

# An Optimized Parallel Admittance Matrix Approach Using the Adjacence-Graph Recursive-Thresholding Technique

Alessandra Esposito, Federico Malucelli, and Luciano Tarricone

**Abstract**—The admittance method is an accurate approach for the analysis of electromagnetic circuits. Unfortunately, until now it has suffered from two main limitations, i.e., its high numerical complexity and its lack of robustness, due to the risk of numerical ill conditioning in a linear system representing the core of the approach. In a previous paper, both drawbacks have been solved, using a strategy based on the system partitioning into many independent and well-conditioned reduced-size subsystems, thanks to the exploitation of the matrix adjacence graph properties. In this paper, we demonstrate that the use of this strategy paves the way to a natural, straightforward, and low-cost migration on distributed platforms, with a consequent substantial reduction in computer times. Furthermore, the use of suitable optimization strategies proposed here allows an optimum partitioning of the system in order to maximize the parallel efficiency.

**Index Terms**—Admittance matrix, mode matching, parallel computing, optimization.

## I. INTRODUCTION

THE mode-matching (MM) technique is one of the most attractive frequency-domain approaches for the analysis of microwave (MW) circuits because of its efficiency and accuracy [1]–[3]. A more recent formulation, using the generalized admittance matrix (GAM) has been proposed [4], based on the interconnection of parallel epipedal elementary cells. This approach, useful for both its opening to enhanced optimization techniques [5], [6] as well as for its flexibility, suffers from the risk of ill conditioning of the system matrix.

Very recently, this problem has been efficiently circumvented by using the so-called adjacence-graph recursive-thresholding (AG–RT) technique [7]. The AG–RT technique, by exploiting some interesting properties of the system representation via adjacence graphs, and recursive operations of thresholding on the matrix entries, decomposes the system into many independent subproblems. Each subsystem can, therefore, be considered separately, and it is observed that its condition number is substantially improved with respect to the original admittance linear system. The original sparse system is, therefore, rearranged into a block-diagonal system, whose blocks represent small-size and well-conditioned systems.

Manuscript received June 26, 2001.

A. Esposito and L. Tarricone are with the Department of Innovation Engineering, University of Lecce, 73100 Lecce, Italy (e-mail: tarricone@diei.unipg.it).

F. Malucelli is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milan, Italy.

Publisher Item Identifier 10.1109/TMTT.2002.802322.

One of the attractive features is that the attained subsystems, thanks to their independency, can be solved all simultaneously. In this paper, this feature is completely exploited. By using suitable optimization strategies, as well as advanced parallel computing techniques, the problem of solving all the subsystems concurrently in the minimum computing time for a given computing system is solved.

Assuming a generic heterogeneous distributed platform, an optimum assignment policy is found so that the load-balancing among processors is maximized, and computing costs are reduced as much as possible.

The whole strategy is tested on a distributed-memory parallel platform (IBM SP2), but can be completely implemented on very low-cost computing environments, i.e., PC clusters, such as the Beowulf<sup>1</sup> and using freeware software available through the Web, such as the Message Passing Interface (MPI).<sup>2</sup> Consequently, the proposed strategy in conjunction with the AG–RT method represents the most effective and low-cost way to achieve high-performance computing when using the GAM approach.

The paper is structured as follows. First, we shortly recall the GAM method and its numerical properties, as well as the AG–RT strategy [7]. Later on, we propose a policy for an optimum parallel migration, based on an optimization formulation. Finally, some results are given, demonstrating the effectiveness of the parallel AG–RT GAM solution, as well as its amenability to a large class of distributed-memory environments.

## II. GAM FORMULATION AND THE AG–RT STRATEGY

### A. Numerical Characteristics of the GAM Approach

The GAM method has been described in several papers [4], [8]–[10] and we just recall here very basic considerations.

