Chapter 2
State-of-the-Art on Automatic Analog IC Sizing

Abstract In this chapter a state-of-the-art review on analog integrated circuit (IC) design automation tools applied to the specification translation problem is presented. Having the right topology for a given set of specifications is indispensable for a high performance design. An inadequate topology makes the design more difficult (or even impossible), and may require unnecessary resources, which is not acceptable in high performance designs. Once the topology is selected, the specifications for the overall block are translated to the specifications for the sub-blocks. The specifications are, in this way, passed through the hierarchy. At the lowest level, the translation reduces to circuit sizing, whereas at the higher levels it produce the sub-blocks performance parameters. In the last years, the scientific community proposed many techniques for the automation of the translation task; some apply only at circuit-level or only at system level, while others apply to both. In this study, several circuit-level sizing techniques are sketched and compared, and then, different model-based optimization approaches are outlined.

Keywords Analog IC design · Automatic specification translation · Knowledge-based sizing · Optimization-based sizing · Electronic design automation · Computer-aided-design

2.1 Automatic Circuit-Level Sizing

The techniques for the automation of circuit-level IC sizing are classified into two main groups [1], knowledge-based and optimization-based based on the techniques used to address the problem.

F. A. E. Rocha et al., *Electronic Design Automation of Analog ICs Combining Gradient Models with Multi-Objective Evolutionary Algorithms*, SpringerBriefs in Computational Intelligence,
DOI: 10.1007/978-3-319-02189-8_2,

2.1.1 Knowledge-Based Sizing

Early strategies tried to systematize the design by using a design plan derived from expert knowledge. In these methods, a pre-designed plan is built with design equations and a design strategy that produce the component sizes that meet the performances requirements. Figure 2.1 shows the strategy flow of knowledge-based sizing methodologies.

In IDAC [2], the designer expertise is captured in a design plan where all design equations are explicitly solved during the execution of the plan. Once the topology is selected, the plan is executed for the given specifications to produce a first design. The tool also included local optimization around this first design. IDAC includes a vast library of plans, featuring voltage references, opAmps, comparators, oscillators, DACs and ADCs. OASYS [3] uses the same overall strategy, but defines the circuits hierarchically, with a design plan for each sub-block. It also adds backtracking with design-reuse methodologies to recover from failed designs. OASYS was extended to include data converters in addition to the original operational amplifiers. TAGUS [4–6] applies the design plan successfully at system-level for CMOS data converters. A slightly different approach is found in BLADES [7], CAMP [8] or ISAID [9, 10], these tools capture the designer's knowledge in expert systems using artificial intelligence techniques.

The knowledge-based approach was applied with moderate success. The main advantage of this approach is the short execution time. On the other hand, deriving the design plan is hard and time-consuming, the design plan requires constant maintenance in order to keep it up to date with technological evolution, and the results are not optimal, suitable only as a first-cut-design.

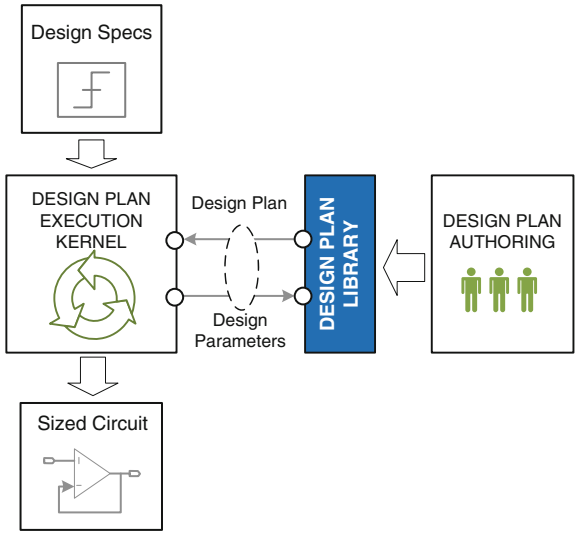

Fig. 2.1 Automatic circuit sizing: knowledge-based methodology

2.1.2 Optimization-Based Sizing

Aiming for optimality, the next generations of sizing tools apply optimization techniques to analog IC sizing. The optimization-based sizing can be classified into three major subclasses based on different techniques, namely, equation-based, simulation-based and model-based, which are addressed in the following subsections. A general flow of an optimization-based strategy can be found in Fig. 2.2.

2.1.2.1 Equation-Based

The equation-based methods use analytic design equations to evaluate the circuit performance. Different optimization techniques are used, the optimization in OPASYN [11] is done using steepest descent, whereas in STAIC [12] it is used a successive solution refinements technique. OPTIMAN [13] uses simulated annealing (SA) applied to analytical models created automatically by ISAAC [14]. DONALD [15] is an interactive design space exploration tool that assists the designer during circuit sizing by automatic analytical manipulations of the circuit equations. Maulik et al. [16] define the sizing problem as a constrained nonlinear optimization problem using spice models and DC operating point constraints, solving it using sequential quadratic programming. In ASTRX/OBLX [17] a simulated annealing optimization is performed using and cost function defined by equations for dc operation point, and small signal Asymptotic Waveform Evaluation based simulation. This evaluation technique is also used in DARWIN [18].

