A FAMILY OF MATROID INTERSECTION ALGORITHMS FOR THE COMPUTATION OF APPROXIMATED SYMBOLIC NETWORK FUNCTIONS

Piet Wambacq[*], Francisco V. Fernández[†], Georges Gielen[*‡], Willy Sansen[*] and Angel Rodríguez-Vázquez[†]

[*] Katholieke Universiteit Leuven, ESAT-MICAS, Kard. Mercierlaan 94,
B-3001 Heverlee, Belgium Tel.: Tel.: +32/16/321077 - Telefax: - +32/16/321975

[†] Centro Nacional de Microelectrónica, Avda. Reina Mercedes S/N
E-6515 Sevilla, Spain, Tel.: +34/5/423/9923 - Telefax: - +34/5/462/4506

[‡] research associate of the National Fund of Scientific Research, Belgium

Abstract

In recent years, the technique of simplification during generation has turned out to be very promising for the efficient computation of approximate symbolic network functions for large transistor circuits. In this paper it is shown how symbolic network functions can be simplified during their generation with any well-known symbolic network analysis method. The underlying algorithm for the different techniques is always a matroid intersection algorithm. It is shown that the most efficient technique is the two-graph method. An implementation of the simplification during generation technique with the two-graph method illustrates its benefits for the symbolic analysis of large analog circuits.

1 Introduction

During the last few years, many research efforts have been performed in order to symbolically analyze large analog circuits. The goal of such a symbolic network analysis is the generation of symbolic expressions that can be used either for interpretation or for repeated evaluations in design automation applications such as automated circuit synthesis, statistical analysis, ... The first modern symbolic analyzers such as ISAAC [1], ASAP [2] that have been developed for these purposes were dedicated to the analysis of linearized analog circuits. In this way, a circuit characteristic corresponds to a network function which is a rational function of the frequency variable s. In a symbolic network function in expanded format, every coefficient of any power of s is a sum of products of symbolic circuit parameters that correspond to the circuit elements of the linearized, small-signal circuit. Symbolic analyzers such as ISAAC and ASAP are able to generate approximate network functions. In this way it is possible to generate interpretable expressions rather than the lengthy exact expressions. The approximation is based upon the numerical values of the symbolic circuit parameters. The approximation in ISAAC and ASAP proceeds by first generating the exact network function in expanded format after which the non-dominant terms are discarded. The main problem of this approach is that the exact network function must be generated first. The number of terms of this network function increases exponentially with the size of the circuit. For example, the determinant of the admittance matrix of the commercial μA741 operational amplifier contains more than 10^{17} terms. Such a huge amount of terms cannot be generated completely. The final approximate result, on the other hand, usually contains very few terms.

The above approximation technique that is used in programs such as ISAAC and ASAP is denoted as a simplification *after* generation technique: the exact network function is generated first, after which it is approximated. In [3] an alternative approximation technique has been proposed that is denoted as a *simplification during generation* technique. With this technique, a symbolic network function is approximated while it is being generated. This is performed by generating only the dominant terms of the network function in decreasing order of magnitude, without skipping any term and until the sum of generated terms is sufficiently close to the exact numerical value.

The technique of simplification during generation has been implemented for example in [5, 6, 7, 8]. In all of these cases the simplification during generation technique has been realized with the two-graph method. However, several other symbolic network analysis methods can be used to realize the technique of simplification during generation technique. In this paper it is shown how this technique can be realized with the two-graph method [4], the directed-tree enumeration method [4], the Coates flow graph method [4], the Laplace expansion of a determinant [1] and the parameter extraction method of Sannuti and Puri [4]. For all these network analysis methods the problem of enumerating the dominant terms in decreasing order can be considered as a problem of enumerating in order of weight bases that are common to three matroids. Every such base corresponds to a term of the network function. This is explained in Section 2. In Section 3 it is shown that the two-graph method is more efficient than the other symbolic network analysis methods.

