecki, "Economics Modeling and Optimization of MCM Test Strategies," in *Proc. European Design and Test Conference 1996* (A, ed.), p. 615, IEEE Computer Soc. Press, Los Alamitos, CA, US, 1996.

[4] J. Turino, "Test Economics in the 21st Century," *IEEE Design and Test of Computers*, p. 41, July - September 1997.

[5] M. Scheffler, D. Ammann, A. Thiel, C. Habiger, and G. Tröster, "Modeling and Optimizing the Cost of Electronic Systems," *IEEE Design + Test of Computers*, vol. 15, no. 3, pp. 20–26, 1998.

[6] M. Scheffler, G. Tröster, R. Mariani, S. Giovanetti, S. Motto, A. Benso, and P. Prinetto, "Test

vs. Cost Tradeoff for a Data Acquisition Chain," in *IEEE/IMAPS 2nd International Workshop on Chip-Package Codesign (CPD2000), Zurich, CH, March 14 and 15*, pp. 41–45, 2000.

[7] Y. Tokat, O. Palusinski, and F. Szidarovszky, "Multicriteria Decision Making in Design of Oriented Wire Boards," *IEEE Trans CPMT B*, vol. 21, no. 1, pp. 73–78, 1998.

[8] V. Nissen, *Einführung in evolutionäre Algorithmen* Computational Intelligence, vieweg Braunschweig Wiesbaden, 1997.

[9] D. Fogel, *Evolutionary Computation: towards a new philosophy of machine*. IEEE Press, Piscataway NJ, 1995.

[10] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Trans Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

0-7803-6482-1/2000/$10.00 ©2000 IEEE