In GPCAD [19] a posynomial circuit model is optimized using Geometrical Programming (GP), the execution time is in the order of few seconds, but the

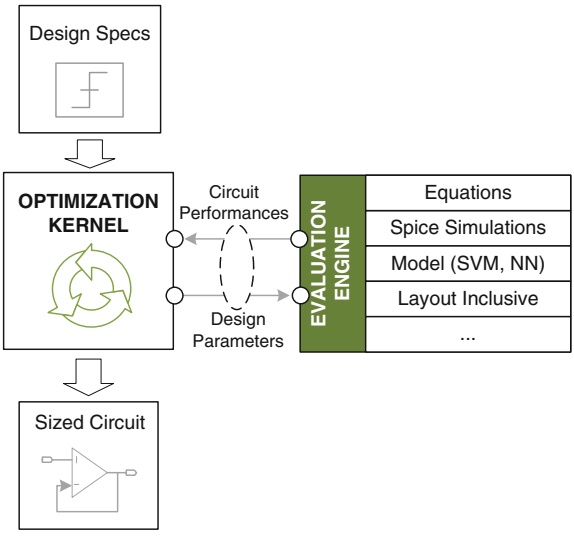

Fig. 2.2 Automatic circuit sizing: optimization-based methodology

general application of posynomial models is difficult and the time to derive the model for new circuits is still high. To reduce the long time spent in model development, automatic techniques were proposed (Gielen et al. in [20] provide a good overview on symbolic analysis applied to analog ICs). However, some design characteristics are still not easy to describe in analytical expressions with sufficient accuracy automatically. Kuo-Hsuan et al. [21] revisited the posynomial modeling recently, surpassing the accuracy issue by introducing an additional generation step, where local optimization using simulated annealing and a circuit simulator is performed. The same strategy is applied in FASY [22, 23] were analytical expressions are solved to generate an initial solution and a simulation-based optimization is performed to fine tune the solution.

The equation-based approaches are applied mostly at circuit-level, but some applications at system-level are also found. In SD-OPT [24] the optimal $\Delta\Sigma$ modulator sub-blocks' specifications are derived using symbolic equations solved using stochastic optimization. The sub-blocks itself are then generated using simulation based techniques. Doboli et al. [25] applies genetic programming techniques to simultaneously derive the sub-blocks specifications, sub-block topology selection and transistor sizing. Matsukawa et al. [26] design $\Delta\Sigma$ and pipeline analog to digital converters solving via convex optimization the equations that relate the performance of the converter to the size of the components.

The equation-based methods' strong point is the short evaluation time, making them, like the knowledge-based approaches, extremely suited to derive first-cut designs. The main drawback is that, despite the advances in symbolic analysis, not all design characteristics can be easily captured by analytic equations, in addition, the approximations introduced in the equations yield low accuracy designs especially for complex circuits.

2.1.2.2 Simulation-Based

With the availability of computing resources simulation based optimization gained ground. In simulation-based sizing a circuit simulator, like SPICE [27], is used to evaluate the circuit. In DELIGTH.SPICE [28] the optimization algorithm (phase I-II-III method of feasible directions) is used to perform local design optimization around a user provided starting point. Kuo-Hsuan et al. [21] and FASY [22, 23] use equation-based techniques to derive an approximate solution, and then use simulation within a simulated annealing optimization kernel to optimize the design. Cheng et al. [29] use the transistor bias conditions to constrain the problem and instead of solving the circuit by finding transistor sizes, the problem is solved by finding the bias of the transistors. The transistor sizes are derived from the bias point using electric simulation.

FRIDGE [30] on the other hand aims for global optimality by using an annealing-like optimization without any restriction to the starting point. However, to restrict the dimensionality of the problem the user still must provide the range for the optimization variables. In MAELSTROM [31] and ANACONDA [32] the

evaluation time is reduced by a parallel mechanism that shares the evaluation load among multiple computers. Given the affinity evolutionary algorithms have with parallel implementations, it was the base technique chosen in MAELSTROM, however and because the success of simulated annealing is demonstrated in many implementations the authors option was to use parallel re-combinative simulated annealing (PRSA). In ANACONDA the approach is similar but instead of the PRSA it is applied a variation of pattern search algorithms, named by the authors as stochastic pattern search.

In order to account for layout induced effects and layout characteristics Castro-Lopez et al. [33] include the layout effects and parameters in the optimization. A template based layout generator is integrated in the optimization loop and the geometrical properties of the layout can be used as constraints or optimized. In addition layout parasitic are also extracted and used during the circuit's evaluation. They use simulated annealing followed by a deterministic method for fine-tuning to perform the optimization. The layout extraction is done using analytical equations and layout sampling or using 3-D geometric extraction models.

