Circuit Design Using Evolutionary Algorithms

Thomas Beielstein[1], **Jan Dienstuhl**[2] **Christian Feist**[1], **and Marc Pompl**[1]

[1]Department of Computer Science XI, University of Dortmund, D-44221 Dortmund, Germany.
{tom,feist,pompl}@LS11.cs.uni-dortmund.de
[2]Microelectronics Department, University of Dortmund, D-44221 Dortmund, Germany.
Jan.Dienstuhl@udo.edu

Abstract- In this paper we demonstrate the applicability of evolutionary algorithms (EAs) to the optimization of circuit designs. We examine the design of a full-adder cell, and show the capability of design of experiments (DOE) methods to improve the parameter-settings of EAs.

1 INTRODUCTION

The growing demand for high performance mobile electronic systems with enhanced battery lifetime requires the design of fast integrated circuits with very low power consumption. Switching speed and power optimization of digital CMOS circuits for nowadays submicron design-flows must already be applied at the lowest abstraction levels of the design hierarchy. Considerable power savings and switching speed raisings can be achieved by addressing circuit level design optimization. In the case of standard cell library design, basic switching elements can be implemented in various circuit techniques and transistor geometries. The estimation of the capabilities of each design choice, regarding different objective functions, e.g. power dissipation, signal delay, chip area, etc., represents a global, high-dimensional optimization problem, that can not be sufficiently solved by applying conventional methods such as steepest-gradient algorithms.

The evolutionary methodology provides such a global high-dimensional optimum seeking algorithm, without the necessity of starting the optimization with an already near-optimal solution in the parameter space. Thus working designs for innovative circuit concepts can be found, even when an a-priori determination of design parameters by a human expert is not simple. Furthermore the relations between various target-parameters for a given circuit topology can be extracted, so that different design concepts may be compared objectively. By observing the resulting Pareto-sets the preferable design for a specific application and given parameter constraints can be determined.

Although the self-adaptation of the strategy parameters is an inherent feature of EAs, especially of evolution strategies (ES), the optimization practitioner might be interested in information about a 'good' initial parameter setting to speed up the search process. For many practitioners the following question might be of great importance: *What is the best EA for my specific optimization problem?*
Design of experiments methods provide good means to extract the important parameter-settings (screening), e.g. the required population size or selective strength (parent-offspring ratio) [LK00, KVG92, BM01]. These methods are already known for many decades in statistics. In this paper, we will give some hints for the practitioner, how DOE methods can be used to set up simulation runs for the optimization of circuit designs. These methodologies can be easily transferred to other real-world optimization problems, that are based on simulation models. The applicability of our methods is not restricted to EAs, but can be applied to any optimization technique that requires certain parameter settings.

Section 2 compares commonly used methods for the optimization of circuit designs to EAs. A full-adder design is discussed in detail, and relevant (single- and multiobjective) fitness functions are presented. Section 3 gives a short introduction into our evolutionary approach. In Sec. 4 we introduce the investigated simulation model. Furthermore important aspects regarding the implementation of EAs for real-world optimization problems are discussed from the viewpoint of an optimization practitioner. The results of the simulation runs are presented and interpreted in Sec. 5. The last section gives a summary and an outlook.

2 CIRCUIT DESIGN

2.1 Classical approaches

Transistor-level design and simulation of different circuit structures result in physical layout realizations of individual gates and registers. For the nominal optimization of standard cells classical linear or non-linear programming methods are used. Well known design tools, such as *Delight.Spice* by [NRSVT88], *JiffyTune* by [CCH$^+$98], and *WiCkeD* by [AEG$^+$00], are found on the non-linear steepest-gradient algorithm. This method is based on the idea of tracking down the steepest descent in an optimization landscape. The gradient algorithm is only suitable for finding a minimum, when a near optimal solution is already known, because it is limited to local search. This classical approach can only optimize individual circuit implementations. The result of the optimization process is one scalar value, for example a simulated signal delay, dissipated power, or estimated chip area. Therefore it is not possible to extract global parameter dependencies, such as the power-delay relation. For the optimization of multiple objectives artificial criterions have to be defined, e.g. the power-delay or energy-delay product.

To enable an objectively global comparison of different design choices the relations between various target-parameters have to be determined. The evolutionary algorithms, that use a population-based approach, can cope with this task.

