



Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and Computational
Mathematics (IMACM)

Preprint BUW-IMACM 11/08

L. De Tommasi, T.G.J. Beelen, M.F. Sevat,
J. Rommes and E.J.W. ter Maten

**Multi-objective optimization of RF circuit blocks via
surrogate models and NBI and SPEA2 methods**

April 2011

<http://www.math.uni-wuppertal.de>

Multi-objective optimization of RF circuit blocks via surrogate models and NBI and SPEA2 methods

L. De Tommasi, T.G.J. Beelen, M.F. Sevat, J. Rommes and E.J.W. ter Maten

Abstract Multi-objective optimization techniques can be categorized globally into deterministic and evolutionary methods. Examples of such methods are the Normal Boundary Intersection (NBI) method and the Strength Pareto Evolutionary Algorithm (SPEA2), respectively. With both methods one explores trade-offs between conflicting performances. Surrogate models can replace expensive circuit simulations so enabling faster computation of circuit performances. As surrogate models of behavioral parameters and performance outcomes, we consider look-up tables with interpolation and Neural Network models.

1 Introduction: Multi-Objective Optimization Problem

The design parameters (input) \mathbf{x} and performances, or performance parameters, (output) \mathbf{f} are assumed to be in the Design Space \mathcal{D} and the Performance Space \mathcal{P} , respectively. We assume that \mathcal{D} is feasible, i.e., all $\mathbf{x} \in \mathcal{D}$ satisfy the imposed constraints (reflected by inequalities for a function $\mathbf{c}(\mathbf{x})$). Also the $\mathbf{f} \in \mathcal{P}$ can be con-

L. De Tommasi

United Technology Research Center Ireland Ltd, Lee Mills House, Prospect Row, Cork, Ireland,
e-mail: Luciano.De.Tommasi@gmail.com

T.G.J. Beelen, J. Rommes

NXP Semiconductors, Central R&D, High Tech Campus 46, 5656AE Eindhoven, the Netherlands,
e-mail: {Theo.G.J.Beelen, Joost.Rommes}@nxp.com

M.F. Sevat

LPD, Nieuwe Eyckholt 292e, 6419 DJ Heerlen, the Netherlands, e-mail: Sevat@scarlet.nl

E.J.W. ter Maten

Eindhoven University of Technology, Dep. Mathematics and Computer Science, CASA, P.O. Box 513, 5600 MB Eindhoven, the Netherlands, e-mail: {E.J.W.ter.Maten}@tue.nl
Bergische Universität Wuppertal, FB C, AMNA, Bendahler Str. 29/503, D-42285 Wuppertal, Germany, e-mail: {Jan.ter.Maten}@math.uni-wuppertal.de

strained (reflected by inequalities for a function $\mathbf{g}(\mathbf{f})$). We define \mathcal{D} and \mathcal{P} by

$$\begin{aligned}\mathcal{D} &= \{\mathbf{x} \in \mathbb{R}^m \mid \mathbf{c}(\mathbf{x}) \leq 0\}, \quad \text{with } \mathbf{c}(\mathbf{x}) \in \mathbb{R}^q, \\ \mathcal{P} &= \{\mathbf{f} \in \mathbb{R}^n \mid \exists \mathbf{x} \in \mathcal{D} \text{ } \mathbf{f} = \mathbf{f}(\mathbf{x}), \mathbf{g}(\mathbf{f}) \leq 0\}, \quad \text{with } \mathbf{g}(\mathbf{f}) \in \mathbb{R}^p.\end{aligned}$$

The design problem is a multi-objective optimization problem, i.e. a constrained simultaneous minimization of several performances $f_k(\mathbf{x})$

$$\text{Minimize}_{\mathbf{x} \in \mathcal{D}} \mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix} \quad \text{such that } \mathbf{g}(\mathbf{f}) \leq 0.$$