A different approach is taken in GENOM-POF [34], where a multi-objective strategy is applied through the use of evolutionary algorithms. The objectives and constraint functions are evaluated by HSPICE®. GENOM-POF outputs the Pareto optimal fronts (POF) with the tradeoff during the synthesis, so the designer has a wider range of solutions and choices to the problem of sizing.

Generality and easy-and-accurate model (the circuit netlist), are the strong points of simulation-based techniques. However, the execution time is large for complex circuits (~ 100 variables) and prohibitive at system level, and without the proper constraints the algorithm may not converge to a good result. Some heuristic schemes exist to automate the process of defining the constraints [35]. However, automatic constraint defining mechanisms are not integrated in sizing tools and their application is somewhat circuit class specific. Cheng et al. [29] uses manually derived DC point equations to limit the search space for the transistors dimensions.

Being the high execution time the weaker point of these methods, some techniques had been proposed to cope with it. Kuo-Hsuan et al. [21] used equation-based techniques to derive an approximate initial solution. Cheng et al. [29] instead of solving the circuit by finding transistor sizes, solved it by finding the bias of the transistors first, and then, the transistor sizes are derived from the bias point using electric simulation. In MAELSTROM [31] and ANACONDA [32] the evaluation time is reduced by a parallel mechanism that shares the evaluation load among multiple computers.

2.1.2.3 Model-Based

For some simulation-based approaches, macro models, like neural-networks or support vector machines (SVM), are also used to reduce the execution time caused by the use of circuit simulator in the loop. These models are automatically generated using an electric simulator to evaluate the performance of the training set.

Unlike the equations-based modeling the learning based modeling application to general circuits is easier; however, there is still the tradeoff between accuracy and model size and generation time.

Alpaydin et al. [36] use a neural-fuzzy model combined with an evolutionary optimization strategy where some of the AC performance metrics are computed using an equation-based approach. De Bernardinis et al. [37] use a learning tool based in SVMs to represent the performance space of analog circuits. The performance space is modeled using the knowledge acquired from a training set via circuit simulation.

Wolfe et al. [38] present a performance macro-model based in a neural network. This model once constructed, is to be used to replace the SPICE [27] simulation during the synthesis of analog circuits, increasing the efficiency of the performance parameter estimates' computation. The training and validation data sets are constructed with discrete points, sampled over the design space. The work explores several sampling methodologies to adaptively improve model quality and applies a sizing rules methodology in order to reduce the design space and ensure the correct operation of analog circuits.

Barros et al. [1, 39] present a cell-level synthesis and optimization approach based on SVMs and evolutionary strategies. The SVM is used to dynamically model performance space and identify the feasible design space regions while at the same time the evolutionary techniques are looking for the global optimum. The evaluation is still done with HSPICE® to ensure accuracy, but the number of evaluation is reduced by using the SVM to prune the candidate solutions.

A different approach is the use of POFs to explore circuit tradeoffs during synthesis [40], and instead of using a model for the circuits, the non-dominated solutions are generated (prior to the design task) and the suitable solution is selected from the already sized solutions. In [41], hierarchically POFs are used to perform system-level sizing. The POF-based-design execution time is large if the setup time (the generation of the POFs) is considered, however with the correct models, the POFs can be generated in a context free manner making then suitable for reuse.

In Tables 2.1, 2.2 and 2.3 the several tools for analog sizing automation are summarized and, in Table 2.4, the specification translation tools based on the techniques applied are compared.

2.2 Motivation for Model-Based Optimization

According to McConaghy and Gielen [42], there is a great improvement on the efficiency of an optimization cycle for analog IC sizing using electrical simulators, if models containing knowledge about the circuit are used. In [42] is presented a study to analyze the impact of different models in the optimization process, which were conducted for several different techniques: polynomials [43], posynomials [44], genetic programming [45], feedforward neural networks [46], boosted feedforward