2.2 Full-Adder Design

To demonstrate the basic capabilities of the presented methodology, the design of a full-adder (FA) cell will be examined. The boolean expressions for the binary adder outputs are $c_o = a_i b_i + b_i c_i + a_i c_i$ and $s_o = a_i b_i c_i + \overline{c_o}(a_i + b_i + c_i)$. The bits a_i and b_i are the adder inputs, c_i is the carry input, s_o is the sum output, and c_o is the carry output. FAs are the basic cells in adder arrays, e.g. carry-save adders, used in multipliers and similar components like dividers. In such applications, efficient full-adder circuits are crucial since this building blocks are often critical and determine the operational speed, power dissipation and chip area of the system [ZF97]. The design methodology will be applied to static CMOS mirror adder depicted in Figure 1. The cells were designed at the transistor level in a standard $0.35\,\mu m$ CMOS process technology ($V_{tn} = 0.47\,V, V_{tp} = -0.62\,V$) and were simulated using HSPICE for a supply voltage range from $1.5\,V$ to $3.3\,V$ at $27\,°C$. Worst case gate delays and average power dissipation, including short circuit currents, are obtained from simulation. The circuit designs from which the optimization process is started are sized by hand with the objective of balanced gate performance. Certain symmetry properties of the adder designs may be derived by taking the semantic equivalence of the input signals a_i and b_i into account. The according transistors are sized equally, so that the number of parameters which have to be optimized is reduced.

2.3 Objective Functions

Consider the following notation:

$a =$	chip area,
$pow =$	dissipated power,
$pun =$	punishment,
$tdf_{co} =$	fall-time carry,
$tdf_{so} =$	fall-time sum,
$tdr_{co} =$	rise-time carry,
$tdr_{so} =$	rise-time sum.

Based on this notation, we define eight different objective functions, that are relevant for practitioners:

$F1$:	$power + pun$,	$F5$:	$\max\{tdc, tds\}$,
$F2$:	tdc,	$F6$:	$(pow, tdc)^T$,
$F3$:	$a + pun$,	$F7$:	$(pow + pun, a)^T$, and
$F4$:	$(pow + pun)/a$,	$F8$:	$(pow, tdc, a)^T$,

where tdc and tds are defined as follows:

$$tdc = 0.5 \cdot (tdf_{co} + tdr_{co}),$$
$$tds = 0.5 \cdot (tdf_{so} + tdr_{so}).$$

3 EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION TECHNIQUES

Multi-objective optimization approaches are classified with regard to the fitness assignment and selection technique: One can distinguish criterion selection, aggregation selection, and Pareto selection [ZDT00].

We are interested in a comparison of the different approaches. Since the aggregating method is easy to implement, we chose this 'classical' approach first. The aim of our first analysis is to find optimal solutions of the fitness function $F1$. The function $F1$ can be interpreted as a degenerated weight function. Furthermore, aggregating methods might generate a strongly non-dominated solution, that can be used as an initial solution for advanced methods [CC99].

Our approach will be extended, thus we are able to handle a broad variety of multi-objective evolutionary algorithms (MOEAs) [TBG00, Deb01, Hor97, LRS99, RA00, Zit99].

4 EXPERIMENTAL DESIGN

4.1 EA Design Considerations

In the following, we can mention only few considerations, that typically arise, when an optimization problem is analyzed and a suitable EA has to be chosen.

Our first goal was to investigate the overall behavior of evolutionary algorithms on a complex problem like circuit design. Since one simulator run is costly with respect to the required CPU-time and the required amount of memory, we chose the following approach: We used pre-computed data to speed-up the fitness function evaluations, obtained by a discrete search area sweep. This results in a large sweep file (≈ 122 M, with $823,543$ entries), that contains the calculated fitness function values. Thus we were able to determine the global optimum and other characteristics of the fitness function. This approach basically has two disadvantages: First, it takes time to generate the sweep file, and, perhaps the most important disadvantage, the sweep-file contains only values generated with discrete steps in the domain of the fitness function. Therefore, we will use a direct connection to the simulator in the next stage of of experiments, that provides a much higher resolution. This analysis will be performed with the promising strategies determined during the first step of our investigations.

Before we can implement the EA, we have to take into consideration, whether the parameters in the given circuit design problem can be independently modified or not. Additionally we have to exclude forbidden combinations, that require repair-mechanisms. Furthermore it may be efficient to limit the step sizes to the technical constraints. Especially very small values should be handled, otherwise mutated individuals will be mapped to the same phenotype ever and anon, because their distances are smaller than a discrete fitness-function can resolve. This arises as a consequence of the discretization mentioned above.

