

Combined simplex-trust-region optimization algorithm for automated IC design

Árpád Bűrmen, Iztok Fajfar, and Tadej Tuma
 Faculty of Electrical Engineering, University of Ljubljana
 Tržaška cesta 25, SI-1000 Ljubljana
 E-mail: arpadb@fides.fe.uni-lj.si

Abstract—One of the basic algorithms employed in IC design automation is parametric optimization. It is used when a minimum of a so-called cost function is sought. There exist many optimization algorithms. One of the most successful ones is the Box simplex algorithm. The main drawback of this algorithm that was observed in practice is its slow progress in the last stages of the search. The paper addresses this issue by combining it with a trust-region based algorithm. The combined algorithm was implemented in the SPICE OPUS tool for circuit optimization and tested on several real-world IC design problems. The results show that the combined algorithm exhibits faster convergence and in most cases results in a lower final cost function value than the Box simplex algorithm.

I. INTRODUCTION

Many integrated circuit (IC) design problems (most notably circuit sizing) can be reduced to solving

$$\arg \min_{x \in \mathcal{A}} f(x)$$

where x represents the vector of n parameters, $\mathcal{A} \subseteq \mathbb{R}^n$ is the design space, and $f(x)$ is the so-called cost-function (CF) which represents the circuit's performance. Lower CF value corresponds to a circuit with better performance. Let x^i denote the i -th component of vector x . In this paper the design space is box shaped and is defined by constraints $b^i \leq x^i \leq B^i, i = 1, \dots, n$. b and B are the vectors of lower and upper bounds on parameters.

In past research the Box simplex [1] algorithm was found to be very successful in solving IC sizing problems [2]. The algorithm manipulates a set of points also referred to as the simplex. The basic idea is to mirror and possibly contract the point with the highest CF value to the centroid of the remaining points. By repeating this operation the simplex gradually shrinks and moves toward a point in space with a lower CF value. The termination criterion of the algorithm is based on the size of the simplex.

The Box simplex algorithm comes from the family of direct search algorithms [3] that require no CF gradient evaluations. As the sensitivity analysis is not fully implemented in most circuit simulators, such methods are the first choice for circuit optimization.

Simplex algorithms are very popular. A large part of their popularity is owed to the fact that they are very simple to implement and have some noise resistance properties. The fact that they work with a population of points gives them some global search capabilities. In the initial part of the search

simplex algorithms progress very rapidly. But as they close in on a CF minimum their convergence deteriorates [4], [5]. Figure 1 demonstrates this shortcoming.

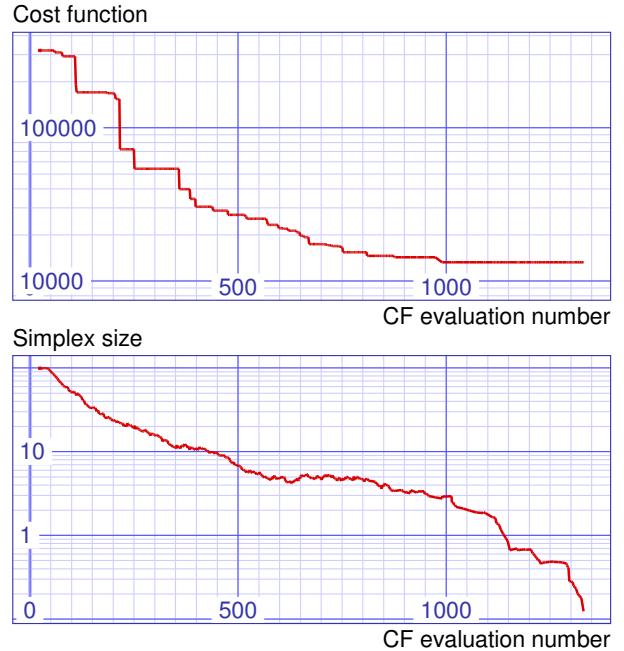


Fig. 1. A plot of the best point cost function (CF) value and the relative simplex size with respect to CF evaluation number for the digital delay testcase.