A simple single-objective optimization can be done by constrained minimizing a weighted sum of performances

$$\text{Minimize}_{\mathbf{x} \in \mathcal{D}} f(\mathbf{x}) = \sum_i k_i \cdot f_i(\mathbf{x}) \quad \text{such that } \mathbf{g}(\mathbf{f}) \leq 0.$$

Here obvious problems arise. The multi-objective problem admits multiple solutions, whereas the single objective problem admits isolated solutions. No rigorous criteria exist to choose the weights $\{k_i\}$. In practice, several optimization runs (with different $\{k_i\}$) are needed to find a suitable solution of the design problem. More basically, in general, there is no single design $\mathbf{x} \in \mathcal{D}$ that can minimize all performances f_k , $k = 1, \dots, n$ simultaneously. The set of solutions of the multi-objective optimization problem are Pareto optimal, i.e. it is only possible to improve one performance at the cost of others. This leads to the concept of ‘dominance’. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$, then $\mathbf{a} = (a_1, \dots, a_n)$ dominates $\mathbf{b} = (b_1, \dots, b_n)$ if and only if

$$\mathbf{a} \prec \mathbf{b} :\Leftrightarrow \forall_{i \in \{1, \dots, n\}} (a_i \leq b_i) \wedge \exists_{i \in \{1, \dots, n\}} (a_i < b_i).$$

A performance vector \mathbf{f}^* is said to be Pareto-optimal if it is non-dominated within \mathcal{P} , i.e. $\neg \exists_{\mathbf{f} \in \mathcal{P}} [\mathbf{f} \prec \mathbf{f}^*]$. The set of all Pareto-optimal points in \mathcal{P} is called the *Pareto Front* of \mathcal{P} . The corresponding set in \mathcal{D} is called the *Pareto Source*.

2 Surrogate Modeling

Recently several techniques emerged to compute the Pareto Front. The most obvious one deals with trade-off analysis from available data. Hence (in principle) no new simulations are needed. The search for Pareto optimal points is done by applying *non-dominated sorting*. An efficient implementation by Yi Cao [2] is found on the MATLAB central website (mex function).

An alternative is to perform *Performance Space Exploration*. Here one builds one or more surrogate models, each of them derived by a set of circuit simulations (samples), starting from an initial design. With adaptive sampling the models are im-

proved [10], which requires accessibility of a (circuit) simulator. The models can be generated by several techniques (including look-up tables with interpolation and neural network models). The approach can also be applied to derive symbolic models, that may include a new trade-off problem between Fitness (approximation error) and Complexity [5]. In practice, in both cases, the number of parameters is still restricting (up to 6-10). Here interesting progress is derived using a nearly orthogonal and space-filling Latin Hypercube [1, 3].

Writing $\mathbf{x} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$ one may reduce the parameter dependency in the surrogate modeling and consider behavioral parameters $\mathbf{b} = \mathbf{b}(\mathbf{x}^{(1)})$, followed by performance computations $\mathbf{f} = \mathbf{f}(\mathbf{b}, \mathbf{x}^{(2)})$ using algebraic expressions. Error amplification from \mathbf{b} to \mathbf{f} may occur (see [7] for the IIP2 performance of a Low Noise Amplifier).

Clearly, when the surrogate models are available one can use them in the forward modeling in more cheaply generating additional data for improving trade-off analysis. However, the models can also be used in reverse modeling, i.e. in applying them to dedicated Pareto Front methods like NBI (Normal Boundary Intersection method [4, 9]) and SPEA2 (Strength Pareto Evolutionary Algorithm 2 [11]).