Table 2.1 Overview of analog sizing tools, part I

Tool/author		Circuits	Design plan/optimization	Evaluation	Robust design	Topology gen.	Layout gen.	Time setup/ execution	Code
IDAC [2]	1987	Analog cells	Design plan plus SA post-optimization	Equations	⦸	✘	After sizing	Months/few sec	Pascal
DELIGTH.SPICE [28]	1988	Analog cells	Feasible directions optimization	SPICE-like	✔	✘	✘	Moderate/18 h	–
OASYS [3]	1989	OPAMP	Design plan (includes backtracking features)	Equations	✔	Before	✘	6 months/3 s	LISP
BLADES [7]	1989	OPAMP	Expert system for analog design	Equations	⦸	Before	✘	Long/20 min	LISP
OPASYN [11]	1990	OPAMP	Steepest descent	Equations	✔	Before	After sizing	2 weeks/5 min	C/LISP
CAMP [8]	1990	OPAMP	Expert system, flexible architecture	SPICE-like	⦸	During	After sizing	−/−	TURBO PROLOG
OPTIMAN [13]	1990	OPAMP	SA	Analytical models	✘	✘	✘	−/1 min	PASCAL
SEAS [52]	1991	OPAMP	SA	Equations	✘	During	✘	−/−	C
DONALD [15]	1991	OPAMP	Equation solver (Newton–Raphson variant)	Equations	⦸	✘	✘	−/−	LISP/ FORTAN
Chang [53]	1992	ADC	Top-down constraint driven	Behavior models	⦸	During	After sizing	−/4–89 h	C++
STAIC [12]	1992	OPAMP	2 step optimization	Equations	⦸	✘	After sizing	Long/2 min	C++
MINLP [54, 55]	1992	OPAMP	Branch & bound	Equations and BSIM models	⦸	During	✘	6 months/ 1 min	–

Table 2.2 Overview of analog sizing tools, part II

Tool/author		Circuits	Design plan/optimization	Evaluation	Robust design	Topology gen.	Latyout gen.	Time setup/ execution	Code
Maulik et al. [16]	1993	OPAMP	Sequential quadratic programming	Equations and BSIM models	⦸	✘	✘	6 months/ 1 min	C
FRIDGE [30]	1994	OPAMP	SA	SPICE-like	✘	✘	✘	1 h/45 min	–
DARWIN [18]	1995	OPAMP	Genetic Algorithm (GA)	Small signal, analytical expressions	✘	During	✘	–/–	–
ISAID [9, 10]	1995	OPAMP	Qualitative reasoning + post optimization	Equation and qualitative reasoning	⦸	✘	✘	–/–	C/PROLOG
SD-OPT [24]	1995	$\sum\Delta$-modulator	SA	Equation adn behavioral simulation	✘	✘	✘	Long/ 1,5 week	–
FAST [22, 23]	1995	OPAMP	SA + Gradient	SPICE-like	✘	Before	✘	–/6 h	–
ASTRX/OBLX [17]	1996	Analog cells	SA	AWE equations	✘	✘	✘	few days/sec	C
Koza [56]	1997	Analog cells	GA	SPICE-like	✘	During	✘	–/–	C
GPCAD [57]	1998	OPAMP	Geometric programming	Posynomial models	✘	✘	✘	–/fast	MATLAB
Lohn [58]	1999	Filters	GA	SPICE-like	✘	During	✘	–/–	C
MAELSTROM [31]	1999	OPAM	GA + SA	SPICE-like	✘	✘	✘	–/3,6 h	C++
ANACONDA [32]	2000	OPAMP	Stochastic patten search	SPICE-like	✘	✘	✘	–/10 h	C++
Sripramong [59]	2002	OPAMP	GA	SPICE-like	✘	During	✘	–/3 days	C

Table 2.3 Overview of analog sizing tools, part III

Tool/author		Circuits	Design plan/ optimization	Evaluation	Robust design	Topology gen.	Layout gen.	Time setup/ execution	Code
Alpaydin [36]	2003	OPAMP	Evolutionary strategies + SA	Fuzzy + NN trained with SPICE-Llike	✔	✘	✘	−/45 min	–
Shoou-Jin [60]	2006	Passive filters	GA	Equations	✘	During	✘	−/−	–
Barros [1, 39]	2006	Analog cells	GA	SPICE-like + feasibility SVM models	✔	✘	✘	−/20 min	C
Castro-Lopez [61]	2008	OPAMP	SA + Powels method	SPICE-like	⦸	✘	✔	−/25 min	–
MOJITO [62], [63]	2009	OPAMP	GP(NSGA-II)	SPICE-like	✔	During	✘	−/< 7 days	Python
Pradhan [64]	2009	OPAMP,filter	Multi-objective SA	Layout aware MNA models	✘	✘	✘	−/16 min	C ++
Matsukawa [26]	2009	ADC	Convex optimization	Convex functions	✔	After	✘	−/−	MATLAb
Cheng [29]	2009	OPAMP	SA	Equations	✘	✘	✔	−/< 1 h	C
Hongying [65]	2010	OPAMP	GA with VDE	SPICE-like	✘	During	✘	−/−	–
Kuo-Hsuan [21]	2011	RFDA	Covex optimization stochastic fine tuning	Posynomial SPICE-like	✘	✘	✘	−/1 h	MATLAB
GENOM-POF [34]	2012	OPAMP	Multi-objective GA	SPICE-like	✘	✘	✔	−/10 min	JAVA

Table 2.4 Classification of specification translation tools based on applied techniques and abstraction level