A so-called cost profile at the final point (x_f) found by the simplex algorithm for the digital delay test problem is displayed in figure 2. Every curve represents a cross section of the CF on a line through x_f along the direction of one of the optimization parameters. All curves cross at the same point at $x = 0$. This point represents the final point x_f . Two things are clearly demonstrated by figure 2. The cost function contains a fairly large amount of noise and local minima, and secondly the final point at $x = 0$ is not even a local minimum. The former observation was the reason for choosing a robust and noise resistant optimization algorithm (Box simplex method) in our past research [2]. The latter one, however, shows that despite a fairly large number of CF evaluations the simplex method stopped prematurely.

We expect a reduction in the number of CF evaluations and a reduction of the final CF value if the simplex algorithm

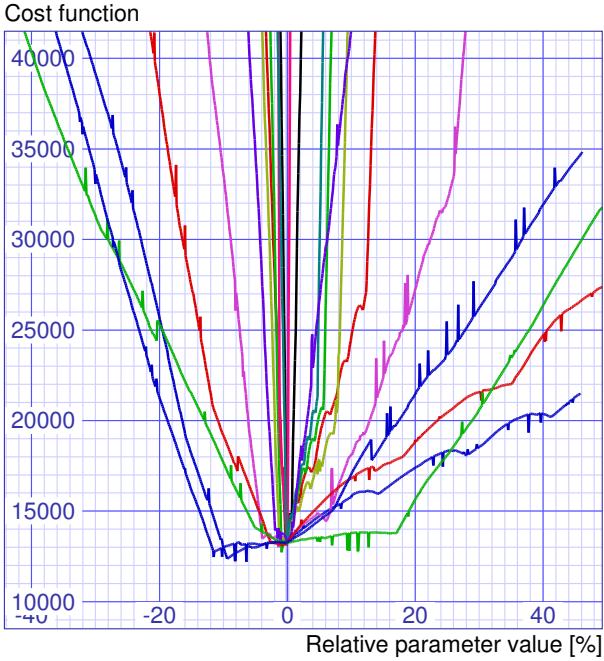


Fig. 2. Cost profile for the digital delay test problem. Each curve represents a cross section along one of the optimization parameters through the final point reached by the simplex algorithm. Zero represents the parameter value that corresponds to the final point.

is replaced by a local optimization algorithm with strong mathematical background (e.g. a trust-region algorithm) in the final part of the optimization.

This paper attempts to improve the performance of the Box simplex algorithm on real-world IC sizing problems by using a trust-region based algorithm in the final part of the search. In the following sections the Box simplex algorithm and the used trust-region algorithm are described, followed by the description of the combined algorithm. The main characteristics of the used test problems are given and the combined algorithm is compared to the original Box-simplex algorithm. Finally the results are discussed and the directions for future work are given.

II. THE BOX SIMPLEX ALGORITHM

The variant of the simplex algorithm that is used throughout this paper was developed by Box [1] and slightly modified [2]. It manipulates a set of $m \geq n+1$ points $x_i, i = 1, \dots, m$. Every point has a corresponding CF value $f_i = f(x_i)$. For simplicity we assume that the points are ordered according to their CF value ($f_1 \leq f_2 \leq \dots \leq f_m$). By mirroring and contracting the worst point x_m , first to the centroid of the remaining $m-1$ points ($x_* = (m-1)^{-1} \sum_{i=1}^{m-1} x_i$), and if that fails to the best point of the simplex, a hopefully better point is obtained which replaces x_m . The algorithm stops when the simplex size ($\Delta = 100/(m-1) \sum_{i=1}^{m-1} (\sum_{j=1}^n (\frac{x_i^j - x_*^j}{B^j - b^j})^2)^{1/2}$) falls below γ_s .

III. THE TRUST-REGION ALGORITHM

Trust-region algorithms [6] build a model of the CF ($m(x)$) and assume that it is valid over a set \mathcal{T} that contains the

current point x_c . \mathcal{T} is also referred to as the trust region. The model is minimized subject to $x \in \mathcal{T}$ resulting in point x_m . If $m(x_m)$ agrees well with $f(x_m)$ the trust region is expanded, otherwise it is contracted. If $f(x_m) < f(x_c)$ the current point x_c is replaced by x_m .

Since the design space has rectangular shape, the most natural form of the trust region is a hypercube. This can be achieved by defining \mathcal{T} using the l_∞ norm as $\{x : \|x - x_c\|_\infty \leq r\}$ where r is the trust-region radius.