3 NBI - Normal Boundary Intersection Method

Assume $\mathbf{f} = \begin{pmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \end{pmatrix} : \mathcal{D} \rightarrow \mathcal{P}$. The Algorithm [4, 9] looks like

1. Determine a minimizer \mathbf{x}^{*k} of each $f_k(\mathbf{x})$. Let $\mathbf{f}^{*k} = \mathbf{f}(\mathbf{x}^{*k})$. This is a global optimization problem for each $f_k(\mathbf{x})$ and critical for the next step. MATLABs **fmincon.m** allows nonlinear constraints. It implements a *local* optimization procedure: it starts from a user-specified point and may stop in a local minimum. More robust was **direct.m** [8] which provides global optimization using Lipschitzian optimization. It only allows domain boundary constraints.
2. Determine the straight line \mathcal{L} (convex hull of the individual minima) in \mathcal{P} between \mathbf{f}^{*1} and \mathbf{f}^{*2} .
3. Determine the normal \mathbf{n} to this line in direction of decreasing \mathbf{f} . Next
 - Original [9]: Select N points $\mathbf{f}_k = \lambda_k \mathbf{f}^{*1} + (1 - \lambda_k) \mathbf{f}^{*2}$, $\lambda_k \in [0, 1]$ on \mathcal{L} .
 - Modification: Select N points $\mathbf{x}_k = \lambda_k \mathbf{x}^{*1} + (1 - \lambda_k) \mathbf{x}^{*2}$, $\lambda_k \in [0, 1]$ on \mathcal{G} , line in \mathcal{D} . For convex \mathbf{f} we have $\mathbf{f}_{\mathbf{x}_k} = \mathbf{f}(\mathbf{x}_k) \prec \mathbf{f}_k \implies$ redefine $\mathbf{f}_k = \min(\mathbf{f}_k, \mathbf{f}_{\mathbf{x}_k})$.
4. For each \mathbf{f}_k determine $\mathbf{p}_k \in \mathcal{P}$ that maximizes the distance t along \mathbf{n} , starting in \mathbf{f}_k . Without constraints these \mathbf{p}_k are on the Pareto Front. We solve

$$\max_{(t, \mathbf{x}) \in \mathbb{R} \times (\mathcal{D} \cap \mathbf{f}^{-1}(\mathcal{P}))} t, \quad \text{subject to} \quad \mathbf{p}(\mathbf{x}) = \mathbf{f} + t\mathbf{n},$$

where F is a point of the convex hull of the individual minima. Note, that \mathbf{x} has to be feasible. Also these are global optimization problems, but less critical. Here the starting point allows fmincon to provide good results. When during the

maximization process a constraint in \mathcal{P} is encountered this process is stopped, say with performance vector $\tilde{\mathbf{f}}$. This does not necessarily mean that $\tilde{\mathbf{f}}$ is located on the Pareto Front: there may be a vector $\hat{\mathbf{f}} \prec \tilde{\mathbf{f}}$ that also satisfies that constraint. We apply a refinement procedure. Let $\tilde{\mathbf{x}} \in \mathcal{D}$ with $\tilde{\mathbf{f}} = \tilde{\mathbf{f}}(\tilde{\mathbf{x}})$. Next

- Determine four neighboring points $\tilde{\mathbf{x}}_N, \tilde{\mathbf{x}}_E, \tilde{\mathbf{x}}_S, \tilde{\mathbf{x}}_W$ at a small distance from $\tilde{\mathbf{x}}$ and calculate $\mathbf{f}(\tilde{\mathbf{x}}_K)$ ($K = N, E, S, W$). Compare step 6 of the SPEA2 Algorithm in Section 4.
- Replace $\tilde{\mathbf{f}}$ by the best performance vector (based on the dominance relation) out of the set $\{\tilde{\mathbf{f}}, \mathbf{f}(\tilde{\mathbf{x}}_N), \dots, \mathbf{f}(\tilde{\mathbf{x}}_W)\}$.

This still does not guarantee a point on the Pareto Front, it just gives an improvement. In general a more sophisticated approach is needed.