		Abstraction level	
		System-level	Cell-level
Knowledge-based		TAGUS [4–6] (+) Fast execution time (+) Use of expert knowledge (−) Expert knowledge is difficult to capture (−) not optimal	IDAC [2]; OASYS [3]; BLADES [7]; CAMP [8]; ISAID [9], [10] (+) Fast execution time (+) Use of expert knowledge (−) Expert knowledge is difficult to capture (−) not optimal
Optimization-based	Equation	SD-OPT [24]; Doboli [25]; Matsukawa [26] (+) Fast execution time (+) Use of expert knowledge (−) Difficult derivation of some equations (−) Simplifications lead to lack of accuracy	OPASYN [11]; STAIC [12]; Kuo-Hsuan [21]; OPTIMAN [13]; DONALD [15]; ASTRX/OBLX [17]; DARWIN [18]; GPCAD [57] (+) Fast execution time (+)* Use of expert knowledge (+)* Automatic symbolic analysis (−) Difficult derivation of some equations (−) Simplifications lead to lack of accuracy
	Simulation		Kuo-Hsuan [21]; FASY [22, 23]; ASTRX/OBLX [17]; DARWIN [18]; DELIGTH.SPICE [28]; Cheng [29]; FRIDGE [30]; MAELSTROM [31]; ANACONDA [32]; Castro-Lopez [61]; GENOM-POF [34] (+) Easy to develop models (++) Accurate and flexible (−) Still requires expert knowledge (−) Long execution time (−) Limited to cell-level
	Model		Alpaydin [36]; De Bernardinis [37]; Wolfe [38]; Barros [1], [39] (+) Accurate and flexible (−) Limited to cell-level

* not present in all approaches

neural networks [47], multivariate adaptive regression splines [48], support vector machines [49] and Kriging [50]. The choice of the models was based on their performance, and the following modeling methods were considered:

- As reference models were used: a constant (set as the mean of the data), a linear model and a 2nd-order polynomial;
- CAFFEINE [45] tool used a modified form of genetic programming (GP), which restricts GP to canonical function forms via a grammar;
- Feed forward neural networks (FFNNs) [46] which used the state-of-art training algorithm OLMAM;
- Boosting [47] creates a "stack" of models, each model is learned on a weighted version of the data. The overall output is the average of the outputs of the individual models;
- Multivariate Adaptive Regression Splines (MARS) [48] are piecewise polynomials. In the constructive steps, input variables are iteratively added on as "as-needed" basis for greedily chosen sub-regions of input space. MARS scales to a high number of input variables but is locally accurate;
- Support vector machines (SVMs) transform inputs into a space of much higher dimension and do linear regression in that space. A fast-learning variant LS-SVM [49] was used;
- Kriging [50] originated in geostatistics, but it has been shown to be useful in optimization. In this model prediction is the value of nearby samples "corrected" by a correlated error calculation.

Of the several existing ways to improve the optimization process efficiency, the study indicates that the construction of all models was based on the use of a Design of Experiments (DOE) technique [51].

Since electrical simulation is the bottleneck of the simulator-in-loop techniques, improving efficiency roughly translates to reducing the number of simulations. For a proper comparison between different models, a point that must be taken into account it is the setup time, i.e., the time necessary to create the model, which generally produces a tradeoff between model performance (accuracy and/or range of applicability) and model setup time.

Table 2.5 presents a summary of the study for the different models. From Table 2.5, CAFEINE is the approach with the better performance concerning the prediction error, while the Polynomial approach has the worst.. Based on this study, it is fair to forecast that with the type of approach made in CAFFEINE available, it could replace the simulator in the loop of an optimization process. However, the setup time of this model is huge when compared to the remaining; a model that has a setup time higher than the overall execution time is a huge contradiction.

Table 2.5 Comparison between several models for sizing automation of ICs

Model	Date	Heuristics	Circuits	Simulator	Time setup/ execution	Lang.	Error prediction (%)
Polynomial [43]	2005	Polynomial	High-speed CMOS OTA, 13 inputs and 6 outputs	SPICE	1–4 min/ <10 min	Matlab	82,6
Posynomial [44]	2002	Posynomial			1–4 min/ <10 min	Matlab	61,7
CAFFEINE [45]	2005	Posynomial			12 h/ <10 min	Matlab	22,7
FFNNs [46]	2002	Neural networks			3,7 min/ <10 min	Matlab	41,7
Boosted FFNN [47]	2002	Neural networks			7 min/ <10 min	Matlab	43,2
MARS [48]	1991	Polynomial			5 min/ <10 min	Matlab	29,4
LS-SVM [49]	2002	Support vector machine			5 min/ <10 min	Matlab	45,9
Kriging [50]	1998	Geostatistics			5 min/ <10 min	Matlab	34,6

2.3 Conclusions

Despite the evolution verified in the high and low abstraction levels, both architecture's selection, sizing and layout optimization remain the focus of research in analog EDA methodologies. The industrial commercial tools follow closely the main trends in academia and R&D workgroups, focusing in the lower level of abstraction levels dealing with device sizing and layout description levels.