The model is built by evaluating the CF at a set of points from \mathcal{T} and then interpolating them with $m(x)$. The most common model in the literature is quadratic. Unfortunately it requires $(n+1)(n+2)/2$ points to be evaluated. This number quickly grows beyond the number of CF evaluations required by the Box simplex algorithm. Therefore a quadratic model is not the best choice. The proposed algorithm uses a linear model that requires only $n+1$ CF evaluations.

In our implementation of the trust-region algorithm all of the optimization parameters are normalized to intervals $[0, 100]$. The algorithm is stopped when the trust-region radius r (expressed in normalized coordinates) falls below some predefined value γ_t .

IV. SWITCH CONDITION

The point at which the Box simplex algorithm is replaced by the proposed trust-region algorithm is determined by observing two quantities: the simplex size and the rate of simplex size change. The latter is obtained with regression from 2-tuples of the form (i, Δ_i) where i denotes the consecutive number of CF evaluation.

The switch condition is not checked until there are at least $j > m\mu_1$ 2-tuples in the sequence. Supposing that the algorithm is at j -th CF evaluation, the set of 2-tuples that satisfies $\max(j - m\mu_2, 0) \leq i \leq j$ is used in the regression. The obtained slope (k_j) is usually negative as the simplex is mostly shrinking and represents the rate of simplex size change.

The switch condition is defined as $-k_j < k_s n^{-1/2} \vee \Delta_j < k_d$ where k_s and k_d are user specified constants.

After the switch condition is satisfied the best point of the simplex becomes the current point of the trust-region algorithm. The initial trust-region radius is set to 100 so that the whole design space is covered by \mathcal{T} . Then the trust-region algorithm is started.

Since the switch condition is based on the simplex size (just as the Box algorithm's termination condition) and the design space is normalized to $[0, 100]$, we expect the switch condition to be applicable to a wide variety of CFs and design space sizes.

V. TEST PROBLEMS

The combined algorithm was tested on 3 mathematical functions and 8 real-world IC design problems. SPICE OPUS [7], [2] was used for evaluating the CF. The test problems are listed in table I. For every problem the number of optimization

parameters (n), the number of circuit analyses, and the number of MOS transistors are listed.

The CF was constructed according to [8] and is not the subject of this paper where we are interested only in the optimization algorithm itself. The performance of the algorithm can be assessed through the final CF value and the number of CF evaluations that are needed to reach this value. A fair comparison is assured by using the same termination condition for all the test problems.

More strict termination conditions mean more CF evaluations (which is not desired) and a lower final CF value (which is desired). If algorithm A outperforms algorithm B both in terms of the number of CF evaluations and final CF value, algorithm A can be considered better.

TABLE I

THE NUMBER OF OPTIMIZATION PARAMETERS, THE NUMBER OF ANALYSES IN ONE CF EVALUATION, AND THE NUMBER OF MOS TRANSISTORS IN THE CIRCUIT FOR DIFFERENT TEST PROBLEMS.

Test problem	n	# analyses	# MOS
lin	10	-	-
sq1	10	-	-
sq2	10	-	-
nand	3	9	4
delay	12	2	6
ddamp	14	44	18
damp1	15	31	13
damp2	27	4	20
damp2c	27	20	20
lfbucf	36	4	32
lfbufc	36	20	32

The first 3 problems are mathematical functions. The first one (lin) is a linear function of 10 variables. The minimum occurs in one of the corners of the design space. The second problem (sq1) is a convex quadratic function with minimum in the center of the design space. The sq2 problem is a concave quadratic function with minimum in one of the corners of the design space. On all of these simple functions the Box simplex algorithm reveals its weaknesses, most notably slow convergence when it is closing in on a corner of the design space.

The 8 problems that follow are real-world IC design problems. The nand problem attempts to size a digital nand gate. It has only 3 optimization parameters. The digital delay problem (delay) is somewhat larger with its 12 parameters. The circuit consists of 2 inverters and a MOS resistor.

The remaining 6 problems are various opamp sizing problems. The ddamp case has a moderate number of optimization parameters and a fairly large number of circuit analyses. The damp1 case is a very simple opamp with a minimal number of MOS transistors.