The problem of enumerating bases that are common to three matroids is NP-complete. Since an enumeration of bases that are common to *two* matroids can be performed in a time that is polynomially bounded, the enumeration procedure of bases common to three matroids is performed as follows: two of the three involved matroids are chosen, and their common bases are enumerated in order of weight. For every base found in this way it is checked whether it is a base in the third matroid as well. The efficiency of the overall procedure is highly influenced by the choice of the two matroids for which common bases are enumerated.

An implementation of the matroid intersection problem is discussed in Section 4. Since it is found that the two-graph method is more efficient than the other symbolic network analysis methods, the matroid intersection problem is solved

for the matroids that are involved for the application of the two-graph method. Finally, some experimental results are presented as well.

2 Simplification during generation in terms of matroids

A *matroid* $M = (S, \mathcal{J})$ is a structure in which S is a finite set of *elements* and $\mathcal{J}$ is a family of subsets of S, such that the following axioms, called *independence axioms*, are satisfied:

1. $\varnothing \in \mathcal{J}$
2. All proper subsets of a set I in $\mathcal{J}$ are in $\mathcal{J}$.
3. If I_p and I_{p+1} are sets in $\mathcal{J}$ containing p and $p + 1$ elements respectively, then there exists an element $e \in I_{p+1} - I_p$ such that $I_p + e \in \mathcal{J}$.

An independent set of maximum cardinality is called a *base*. The notion of independence is made concrete for a specific matroid. A *graphic matroid* is defined on the set of edges of a graph. Here, an independent set is defined as a set of edges that don't contain a cycle. It is clear that a base in this matroid corresponds to a spanning tree. A *partition matroid* is defined as follows. Let π be a partition which separates the finite set S into m disjoint blocks $B_1, B_2, \ldots, B_m$ and let d_i, $i = 1, 2, \ldots m$ be m given nonnegative integers. Then for any S, π and d_i $(i = 1, 2, \ldots, m)$ $M = (S, \mathcal{J})$ is a matroid, in which $\mathcal{J} = \{I \mid I \subseteq S, |I \cap B_i| \leq d_i, i = 1, 2, \ldots, m\}$. Any matroid with such a structure is called a *partition matroid*. Usually, it is assumed that each of the numbers d_i, $i = 1, 2, \ldots, m$, is equal to one. If any d_i deviates from one, then this will be mentioned explicitly.

It is possible to define more than one matroid on a set S. Also, it is possible that a subset of S is independent for more than one matroid. Such subset is denoted as a matroid intersection. When the cardinality of a base is the same for more than one matroid defined on a set S then a matroid intersection of maximum cardinality is a *base that is common to more than one matroid*.

For different symbolic network analysis methods the problem of simplification during generation can be formulated as the enumeration of bases that are common to three matroids. This is explained now for the two-graph method, the directed-tree enumeration method, the Coates flow graph method and the parameter extraction method of Sannuti and Puri. It can be proven that with the Coates flow graph method exactly the same terms are generated as with the Laplace expansion of a determinant in which all cancelling terms are elaborated only after a complete expansion. Hence, the considerations for the Coates flow graph method also apply to the Laplace expansion method.

Two-graph method Given two weighted graphs, the voltage and the current graph of the circuit, that contain the same n vertices and m edges, but their topology is different. Each edge has a weight and a color, red or green. The red edges correspond to conductances or transconductances, the green edges to capacitors. The problem of simplification during generation with the two-graph method is as follows: *Enumerate in decreasing order of weight spanning trees that are common*

to both the voltage and the current graph and that contain k green edges and $n - 1 - k$ red edges. In this way, terms of the kth power of s in the numerator or denominator of the network function are enumerated. This problem is clearly an intersection problem in which three matroids are involved: the graphic matroid on the voltage graph, the graphic matroid on the current graph and the partition matroid on the edges induced by the coloring. A polynomial-time algorithm that exploits the special structure of the involved matroids has not been found.