Although much has been accomplished in automatic design of analog circuits, the fact is that custom generators usable in industrial design environment are not available. In this survey, some of the most significant analog design automation tools for circuit-sizing were presented and analyzed to provide a better understanding of its advantages and shortcomings. The tools are classified according to the techniques used and the applicability to cell and (or) system level.

Particularly, the results of Sect. 2.2 present a real motivation for a model-based optimization. The opportunity to create a new and innovative model, with a good performance both in terms of accuracy and setup time, arises. In this work, the idea of acquire knowledge of a circuit and embedding it into the evolutionary optimization kernel is explored. However, the model is used to guide the optimization kernel in a more efficient search of the solution space rather than replacing the usage of the circuit simulator to evaluate the performance of the circuit. The methodology adopted is to automatically generate a model that estimates how move to better solutions during the optimization. Chapter 3, describes the Gradient Model introduced in this work, and how it is automatically generated using DOE

with two alternatives strategies, the Full Factorial Design and the Fractional Factorial Design. The model is then integrated into the synthesis tool AIDA, as will be presented in Chap. 4, and the obtained results are shown in Chap. 5.

References

1. M.F.M. Barros, J.M.C. Guilherme, N.C.G. Horta, *Analog circuits and systems optimization based on evolutionary computation techniques* (Springer, Berlin, 2010)
2. M.G.R. Degrauwe, O. Nys, E. Dijkstra et al., IDAC: an interactive design tool for analog CMOS circuits. IEEE J. Solid-State Circuits **22**(6), 1106–1116 (1987)
3. R. Harjani, R.A. Rutenbar, L.R. Carley, OASYS: a framework for analog circuit synthesis. IEEE Trans. Comput.Aided Des. Integr. Circuits Syst. **8**(12), 1247–1266 (1989)
4. N.C. Horta, J.E. Franca, High-level data conversion synthesis by symbolic methods, in *Proceedings of the IEEE International Symposium on Circuits and Systems,* vol. 4 (1996), pp. 802–805
5. N. Horta, Analogue and mixed-signal systems topologies exploration using symbolic methods. Analog Integr. Circ. Sig. Process **31**(2), 161–176 (2002)
6. N.C. Horta, J.E. Franca, Algorithm-driven synthesis of data conversion architectures. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **16**(10), 1116–1135 (1997)
7. F. El-Turky, E.E. Perry, BLADES: an artificial intelligence approach to analog circuit design. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **8**(6), 680–692 (1989)
8. B.J. Sheu, J.C. Lee, A.H. Fung, Flexible architecture approach to knowledge-based analogue IC design. IEEE Proc. G Circuits Devices Syst. **137**(4), 266–274 (1990)
9. C.A. Makris, C. Toumazou, Analog IC design automation. II. Automated circuit correction by qualitative reasoning. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **14**(2), 239–254 (1995)
10. C. Toumazou, C.A. Makris, Analog IC design automation. I. Automated circuit generation: new concepts and methods. IEEE Trans. Comput.Aided Des. Integr.Circuits Syst. **14**(2), 218–238 (1995)
11. H.Y. Koh, C.H. Sequin, P.R. Gray, OPASYN: a compiler for CMOS operational amplifiers. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **9**(2), 113–125 (1990)
12. J.P. Harvey, M.I. Elmasry, B. Leung, STAIC: an interactive framework for synthesizing CMOS and BiCMOS analog circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **11**(11), 1402–1417 (1992)
13. G.G.E. Gielen, H.C.C. Walscharts, W.M.C. Sansen, Analog circuit design optimization based on symbolic simulation and simulated annealing. IEEE J. Solid-State Circuits **25**(3), 707–713 (1990)
14. G.G.E. Gielen, H.C.C. Walscharts, W.M.C. Sansen, ISAAC: a symbolic simulator for analog integrated circuits. IEEE J. Solid-State Circuits **24**(6), 1587–1597 (1989)
15. K. Swings, W. Sansen, DONALD: a workbench for interactive design space exploration and sizing of analog circuits, in *Proceedings of the European Conference on Design Automation,* (1991), pp. 475–479
16. P.C. Maulik, L.R. Carley, D.J. Allstot, Sizing of cell-level analog circuits using constrained optimization techniques. IEEE J. Solid-State Circuits **28**(3), 233–241 (1993)
17. E.S. Ochotta, R.A. Rutenbar, L.R. Carley, Synthesis of high-performance analog circuits in ASTRX/OBLX. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **15**(3), 273–294 (1996)
18. W. Kruiskamp, D. Leenaerts, DARWIN: CMOS opamp synthesis by means of a genetic algorithm, in *Proceedings of the Design Automation Conference,* (1995), pp. 433–438