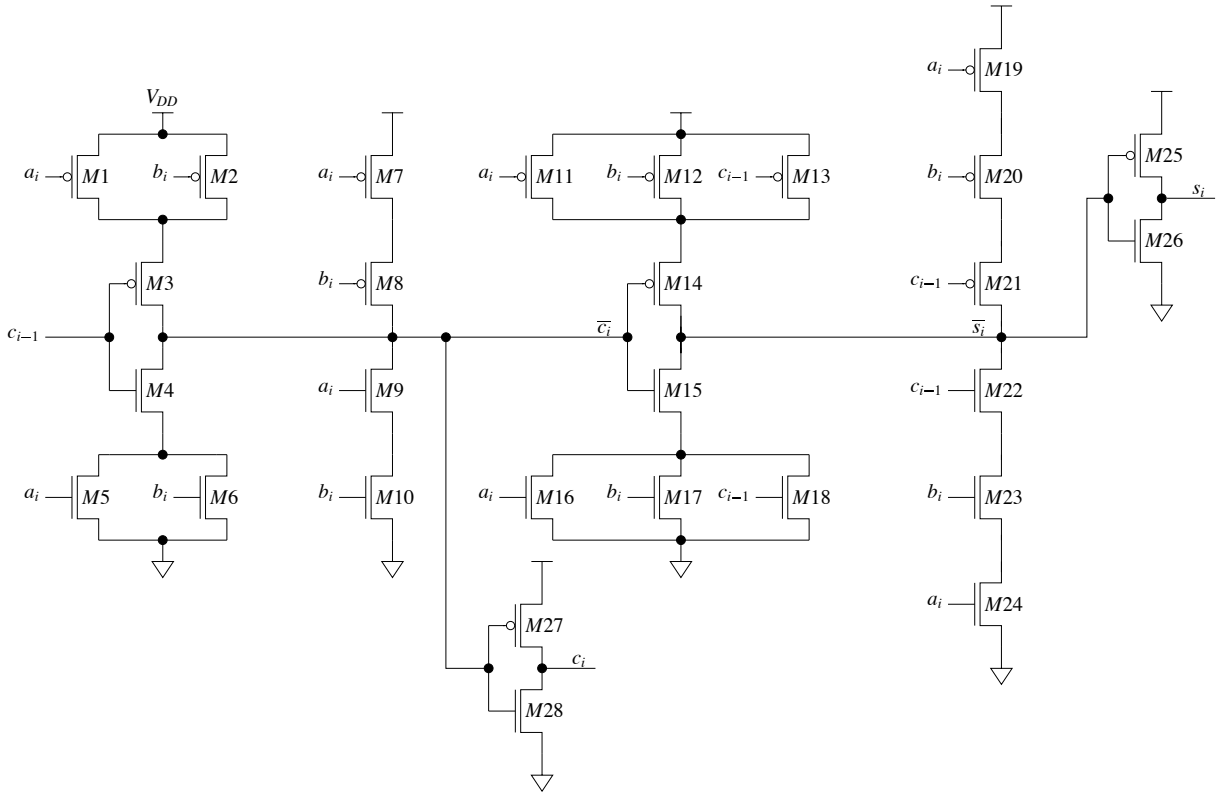

Figure 1: Static CMOS mirror adder.

Factor:	**Level:**
(A) Selection	comma-strategy $(-)$, plus-strategy $(+)$
(B) Selective Pressure	$2.0, 7.0,$ and 10.0
(C) Population Size	$2, 3,$ and 5.

Table 1: FACTORS A, B AND C. FACTORIAL DESIGN USED THROUGHOUT THIS PAPER.

Strategy:	$2,4$	$2,14$	$2,20$	$3,6$	$3,21$	$3,30$	$5,10$	$5,35$	$5,50$
Rejects:	44	0	0	12	0	0	6	0	0
Optimum hits:	0	8	4	0	10	6	0	14	14
Strategy:	$2+4$	$2+14$	$2+20$	$3+6$	$3+21$	$3+30$	$5+10$	$5+35$	$5+50$
Rejects:	2	0	0	0	0	0	0	0	0
Optimum hits:	10	12	10	8	18	18	12	16	14

Table 2: REJECTS AND OPTIMUM HITS IN %.