Directed-tree enumeration The directed-tree enumeration method finds terms of a network function by enumerating directed trees in a directed graph [4]. Let $G = (V, E)$ be a weighted directed graph. Suppose that a spanning tree directed from a pre-described root vertex with in-degree zero is to be found. Any subset of edges I forming such a spanning tree must satisfy two conditions. First, it must contain no cycle. Hence I must be an independent set of the graphic matroid of G in which the directions of the edges are ignored. Secondly, it must contain no more than one edge into any given vertex. Hence, I must be an independent set of the partition matroid which is defined by a partition of the edges which places all the edges directed into a given vertex in the same block. A directed spanning tree exists if and only if there is an $(n - 1)$-element intersection of these two matroids (n is the number of vertices in G). Assume that the directed-tree enumeration method is used for the realization of the technique of simplification during generation. Then it is necessary to generate the dominant terms of every coefficient of every power of s in the network function. This corresponds to the enumeration of directed spanning trees with exactly k green edges corresponding to capacitors, with k being the power of s the dominant terms of which are being generated. From the above considerations it is clear that the directed-tree enumeration method that takes into account the color constraint corresponds to an intersection problem of three matroids. A polynomial-time algorithm that exploits the special structure of the involved matroids has not been found.

Coates flow graph method Consider a (weighted) Coates flow graph in which the directed edges have a color, red or green. The problem of simplification during generation for the Coates flow graph method is as follows: *Determine the largest 0-connection that contains exactly k green edges.*

A subset I of edges of the Coates flow graph forms a 0-connection if the in-degree as well as the out-degree of every vertex is exactly one. Hence I must be an independent set of two partition matroids: the first is defined by a partition of the edges which places all the edges directed *into* a given vertex in the same block, the other one by a partition of the edges which places all the edges directed *from* a given vertex in the same block. In order to satisfy the color constraint, the 0-connection must also be an independent set in a third matroid which is induced by the coloring of the elements of the Coates flow graph either red or green. A polynomial-time algorithm that exploits the special structure of the involved matroids has not been found.

807

Parameter-extraction method of Sannuti and Puri Given p sets $B_1, B_2, \ldots, B_p$ that each contain m_i elements ($i = 1, \ldots, p$). Each element has a weight and a color, red or green. Determine the combination of elements with the largest weight[1] that contains k_i elements from set B_i ($k_i \leq m_i$ and $i = 1, 2, \ldots, p$), such that the combination contains a total of n elements ($n = \sum_{i=1}^{p} k_i$). From these n elements exactly k elements are green. This problem is a subproblem that arises when the parameter-extraction method of Sannuti and Puri is used to find the largest term of the kth power of s in a network function. This problem can be considered as the determination of the largest intersection of two partition matroids.

However, in the above formulation, the extra conditions that can be imposed to weed out more symbol combinations than with a mere dimensional check, have been omitted. The incorporation of these constraints in the problem formulated above yields a complex problem that can be considered as a problem that consists of p matroid intersection problems, with p growing exponentially with the number of symbols.

3 Comparison of symbolic network analysis methods

All the above formulations of the simplification during generation problem correspond to problems that cannot be solved in a polynomially bounded time. The problem of enumerating bases that are common to three matroids is solved by enumerating bases common to two matroids, after which each such base is tested for independence in the third matroid. The enumeration of bases that are common to two matroids in decreasing order of weight can be performed with the algorithm of Camerini and Hamacher described in [10]. The running time limit of this algorithm is $O(KmRc(m))$ in which K is the number of enumerated bases, m is the number of elements in the matroid, R is the cardinality of the bases, also denoted as the *rank* of the matroid and $c(m)$ is the maximum of the running time limits of the independence tests in each of the two matroids. The total running time limit of this approach can be improved by exploiting the special structure of the involved matroids.