19. M. del Mar Hershenson, S.P. Boyd, T.H. Lee, GPCAD: a tool for CMOS op-amp synthesis, in *InternationBaal Conference on Computer-Aided Design, Digest of Technical Papers of the IEEE/ACM,* (1998), pp. 296–303
20. G. Gielen, P. Wambacq, W.M. Sansen, Symbolic analysis methods and applications for analog circuits: a tutorial overview. Proc. IEEE **82**(2), 680–692 (1994)
21. M. Kuo-Hsuan, P. Po-Cheng, C. Hung-Ming, Integrated hierarchical synthesis of analog/RF circuits with accurate performance mapping, in *Symposium on Quality Electronic Design (ISQED),* (2011), pp. 1–8
22. A. Torralba, J. Chavez, L.G. Franquelo, FASY: a fuzzy-logic based tool for analog synthesis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **15**(7), 705–715 (1996)
23. A.J. Torralba, J. Chavez, L.G. Franquelo, Fuzzy-logic-based analog design tools. IEEE Micro **16**(4), 60–68 (1996)
24. F. Medeiro, B. Perez-Verdu, A. Rodriguez-Vazquez et al., A vertically integrated tool for automated design of Sigma&Delta modulators. IEEE J. Solid-State Circuits **30**(7), 762–772 (1995)
25. A. Doboli, N. Dhanwada, A. Nunez-Aldana et al., A two-layer library-based approach to synthesis of analog systems from VHDL-AMS specifications. ACM Trans. Des. Autom. Electron. Syst. **9**(2), 238–271 (2004)
26. K. Matsukawa, T. Morie, Y. Tokunaga et al., Design methods for pipeline delta-sigma A-to-D converters with convex optimization, in *Design Automation Conference,* (2009), pp. 690–695
27. L.W. Nagel, *SPICE2: A Computer Program to Simulate Semiconductor Circuits* (EECS Department, University of California, Berkeley, 1975)
28. W. Nye, D.C. Riley, A. Sangiovanni-Vincentelli et al., DELIGHT.SPICE: an optimization-based system for the design of integrated circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **7**(4), 501–519 (1988)
29. L. Cheng-Wu, S. Pin-Dai, S. Ya-Ting et al., A bias-driven approach for automated design of operational amplifiers., in *International Symposium on VLSI Design, Automation and Test,* (2009), pp. 118–121
30. F. Medeiro, F.V. Fernandez, R. Dominguez-Castro et al., A Statistical Optimization-based Approach For Automated Sizing Of Analog Cells., in *Conference on Computer-Aided Design,* (1994), pp. 594–597
31. M. Krasnicki, R. Phelps, R.A. Rutenbar et al., MAELSTROM: efficient simulation-based synthesis for custom analog cells, in *Design Automation Conference,* (1999), pp. 945–950
32. R. Phelps, M. Krasnicki, R.A. Rutenbar et al., Anaconda: simulation-based synthesis of analog circuits via stochastic pattern search. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **19**(6), 703–717 (2000)
33. R. Castro-Lopez, O. Guerra, E. Roca, F. Fernandez, An integrated layout-synthesis approach for analog ICs. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(7), 1179–1189 (2008)
34. N. Lourenço, N. Horta, GENOM-POF: Multi-Objective evolutionary synthesis of analog ICs with corners validation, in *GECCO' 12: Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference*, (2012), pp. 1119–1126
35. T. Massier, H. Graeb, U. Schlichtmann, The sizing rules method for CMOS and bipolar analog integrated circuit synthesis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(12), 2209–2222 (2008)
36. G. Alpaydin, S. Balkir, G. Dundar, An evolutionary approach to automatic synthesis of high-performance analog integrated circuits. IEEE Trans. Evol. Comput. **7**(3), 240–252 (2003)
37. F. De Bernardinis, M.I. Jordan, A. SangiovanniVincentelli, Support vector machines for analog circuit performance representation, in *Design Automation Conference,* (2003), pp. 964–969
38. G.A. Wolfe, Performance Macro-Modeling Techiniques for Fast Analog Circuit Synthesis, University of Cincinnati, 2004