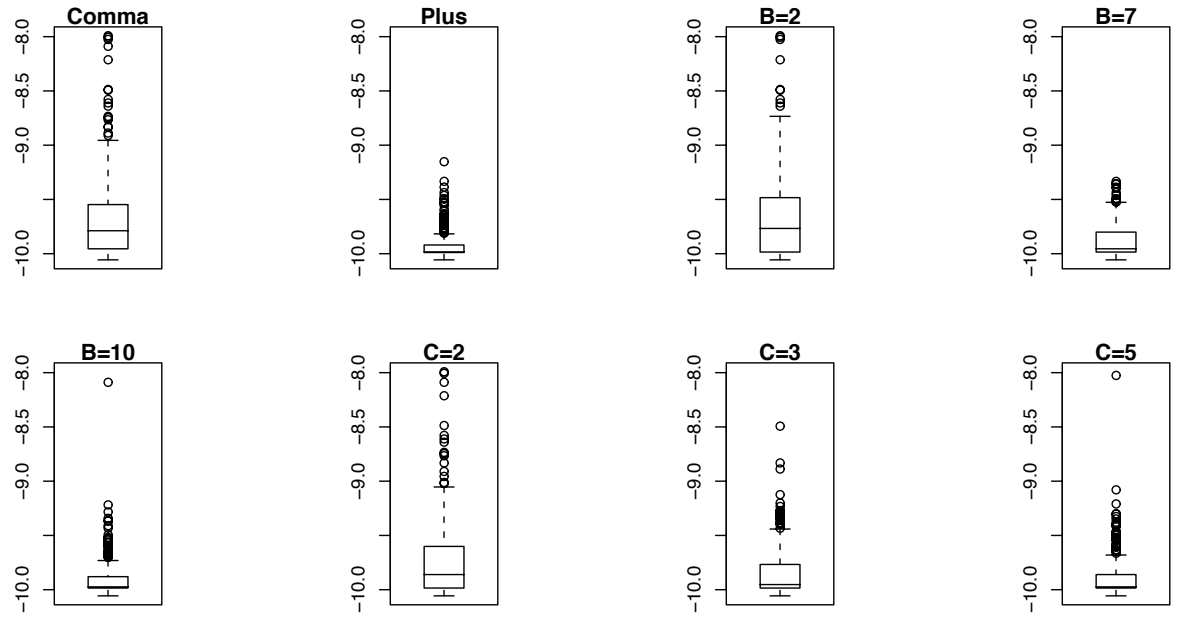

Figure 2: Boxplots. Visualizing the fitness function values, if one parameter is kept constant.

4.2 EA Parameter Settings

To optimize the fitness function $F1$, we chose the following implementation:

Individual: An individual represents transistor sizes and voltages in a circuit. Every given parameter is mapped to a field in the object-variables array. The strategy-parameters consist of globally adapted step-size values for each parameter (isotropic self-adaptation [Sch95]).

Recombination: A discrete recombination is implemented. Every parameter value of a recombined individual is obtained from the corresponding values of a pair of randomly selected parents.

Mutation: All variables are mutated by isotropic self-adaption. Because the parameters differ in dimension, the values are mapped to a normalized genotypic representation. Thus recombination and mutation can be used efficiently.

Selection scheme and fitness evaluation: The implementation allows selection of comma- and plus-strategies.

Furthermore, $1,000$ fitness function evaluations were performed in every simulation run, and every factor-level combination was repeated 50 times. The algorithm terminates after the given number of fitness function evaluations were performed. Due to the genotype-phenotype-mapping, no explicit repair function was required.

Now we consider the parameters of our strategy, that shall be optimized. We used a factorial design, with the factorial settings shown in Tab. 1. The selective pressure is defined as the offspring-parent ratio λ/μ. This design leads to 9 different combinations for each selection scheme, cp. Tab. 2. Thus we have $50 \cdot 18 = 900$ experiments altogether.

Since the simulations were run on different operating systems (Unix/Linux, resp. MS-Windows), we implemented the EA in JAVA using Sun's J2SE 1.3.