The domain is generally partitioned into atomic elements with a simple geometry so that their admittance matrix can be evaluated by using the approach of metallic resonators and circuit theory [11]. By using suitable field expansions over the resonator apertures, and by indicating with  $\underline{\underline{G}}^A(\mathbf{r}, \mathbf{r}')$  the dyadic electric Green's function, we can evaluate the generic entry in the admittance matrix, as referred to the  $i$ th and  $j$ th apertures, and the  $k$ th and  $m$ th ports

$$Y_{mn,kl}^{ji} = \int_{S_j} \int_{S_i} \mathbf{h}_{mn}(\mathbf{r}) \cdot \underline{\underline{G}}^A(\mathbf{r}, \mathbf{r}') \cdot \mathbf{e}_{kl}(\mathbf{r}') d\mathbf{r}' d\mathbf{r}. \quad (1)$$

<sup>1</sup>[Online]. Available: <http://www.beowulf.org>

<sup>2</sup>[Online]. Available: <http://www.mpi-forum.org>

The  $Y$  matrix dimension is  $\sum_{i=1}^N N_i$  ( $N_i$  being the number of modes retained to represent the field on the  $i$ th aperture, and  $N$  being the number of apertures).

Consequently, the GAM approach has its bottleneck in the solution of a linear system

$$\mathbf{I} = \mathbf{Y}_G \mathbf{V} \quad (2)$$

where  $\mathbf{V}$  is the vector of unknown voltages inside the circuit and at its output ports, while  $\mathbf{I}$  is the vector of excited currents inside the circuit and over its output ports. The two main issues for improving the numerical performance are: 1) avoid risks of  $Y_G$  ill conditioning and 2) reduce the system solution time. Both goals have been achieved by using the AG-RT strategy, which is now revisited for the reader's ease.

### B. AG-RT Strategy

The AG-RT is one of several strategies based on the simple idea of reducing the order of the problem by decomposing the system matrix into suitable fragments or the solution space into smaller subspaces [12]–[18]. The AG-RT approach can best be resumed by the following simple example, which is given in a more expanded form in [7].

We suppose that the system matrix is, for instance,

$$A = \begin{pmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x \\ 0 & x & x & x & 0 & 0 & 10^{-8} & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & x & x & 0 & 0 & 10^{-8} & 0 \\ 0 & 0 & 0 & x & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 10^{-7} & 0 \\ 0 & 10^{-8} & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 10^{-8} & 0 & 10^{-7} & x & x & x \\ x & 0 & 0 & 0 & 0 & 0 & x & x & x \end{pmatrix} \quad (3)$$

where entries larger than  $10^{-7}$  are represented with  $x$ . If we perform a first thresholding, neglecting values smaller than  $v_t = 10^{-8}$ , we are not able to identify any independent subproblems, as the resulting system matrix is

$$\hat{A} = \begin{pmatrix} x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x \\ 0 & x & x & x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & x & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & x & x & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x & x & 0 & 10^{-7} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 10^{-7} & x & x & x \\ x & 0 & 0 & 0 & 0 & 0 & x & x & x \end{pmatrix}. \quad (4)$$

We now introduce the concept of the *adjacence graph*. Each row/column of the matrix is numbered and represented by a node identified with the corresponding number. An arc connects nodes  $i$  and  $j$  if and only if the entry  $\hat{a}_{ij}$  in the matrix is not a zero. Referring to the current example, the attained graph is potentially partitionable into two subgraphs, provided that the entry  $\hat{a}_{68}$  (or equivalently  $\hat{a}_{86}$ ) is negligible. This is the case, provided that a new thresholding action is performed, with  $v_t = 10^{-7}$ . Now, we actually have identified two independent

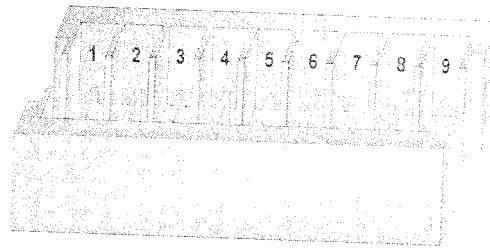


Fig. 1.  $E/H$ -plane filter analyzed by applying the AG-RT GAM strategy.

subproblems, corresponding to rows/columns 1,7–9 and 2–6. It is also theoretically proven and empirically verified that the recursive thresholding improves the condition number [19].