The efficiency of the simplification during generation approach not only depends on the efficiency of the enumeration algorithm, but it also depends on the number of terms that has to be generated. A symbolic network analysis method can generate cancelling or invalid terms. The less cancelling or invalid terms are generated, the more advantageous a symbolic network analysis method is. In [9] it is found that the two-graph method produces the least number of cancelling terms. With the two-graph method, cancelling terms are only present when matching elements are considered, i.e. different circuit elements that are represented by the same symbol. The number of terms that would be generated with every symbolic network analysis method can be predicted with formulas evolving from graph theory or combinatorics [9]. These formulas are applied to the determinant of the admittance matrix of the fully-differential BiCMOS amplifier with twenty transistors, which is depicted in Figure 1.
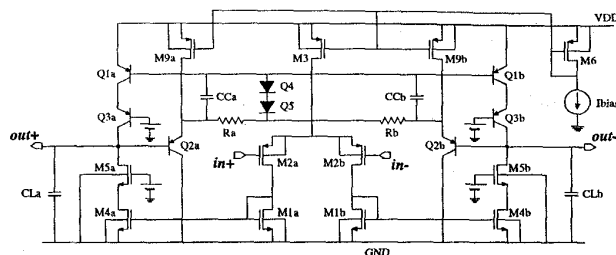

Figure 1. A fully differential BiCMOS OTA.

The number of terms that would be generated with the different symbolic network analysis methods is listed in Table 1.

lower bound # common spanning trees	6.57×10^{10}
# spanning trees in voltage graph	4.51
# spanning trees in current graph	9.91
# terms with directed-tree enum.	53.7
# terms with Coates flow graph	23.4
# terms with Sannuti & Puri	1.15×10^8

Table 1. Number of terms that would be generated with the different symbolic network analysis methods for the computation of the determinant of the admittance matrix of the BiCMOS amplifier from Figure 1. The numbers below the first row have been normalized to the number of the first row.

The numbers of terms for the different methods are referred to the lower bound on the number of spanning trees that are common to the voltage graph and the current graph that are set up for the computation of the determinant of the admittance matrix. This is also a lower bound on the number of terms in the expanded determinant after all cancelling terms have been elaborated. It is seen that the two-graph method is superior to the other symbolic network analysis methods. For each choice of the two matroids for which common bases are generated, there will always be redundant terms, either invalid terms or valid terms that are generated more than once. Assume that common spanning trees are generated by first generating spanning trees in one graph after which it is examined whether this set of edges also corresponds to a spanning tree in the other graph. Then it is seen in the table that still less spanning trees of the voltage or current graph would be considered than the number of terms that would be generated by any algorithm that would be able to directly generate the terms that evolve from the application of any other symbolic network analysis method.

Further it is seen that the number of spanning trees in the voltage graph is lower than the number of spanning trees in the current graph.

The conclusions that have been drawn from the numbers for the example of Figure 1 also apply to other circuits of practical size [9]. Hence it can be concluded that the two-graph method is the most efficient method from the point of

---

[1]The weight of a combination is equal to the sum of the weights of the elements in the combination.

view of generating the least number of cancelling or invalid terms.

4 Specialization to specific matroids

The technique of simplification during generation has been implemented for the two-graph method. The choice of this method is justified by the considerations of Section 3. For the two-graph method the involved matroids are the graphic matroid on the voltage graph, the graphic matroid on the current graph and the partition matroid on the edges of the voltage or current graph that is induced by the coloring of the edges. For the enumeration of valid terms in order, bases that are common to two matroids are first enumerated after which for each base it is checked whether it also constitutes a base in the third matroid. In our implementation the common bases of the graphic matroid on the voltage graph and the partition matroid are enumerated. The generation of bases that are common to the graphic matroid on the current graph and the partition matroid is less efficient since in general the current graph contains more spanning trees than the voltage graph. This is also illustrated in Table 1. The approach of generating common spanning trees without taking into account the color constraint is used in [8]. With this approach common spanning trees are generated at different frequencies. In this way, the absolute value of the weight of the capacitive edges equals $\omega_i C$, in which ω_i is the frequency under consideration and C is the capacitance value. This value can be compared to the weight of an edge that corresponds to a resistor or a voltage-controlled current source. The disadvantage of this approach is that errors in the approximate transfer function may occur when the chosen frequency grid is too coarse. On the other hand, many terms are generated much more than once when the frequency grid is very fine. The latter case might be necessary for example beyond the gain-bandwidth of an amplifier where many poles can occur close to each other.