5 RESULTS

5.1 Hits and Rejects

In order to obtain a first impression of the quality of the different strategies, we can calculate the percentage of *rejects* and the number of optimum *hits* (without loss of generality, we will only consider minimization problems):
A simulation run is called a *reject*, if the value of the best individual in the last generation is larger than a pre-defined constant c. If a simulation run was able to find the global optimum (that is known from the pre-evaluated fitness function values), it is called a *hit*. The results in Tab. 2 are based on the following values: $c = 10^{-4}$, whereas the best pre-computed fitness function value equals $4.29 \cdot 10^{-5}$.
Tab. 2 can give a first impression of the behavior of the algorithm. To gain further insight into the relevance of and interactions between the parameters, we use well-known statistical methods, such as analysis of variance and regression analysis [LK00, Kle87, KVG92, Kle99, Bei01, BD87, BHH78]. The statistical results and plots presented in this paper were generated with the software package R [IG96].

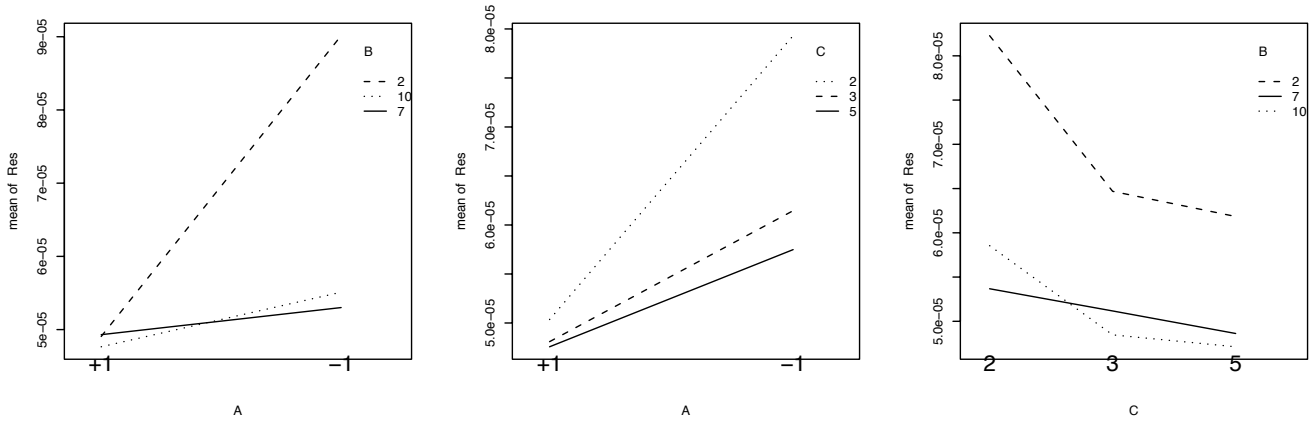

Figure 3: Interaction plot. Visualizing interactions between different factor level settings as defined in Tab. 1.

5.2 Descriptive Statistics

To apply statistical tests to our model, it has to be validated beforehand. Common techniques for the validation of simulation models are discussed in [Kle99, LK00] and will be omitted here.

DOE methods enable us to apply well-known visualization techniques from descriptive statistics to our data. Thus we may obtain more information from the data than the 'classical' *fitness plotted against the number of generations* plots can provide. We present two different visualization methods:

1. Boxplots, shown in Fig. 2: For each boxplot, one parameter was kept constant. The first plot reveals the behavior of the comma-strategies, based on their logarithmized fitness function values. On average, plus-strategies perform better than comma-strategies, as can be seen from a comparison of the first and the second plot.

2. Interaction plots, shown in Fig. 3: The labels on the x-axis are corresponding with the following parameter settings: Selection scheme (comma-strategy $= -1$, plus-strategy $= +1$) in the first two figures (viewed from the left to the right), and population size ($2, 3,$ and 5), in the last figure. Factorial design gives information about how different the factors interact with each other.

5.3 Interpretation of the Results

Boxplots and interaction plots provide an excellent way to visualize the influence of different parameter-settings on the behavior of the EA. Obviously the plus-strategies perform better than the comma-strategies – independently from all other parameter settings. Increasing the population size from $\mu = 2$ over $\mu = 3$ to $\mu = 5$ shall also lead to a better performance of the algorithm. The optimal settings of the selective pressure depends on the settings of the other parameters, as can be seen from Fig. 3.

6 SUMMARY AND OUTLOOK

A promising approach for circuit analysis and design using EAs, that extends the methods presented in [TBG00], was shown. Based on pre-determined fitness function values, the required time for a simulation run could be drastically reduced. Hints for an improved setting of strategy parameters could be derived. Further experiments shall be performed, i. e. to reveal an upper limit for the optimal population size. DOE methodologies simplify and enhance the analysis of the obtained data. The transferability of the data to the 'real' simulator should be possible, since the internal calculations of the simulator are also based on discrete steps. Thus in the next step of our experimental analysis we will use fitness function values, that were directly generated by the simulator. Multi-objective fitness functions, as defined in Sec. 2, will also be included in our investigations.