### III. PARALLELIZATION OF AG-RT GAM AS AN OPTIMIZATION PROBLEM

The description of the optimum policy for a parallel solution is now given. In the following, for the sake of clarity, we refer to a real example, i.e., the circuit in Fig. 1, with a system matrix  $\mathbf{Y}$  of size 1308. The  $\mathbf{Y}$  matrix is 18% sparse, having 30386 nonzero entries. Its condition number is very bad, thus, even very sophisticated system solvers are not able to find the system solution.

The application of the AG-RT strategy, with a final threshold  $v_t$  of  $10^{-6}$ , results in the system decomposition into 38 independent subsystems (generally called *connected components* (CCs)). One of them has dimension 967, and all the others are smaller than 90. Apart from the largest CCs, all the other subsystems have quite a good condition number, with a determinant larger than  $10^{-4}$  and a condition number smaller than  $10^3$ .

On the contrary, the largest subsystem is still ill conditioned, but this problem can be avoided by performing one additional recursion of the AG-RT strategy. With a final  $v_t$  of  $10^{-5}$ , the submatrix of size 967 can be partitioned into a large number of CCs (over 500). The condition number of nearly all the subsystems is quite good (smaller than  $10^5$ ). This finally allows the complete solution of the problem. The error when computing the scattering parameters, due to the thresholding action with  $v_t = 10^{-5}$ , is 1.1%, and therefore, is comparable with respect to typical manufacturing or experimental tolerances. The error is referred to the results attained by using the complete admittance matrix, when no zeroing is performed.

The starting matrix of size 1308 is finally partitioned into 579 subsystems, ranging from one system of size 174 to many others of smaller size down to size 1. All these subsystems are independent, and they can be solved concurrently. Therefore, the analysis of the complex circuit of Fig. 1 and the solution of the relative linear system can be performed in a distributed or parallel environment, and different strategies can be followed when distributing the partitions and tasks through the processors. For instance, a maximum load-balance policy or a minimum wall-clock (*makespan*) strategy, just to make some examples, can be pursued (with the consequent different choices for several parameters). They are linked to one another, thus influencing the numerical characteristics of the problems, i.e., value of  $v_t$ , error affecting the solution, number of CCs, and numerical complexity of the possible different linear system

solvers to be used on each subsystem (iterative sparse or direct banded). Other important issues are also the computing environment (shared versus distributed memory, processor interconnectivity, network performance, etc.) and the programming paradigm [(single program multiple data (SPMD) versus, for instance, master/slave (MS)].

The problem to be solved, revisited in a very simple way, is as follows: *for a given admittance matrix, and considering both the number of processors available, and the performance of each processor, which is the optimum  $v_t$  value, the best number of AG-RT iterations, and how must several CCs be distributed among the available processors, in order to minimize the computing time?*

Due to the high complexity of the problem, it is useful to attack and solve it in a general fashion by using the typical optimization formulations and techniques. This is performed and described in the following section.

#### A. Optimal Thresholding

Consider an  $N \times N$  matrix  $\mathbf{Y}$  with elements  $y_{ij}$  having values in  $[-1, 1]$  and let  $v_t \in [0, 1]$  be a given *threshold* parameter. Let  $\mathbf{Y}(\mathbf{v}_t)$  be the filtered matrix obtained as follows:

$$y_{ij}(v_t) = \begin{cases} y_{ij}, & \text{if } |y_{ij}| > v_t \\ 0, & \text{otherwise} \end{cases}, \quad i, j = 1, \dots, n. \quad (5)$$

For any given threshold  $v_t$  and by suitably reordering rows and columns,  $\mathbf{Y}(\mathbf{v}_t)$  reduces to a block diagonal matrix where, in practice, each block identifies an independent system of the form (2). In principle, setting  $v_t = 1$  reduces  $\mathbf{Y}(\mathbf{v}_t)$  to the identical matrix. Let us call  $\mathbf{Y}^1(\mathbf{v}_t), \mathbf{Y}^2(\mathbf{v}_t), \dots, \mathbf{Y}^{p(v_t)}(\mathbf{v}_t)$  the  $p(v_t)$  submatrices yielded by a threshold  $v_t$ , corresponding to the CCs of the related adjacency graph  $AG(v_t)$ .