4 SPEA2 - Strength Pareto Evolutionary Algorithm 2

The SPEA2 Algorithm [11] allows constraints both in \mathcal{D} and in \mathcal{P} . It looks like

- Initialize an internal \mathcal{I} and an external \mathcal{E} set of points in \mathcal{P} (last being approximations of Pareto Front).
- Iteration loop
 1. $\mathcal{E}^c = \text{copy}(\mathcal{E})$. $\mathcal{U} = \mathcal{I} \cup \mathcal{E}^c$.
 2. Determine fitness of individuals in \mathcal{U} [‘fitter’ when not dominated in \mathcal{P} and not too close to each other; impose constraints in \mathcal{P}].
 3. Update \mathcal{E} with fittest individuals from \mathcal{U} .
 4. Select individuals from \mathcal{U} , randomly based; ‘fitter’ points have a higher probability in being chosen.
 5. Recombine selected individuals. This exploits convexity using a randomly chosen weighting.
 6. Mutate recombined individuals. By properly defining the probability density function in mutating the result (f.i. after a gradient calculation) one can push the convex hull in \mathcal{P} to the Pareto front.
 7. Repopulate \mathcal{I} with mutated individuals.
 8. Verify iteration termination criterion.
- Output \mathcal{E} as best approximation found to the source of the Pareto front.

5 Examples

A good testing example appeared to be $p := f_1(x, y) = x^2 + (y - 1)^2$, $q := f_2(x, y) = (x - 2)^2 + y^2$, for $(x, y) \in [x_L, x_U] \times [y_L, y_U]$ and $(p, q) \in [p_L, p_U] \times [q_L, q_U]$. Observe that \mathbf{f} is convex. By considering the mapping of vertical and horizontal lines in \mathcal{D} into \mathcal{P} one can obtain impressions of the Pareto Front to check the outcomes of the

algorithms. One can also observe the effect of constraints.

A more realistic example is provided by a weakly nonlinear, narrowband Low Noise Amplifier (LNA)

- Design parameters: $\mathbf{x}^{(1)} = (W, L, L_s, L_m, f, V_{GS})$.
- Extra circuit parameters $\mathbf{x}^{(2)} = (Z_s, Z_l)$.
- Typical circuit performances $\mathbf{f} = \mathbf{f}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = (P, A_v, \Gamma_a, IIP2, IIP3, NF)$.

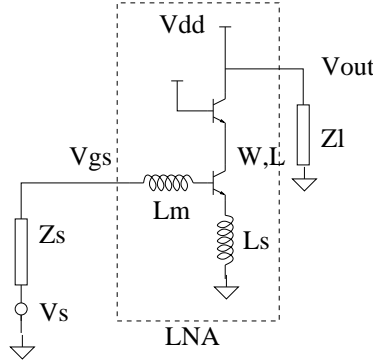


Fig. 1 A weakly nonlinear, narrowband, low noise amplifier (LNA) [6, 7]. Design parameters: W, L are transistor width and length; L_s, L_m are inductances; V_{GS} is the gate-source bias voltage difference; f is the frequency. Z_s, Z_l are the source and load impedances. Performances: power P , voltage gain A_v , input reflection Γ_a , 2nd order and 3rd order linearity $IIP2, IIP3$, noise figure NF .

We considered reverse modeling using look-up table models vs analytic expressions, both with constrained optimization.

- Normalized design constraints: $0 < W_n < 1$ and $0 < L_{mn} < 0.6$.
- Performance constraints: $A_v > 13$ dB, $\Gamma_a < -10$ dB, $\min(IIP2, IIP3) > 0$ dBm.
- (O1) Minimize P and maximize $IIP3$ and (O2) maximize A_v and maximize $IIP2$.

For (O1) NBI and SPEA2 worked successfully using surrogate models based on neural networks. For (O2) we used look-up table models. Here the NBI method (using fmincon) failed in finding a global minimum. Fig 2 shows the SPEA2 result.

6 Conclusions

Direct modeling of performances was more robust than modeling of intermediate ‘behavioral’ parameters. We considered look-up tables and applied interpolation. Also the size of tables was investigated. Neural network models were accurate, but expensive in generating.

The NBI method was improved in several ways. DIRECT provided a robust global optimizer for the start. Also the start of the directional optimization step was improved. Without constraints it covers the whole Pareto front in nice detail. However, with constraints, as above in (O2), still more work has to be done.