For the option that is chosen in our implementation, namely the enumeration of spanning trees in the voltage graph that satisfy the color constraint, the running time limit of the algorithm of Camerini and Hamacher, $O(KmRc(m))$ [10] reduces to $O(Kmn^2)$, in which K is the number of enumerated spanning trees, m is the number of edges in the voltage graph and n is the number of vertices in the voltage graph. However, by exploiting specific features of the two involved matroids the running time limit of this procedure can be lowered to $O(Km)$ [9]. This limit is lower than the running time limit of the algorithm described in [7] that solves the same problem.

The matroid intersection algorithm mentioned above is the kernel of the symbolic analyzer ADAGIO that has been implemented in C. A symbolic analysis of the BiCMOS amplifier from Figure 1 with a relative error of 20% takes 1149 seconds on a SUN Sparc station 2. The CPU time can of course be lowered by simplifying the voltage and the current graph before the enumeration procedure starts, as explained in [7] but care must be taken not to introduce inaccuracies.

5 Conclusions

The technique of simplification during generation for the generation of approximate network functions is generally accepted as a major breakthrough for the symbolic analysis of large linearized analog circuits. In this paper it has been shown how this technique can be implemented with the different existing symbolic network analysis methods using matroid intersection algorithms. It has been shown that the two-graph method is the most suitable method for the implementation of the technique of simplification during generation. Experimental results illustrate that a symbolic analysis of large analog circuits with twenty transistors or more is feasible in a reasonable CPU time.

References

[1] G. Gielen, H. Walscharts, and W. Sansen, "ISAAC: a symbolic simulator for analog integrated circuits," *IEEE J. Solid-State Circuits*, Vol. 24, No. 6, pp. 1587–1597, December 1989.

[2] F. Fernández, A. Rodríguez-Vázquez and J. Huertas, "Interactive ac modeling and characterization of analog circuits via symbolic analysis," *Kluwer J. Analog Integrated Circuits and Signal Processing*, Vol. 1, pp. 183–208, November 1991.

[3] P. Wambacq, G. Gielen, and W. Sansen, "A cancellation-free algorithm for the symbolic analysis of large analog circuits," *Proc. ISCAS*, 1992, pp. 1157–1160.

[4] P. Lin, *Symbolic Network Analysis*. Elsevier, 1991.

[5] P. Wambacq, F. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Algorithm for efficient symbolic analysis of large analogue circuits," *IEE Electronics Letters*, Vol. 30, No. 14, pp. 1108–1109, July 1994.

[6] P. Wambacq, F. Fernández, G. Gielen, W. Sansen, and A. Rodríguez-Vázquez, "Efficient symbolic computation of approximated small-signal characteristics," *IEEE J. Solid-State Circuits*, Vol. 30, No. 3, pp. 327–330, March 1995.

[7] Q. Yu and C. Sechen, "Approximate symbolic analysis of large analog integrated circuits," *Proc. IEEE/ACM IC-CAD*, 1994, pp. 664–671.

[8] Q. Yu and C. Sechen, "Efficient approximation of symbolic network functions using matroid intersection algorithms," *Proc. ISCAS*, pp. 2088–2091, 1995.

[9] P. Wambacq, "Symbolic analysis of large and weakly nonlinear analog integrated circuits," Ph.D. Dissertation Katholieke Universiteit Leuven, 1996.

[10] P. Camerini and H. Hamacher, "Intersection of two matroids: (condensed) border graphs and ranking," *SIAM J. Disc. Math.*, Vol. 2, No. 1, pp. 16–27, February 1989.