Acknowledgments

This work is a product of the Collaborative Research Center 'Computational Intelligence' (531), at the University of Dortmund and was supported by the Deutsche Forschungsgemeinschaft (DFG).

Bibliography

[AEG+00] K. Antreich, J. Eckmüller, H. Gräb, M. Pronath, F. Schenkel, R. Schwencker, and S. Zizala. Wicked: Analog circuit synthesis incorporating mismatch. *IEEE Custom Integrated Circuits Conference (CICC)*, Mai 2000.

[BD87] G. E. P. Box and N. R. Draper. *Experimental Model Building and Response Surfaces*. Wiley, 1987.

[Bei01] T. Beielstein. Design of experiments and evolutionary algorithms. Part I: The classical model. Technical report, Universität Dortmund, Fachbereich Informatik, 2001. (to appear).

[BHH78] G. E. P. Box, W. Hunter, and J. S. Hunter. *Statistics for experimenters*. Wiley series in probabilty and mathematical statistics: Applied probability and statistics. Wiley, 1978.

[BM01] T. Beielstein and S. Markon. Threshold selection, hypothesis tests, and DOE methods. Technical Report of the Collaborative Research Center 531 *Computational Intelligence* CI–121/01, University of Dortmund, December 2001.

[CC99] C. Coello and A. Carlos. A survey of constraint handling techniques used with evolutionary algorithms. Technical Report Lania-RI-99-04, Laboratorio Nacional de Informática Avanzada, 1999.

[CCH+98] A.R. Conn, P.K. Coulmann, R.A. Haring, G.L. Morill, C. Visweswariah, and C.W. Wu. Jiffy tune: Circuit optimization using time-domain sensitivities. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1292–1309, 1998.

[Deb01] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley, New York, 2001.

[Hor97] J. Horn. Multicriterion decision making. In Z. Michalewicz T. Bäck, D.B. Fogel, editor, *Handbook of Evolutionary computation*, pages pp F1.9:1–15. IOP Publishing and Oxford University Press, New York and Bristol, 1997.

[IG96] R. Ihaka and R. Gentleman. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314, 1996.

[Kle87] J. Kleijnen. *Statistical Tools for Simulation Practitioners*. Marcel Dekker, New York, 1987.

[Kle99] J. Kleijnen. Validation of models: Statistical techniques and data availability. In P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans, editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 647–654, 1999.

[KVG92] J. Kleijnen and W. Van Groenendaal. *Simulation - A Statistical Perspective*. Wiley, Chichester, 1992.

[LK00] A. M. Law and W. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Series in Industrial Egineering and Management Science. McGraw-Hill, New York, 3 edition, 2000.

[LRS99] M. Laumanns, G. Rudolph, and H.-P. Schwefel. Approximating the Pareto Set: Concepts, Diversity Issues, and Performance Assessment. Technical Report CI-72/99, Dortmund: Department of Computer Science/LS11, University of Dortmund, Germany, March 1999. ISSN 1433-3325.

[NRSVT88] W. Nye, D.C. Riley, A. Sangiovanni-Vincentelli, and A.L. Tits. Delight.spice: An optimization-based system for the design of integrated circuits. *IEEE Transactions on Computer-Aided Design*, 7(4):501–519, 1988.

[RA00] G. Rudolph and A. Agapie. Convergence Properties of Some Multi-Objective Evolutionary Algorithms. In *Proceedings of the 2000 Conference on Evolutionary Computation*, volume 2, pages 1010–1016, Piscataway, New Jersey, July 2000. IEEE Press.

[Sch95] H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology. Wiley Interscience, New York, 1995.

[TBG00] M. Thomas, C. Burwick, and K. Goser. Circuit Analysis and Design using Evolutionary Algorithms. Technical Report CI–85/00, SFB 531, Universität Dortmund, 2000.

[ZDT00] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, April 2000.

[ZF97] R. Zimmermann and W. Fichtner. Low Power Logic Styles: CMOS Versus Pass-Transistor Logic. *IEEE Journal of Solid-State Circuits*, 32(7):1079–1090, July 1997.

[Zit99] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Application*. Berichte aus der Informatik. Shaker Verlag, Aachen - Maastricht, 1999.