39. M. Barros, J. Guilherme, N. Horta, GA-SVM optimization kernel applied to analog IC design automation., in *IEEE Internation Conference on Electronics*, (2006), pp. 486–489
40. R. Castro-Lopez, E. Roca, F.V. Fernandez, Multimode pareto fronts for design of reconfigurable analogue circuits. Electron. Lett. **45**(2), 95–96 (2009)
41. E. Denize, G. Dundar, Hierarchical performance estimation of analog blocks using pareto fronts. Ph.D., Research in Microelectronics and Electronics, 2010
42. T. McConaghy, G. Gielen, Analysis of simulation-driven numerical performance modeling techniques for application to analog circuit optimization, in *IEEE International Symposium on Circuits and Systems (ISCAS),* (2005), pp.1298–1301
43. T. McConaghy, G. Gielen, Analysis of simulation-driven numerical performance modeling techniques for application to analog circuit optimization, in *IEEE International Symposium on Circuits and Systems (ISCAS)*, (2005), pp. 1298–1301
44. W. Daems, G. Gielen, W. Sansen, Simulation-based generation of posynomial performance models for sizing of analog integrated circuits. IEEE Trans. CAD **22**(5), 517–534 (2003)
45. T. McConaghy, T. Eecklelaert, G. Gielen, CAFFEINE: Template-free symbolic model generation of analog circuits via canonical form functions and genetic programming, in *Design, Automation and Teste in Europe* **2**, (2005), pp. 1082–1087
46. N. Ampazis, S.J. Perantonis, OLMAN neural networks toolbox for Matlab (2002), http://iit.demokritos.gr/∼abazis/toolbox/
47. R.E. Schapire, The boosting approach to machine learning: an overview, in *MSRI Workshop on Nonlin. Estimation and Classification*, (2002)
48. J.H. Friedman, Multivariate adaptive regression splines. Ann. Stat **19**, 1–141 (1991)
49. H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik, in *Adv. in Neural Information Processing Systems 9,* ed. by M.C. Mozer, J.I. Jordan, T. Petscbe.Support vector regression machines, (MIT Press, Cambridge, 1997), pp. 155–161
50. D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions. J. Glob. Opt **13**(4), 455–492 (1998)
51. D.C. Montgomery, *Design and Analysis of Experiments*, 5th edn. (John Wiley and Sons, New York, 2001)
52. Z.Q. Ning, T. Mouthaan, H. Wallinga, SEAS: a simulated evolution approach for analog circuit synthesis, in *Proceedings of the IEEE Custom Integrated Circuits Conference,* (1991) pp. 5.2.1–5.2.4
53. H. Chang, A. Sangiovanlli-Vincentelli, F. Balarin et al., A top-down, constraint-driven design methodology for analog integrated circuits, in *Proceedings of the IEEE Custom Integrated Circuits,* vol. 3–6 (1992), pp. 8.4.1–8.4.6
54. P.C. Maulik, L.R. Carley, R.A. Rutenbar, A mixed-integer nonlinear programming approach to analog circuit synthesis, in *Proceedings of Design Automation Conference,* (1992), pp. 698–703
55. P.C. Maulik, L.R. Carley, R.A. Rutenbar, Integer programming based topology selection of cell-level analog circuits. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **14**(4), 401–412 (1995)
56. J.R. Koza, F.H. Bennett III, D. Andre et al., Automated synthesis of analog electrical circuits by means of genetic programming. IEEE Trans. Evol. Comput. **1**(2), 109–128 (1997)
57. M. Del Mar Hershenson, S. P. Boyd, T. H. Lee, GPCAD: a tool for CMOS op-amp synthesis, in *International Conference on Computer-Aided Design, Digest of Technical Papers of the IEEE/ACM,*, vol. 8–12 (1998), pp. 296–303
58. J.D. Lohn, S.P. Colombano, A circuit representation technique for automated circuit design. IEEE Trans. Evol. Comput. **3**(3), 205–219 (1999)
59. T. Sripramong, C. Toumazou, The invention of CMOS amplifiers using genetic programming and current-flow analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **21**(11), 1237–1252 (2002)
60. C. Shoou-Jinn, H. Hao-Sheng, S. Yan-Kuin, Automated passive filter synthesis using a novel tree representation and genetic programming. IEEE Trans. Evol. Comput. **10**(1), 93–100 (2006)

61. R. Castro-Lopez, O. Guerra, E. Roca et al., An Integrated Layout-Synthesis Approach for Analog ICs. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **27**(7), 1179–1189 (2008)
62. T. McConaghy, P. Palmers, M. Steyaert et al., Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks. IEEE Trans. Evol. Comput. **99**, 1–14 (2011)
63. P. Palmers, T. McConnaghy, M. Steyaert et al., *Massively Multi-Topology Sizing of Analog Integrated Circuits* (Design, Automation and Teste in Europe, 2009), pp. 706–711
64. A. Pradhan, R. Vemuri, Efficient synthesis of a uniformly spread layout aware pareto surface for analog circuits, in *22nd International Conference on VLSI Design*, (2009), pp. 131–136
65. Y. Hongying, H. Jingsong, Evolutionary design of operational amplifier using variable-length differential evolution algorithm, in *International Conference on Computer Application and System Modeling (ICCASM)*, (2010), pp. V4-610–V4-614
66. E. Roca, R. Castro-Lopez, F.V. Fernandez, Hierarchical synthesis based on pareto-optimal fronts, *in European Conference on Circuit Theory and Design*, (2009), pp. 755–758

Springer

http://www.springer.com/978-3-319-02188-1

Electronic Design Automation of Analog ICs combining Gradient Models with Multi-Objective Evolutionary Algorithms

Rocha, F.A.E.; Martins, R.M.F.; Lourenço, N.C.C.; Horta, N.C.G.

2014, XI, 69 p. 39 illus., Softcover
ISBN: 978-3-319-02188-1