The damp2 case (see figure 3) is a fairly complex opamp. It is sized for a single combination of operating conditions and process variations (one corner point). The damp2c is the same topology sized for 5 different corner points simultaneously. This increases the amount of simulation needed to evaluate the CF and also makes the CF more challenging to optimize (local minima and noise).

The lfbucf and lfbufc cases represent an opamp that serves as a buffer for low frequency signals. It is the largest test circuit. The first version (lfbucf) is sized for a single corner point, and the second one (lfbufc) for 5 corner points.

VI. RESULTS

The proposed combination of algorithms was implemented in the SPICE OPUS circuit simulation and optimization tool. The test problems were optimized with the Box simplex algorithm (main optimization algorithm in SPICE OPUS) and with the proposed combined algorithm. The results are listed in table II. The following algorithm parameters and termination conditions were used: $m = 2n$, $\gamma_s = \gamma_t = 0.001$, $\mu_1 = 10$, $\mu_2 = 100$, $k_s = 0.22$, and $k_d = 1.5$.

Of the three mathematical functions the combined algorithm outperforms the Box simplex algorithm on the lin and sq2 test problems. For the sq1 test problem the combined algorithm ends with a worse final CF value. If a more strict termination condition is used ($\gamma_t = 1.4 \cdot 10^{-5}$), the combined algorithm takes 2120 CF evaluations to reach the final CF value of $1.65 \cdot 10^{-6}$ (which is only slightly worse than the result obtained by the Box simplex algorithm).

Of the real world IC design problems the nand case exhibits worst performance when the final CF value is considered. Nevertheless it is almost as good as the one obtained by the Box simplex algorithm. All of the remaining cases resulted in a better final CF value than the one obtained by the Box simplex algorithm, most notably ddamp, damp2, and lfbucf.

Regarding the number of CF evaluations all of the IC design test problems took less CF evaluations with the combined algorithm than with the Box simplex algorithm. The savings were highest with the largest cases and reached beyond 50% (damp2c and lfbucf).

Overall the combined algorithm outperformed the simplex algorithm on 7 out of 8 real world IC design problems.

TABLE II
THE COMPARISON OF THE NUMBER OF CF EVALUATIONS AND THE FINAL CF VALUE FOR THE SIMPLEX AND THE COMBINED OPTIMIZATION ALGORITHM.

	Simplex		Combined	
	N	f	N	f
lin	5066	-999.993	888	-1000.000
sq1	2130	1.44e-6	1028	3.70e-3
sq2	6871	-24999.1	363	-25000.0
nand	295	166.644	172	166.735
delay	1334	13284.6	1215	12975.1
ddamp	1543	104.426	1453	91.2954
damp1	1716	9.58849	1365	9.58514
damp2	5719	2.64345	3369	2.05055
damp2c	4732	7.36757	2314	7.02317
lfbucf	9152	0.763508	3786	0.707504
lfbufc	5759	4.18654	3028	4.01531

Table III compares the performance measures of the damp2c circuit obtained with the simplex algorithm and the combined algorithm. The damp2c test problem is optimized across 5 corners ranging temperatures from 0 to 50°, supply voltages from 1.6V to 2.0V, and 5 CMOS process corners. 20 MOS

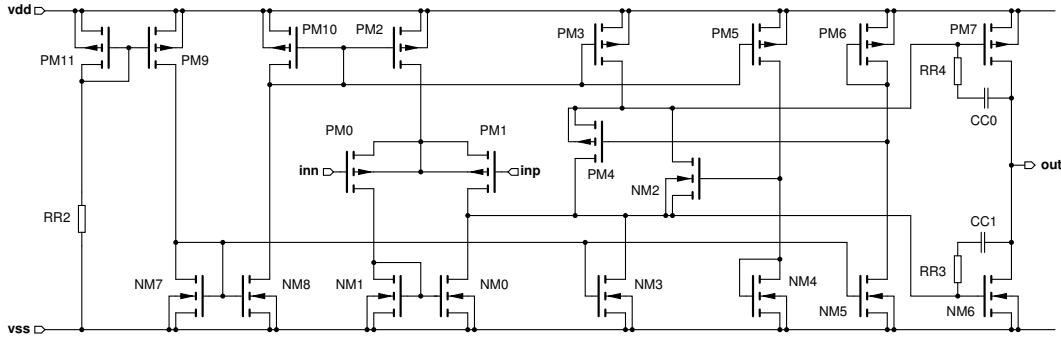


Fig. 3. The schematic of the damp2 test problem.