SPEA2 is more robust than NBI. Constraints can be applied on both design variables and on performances (including those not involved in the trade-off). The results were confirmed by considering a Low Noise Amplifier.

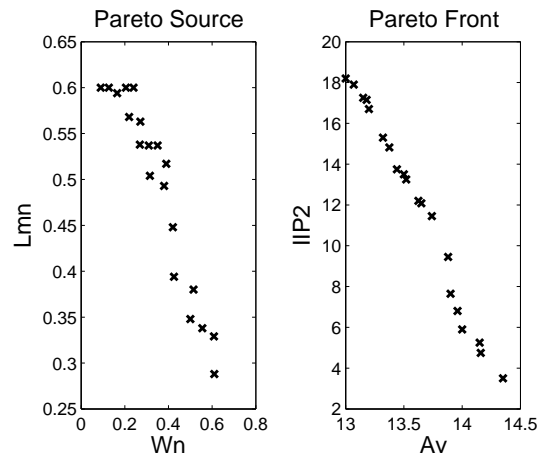


Fig. 2 Pareto Front determined by SPEA2 for (O2). This involved reverse modeling using look-up table models (100x100 meshpoints). Here NBI (using fmincon) failed in finding a global minimum.

Acknowledgements The work by the first (LDT, Univ. of Antwerp) and fourth (JR, NXP) author was funded by the EU Marie Curie FP7 MTKI-CT-2006-042477 project O-MOORE-NICE!

References

1. Beelen, T.G.J., ter Maten, E.J.W., Sihalohe, H.J., van Eijndhoven, S.J.L.: Behavioral modeling of the dominant dynamics in input-output transfer of linear(ized) circuits. *Procedia Computer Science*, Vol. 1, Issue 1, pp. 347–355 (2010)
2. Cao, Y.: <http://www.mathworks.com/matlabcentral/fileexchange/17251-pareto-front> (2007, 2008)
3. Cioppa, T.M., Lucas, T.W.: Efficient nearly orthogonal and space-filling Latin Hypercubes. *Technometrics*, 49-1, pp. 45–55 (2007)
4. Das, I., Dennis, J.E.: Normal-Boundary Intersection: A new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM J. Optim.* 8-3, pp. 631–657 (1998)
5. DataModeler: <http://www.evolved-analytics.com>
6. De Tommasi, L., Gorissen, D., Croon, J., Dhaene, T.: Surrogate modeling of low noise amplifiers based on transistor level simulations. In: Roos, J., Costa, R.J. (Eds.): *Scientific Computing in Electrical Engineering SCEE 2008. Mathematics In Industry*, Vol. 14, Springer-Verlag, Berlin Heidelberg, pp. 225–232 (2010)
7. De Tommasi, L., Rommes, J., Beelen, T., Sevat, M., Croon, J.A., Dhaene, T.: Forward and reverse modeling of low noise amplifiers based on circuit simulations. In: Benner, P., Hinze, M., ter Maten, E.J.W. (Eds.): *Model Reduction for Circuit Simulation. Lect. Notes in Electrical Engineering*, Vol. 74, Springer-Verlag, Berlin Heidelberg, pp. 111–124 (2011)
8. Finkel, D.E.: Global optimization with the direct algorithm. PhD-Thesis North Carolina State University (2005). <http://pages.cs.wisc.edu/~ferris/cs726/direct.m>
9. Stehr, G., Gräß, H.E., Antreich, K.J.: Analog performance space exploration by Normal-Boundary Intersection and by Fourier-Motzkin elimination. *IEEE Trans. Comp.-Aided Design of Integr. Circ. and Systems*, 26-10, pp. 1733–1745 (2007)
10. SUMO toolbox: http://www.sumo.intec.ugent.be/?q=sumo_toolbox
11. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Techn. Report TIK Report 103, ETH Zürich, http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2_c_source.html (2001)