TABLE III

RESULTING CIRCUIT PERFORMANCE FOR THE DAMP2C TESTCASE.

Performance measure	Target	Simplex	Combined
Cost	min	7.37	7.02
Area (μm^2)	min	7310	7160
Supply current (mA)	min	0.992	0.984
AC gain (dB)	max	63.6	64.2
Unity gain bw. (MHz)	max	3.6	4.11
Phase margin ($^\circ$)	max	41.1	40.4
Gain margin (dB)	max	14.5	13.5
Swing (V)	max	0.893	0.896
DC Gain (dB)	max	43.3	43.4
Settling time (ns)	min	275	333
Overshoot (%)	min	0.765	0.857
Slew rate (V/ μs)	max	3.4	3.8
Rise time (ns)	min	141	126
Fall time (ns)	min	315	301

transistors, 2 capacitors, and 3 resistors are subject to optimization resulting in 27 optimization parameters.

Of the 13 performance measures, the circuit that resulted from the combined algorithm run has 9 performance measures that are better than for the circuit that resulted from the simplex algorithm run. The total number of candidate circuit evaluations was 51% lower and the final CF value was 5

VII. CONCLUSION

A combined optimization algorithm consisting of the Box simplex algorithm used in the first stage and a trust-region algorithm in the second stage of the search was presented. The reason for combining the two algorithms was the slow convergence of the Box simplex algorithm observed toward the end of the search. It was hypothesized that the combined algorithm can be faster and obtain a better final result than the original Box simplex algorithm due to the strong mathematical background of trust-region algorithms.

The simplex algorithm manipulates a set of points in the design space. This makes it noise resistant and gives it some global search capabilities. A measure for the diameter of the simplex is the simplex size. The point at which the algorithm switches to the trust region algorithm is determined based on the size of the simplex and the rate at which the simplex size is decreasing. A linear model was used in our trust-region algorithm to make it feasible for higher dimensional problems.

The combined algorithm was compared to the Box simplex algorithm on 3 mathematical functions that point out the weaknesses of the simplex algorithm and a set of real-world IC design problems. The combined algorithm outperformed the Box simplex algorithm in terms of the final CF value and the number of CF evaluations on most test problems (9 out of 11).

The use of a fast local search method in the second part of the optimization when the algorithm is already in the neighborhood of a good local minimum has proven to be most beneficial. In the future we intend to utilize a local search algorithm in combination with a global optimization method that is capable of finding the global minimum of the cost function.

ACKNOWLEDGEMENT

The research was co-funded by the Ministry of Education, Science, and Sport (Ministrstvo za Šolstvo, Znanost in Šport) of the Republic of Slovenia through the programme P2-0246 Algorithms and optimization methods in telecommunications.

REFERENCES

- [1] M. J. Box, "A new method of constrained optimisation and comparison with other methods," *Computer Journal*, vol. 7, pp. 42–52, Apr. 1965.
- [2] J. Puhan, T. Tuma, and I. Fajfar, "Optimisation methods in spice, a comparison," in *Proceedings, European Conference on Circuit Theory and Design*, vol. 2, Stresa, Italy, Sept. 1999, pp. 1279–1282.
- [3] V. T. T. G. Kolda, R. M. Lewis, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Review*, vol. 45, pp. 385–482, 2003.
- [4] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, pp. 112–147, 1998.
- [5] K. I. M. McKinnon, "Convergence of the Nelder-Mead simplex method to a nonstationary point," *SIAM Journal on Optimization*, vol. 9, pp. 148–158, 1999.
- [6] A. Conn, N. Gould, and P. Toint, *Trust-region methods*. Philadelphia, USA: SIAM, 2000.
- [7] J. Puhan and T. Tuma, "Optimization of analog circuits with spice 3f4," in *Proceedings of the ECCTD'97*, vol. 1, Budapest, Hungary, Sept. 1997, pp. 177–180.
- [8] A. Bürmen, D. Strle, F. Bratkovič, J. Puhan, I. Fajfar, and T. Tuma, "Automated robust design and optimization of integrated circuits by means of penalty functions," *International Journal of Electronics and Communications*, vol. 57, pp. 47–56, 2003.