In a distributed computing environment with  $P$  heterogenous processors, we can give an estimate of the execution time needed to solve each subsystem

$$\mathbf{Y}^i(\mathbf{v}_t) \mathbf{V}^i = \mathbf{I}, \quad i = 1, \dots, p(v_t) \quad (6)$$

with the most efficient method depending on the characteristics of the matrix. Let  $e_{ih}$  be the execution time estimation for solving system  $i$  on processor  $h$ ,  $i = 1, \dots, p(v_t)$ ,  $h = 1, \dots, P$ .

Once the decomposition is obtained and the execution times are evaluated, the crucial point is to assign the jobs (i.e., the computation of system solutions) to the processors so that the load is as balanced as possible, or the completion time of the last job is minimized (minimum makespan). We can assume that the submatrices are available in all the processors so that the communication times can be disregarded (as proven in Section IV). The evaluation of the execution times depends on the number and size of the submatrices.

The problem can be mathematically formulated as follows:

$$z = \min d$$

so that

$$\begin{aligned} \sum_{j=1}^m x_{ij} &= 1, \quad i = 1, \dots, p(v_t) \\ d &\geq \sum_{i=1}^{p(v_t)} e_{ij} x_{ij}, \quad j = 1, \dots, m \\ x_{ij} &\in \{0, 1\}, \quad i = 1, \dots, p(v_t); \quad j = 1, \dots, P \end{aligned} \quad (7)$$

where the variable  $x_{ij} = 1$  if system  $i$  is assigned to processor  $j$ , and 0 otherwise. The variable  $d$  gives the makespan.

The minimization of the makespan is a hard combinatorial optimization problem, indeed the known subset sum problem (i.e., the problem of partitioning a set of weighted elements into two subsets of equal total weight [20]) is a particular case where only two identical processors are available.

Since there is the need for assigning jobs to processors in a short time (otherwise the load balancing would be pointless), the problem must be coped with a heuristic algorithm. The reader can refer to [22] for a survey on heuristic algorithms and polynomial approximation schemes for this kind of problems. For the sake of simplicity, we adopt the heuristic known as longest processing time (LPT) assuming that all processors are identical and preemption is not allowed. However, any other, even more sophisticated, heuristic can be used instead. This heuristic guarantees a makespan not larger than twice the optimal one. In this heuristic the current LPT is assigned to the first idle processor.

Let us assume that the estimation of the makespan evaluated when the threshold is fixed to  $v_t$  is  $z(v_t)$ . Note that  $z(v_t)$  monotonically decreases when  $v_t$  increases. Now we address the problem of finding the smallest threshold value  $v_t^*$ , which yields a makespan not larger than a given desired value  $z^*$ . As each evaluation of  $z(v_t)$  can be quite time consuming, the key point is to determine  $v_t^*$  with the minimum number of guesses.

We now describe an efficient method to keep the CC information updated when the parameter  $v_t$  changes. Note that we can restrict ourselves to consider the actual  $y_{ij}$  values only, as possible threshold values, thus limiting the search of the  $v_t^*$  to the  $m$  nonzero entries of matrix  $\mathbf{Y}$ . Starting with  $v_t = 1$ , we have  $AG(v_t)$  defined by  $N$  singleton nodes, i.e.,  $N$  CCs of size 1. When  $v_t$  decreases, some edges are added to the adjacency graph, corresponding to the nonzero elements greater than  $v_t$ . In this case, some CCs may melt in a single component. We can efficiently keep track of the CCs evolution when  $v_t$  decreases by means of a suitable data structure [23]. At the beginning, when  $v_t = 1$ , we have  $N$  disjoint singleton sets. Each time an edge  $(i, j)$  is added to the graph, we perform the operations  $\text{FIND}(i)$  and  $\text{FIND}(j)$ , which give the sets  $i$  and  $j$  to which they belong, respectively. When  $i$  and  $j$  belong to different CCs (i.e.,  $\text{FIND}(i) \neq \text{FIND}(j)$ ), the addition of edge  $(i, j)$  causes the reunion of the two components in a single one; thus, we perform the operation  $\text{UNION}(i, j)$ , which updates the data structures so that the elements of the two components will be recognized as elements of the same new component. The implementation of these operations is quite efficient and simple: the amortized complexity of performing  $O(m)$  FIND-UNION operations is almost linear [23].

As a consequence, we can perform a binary search over the set of nonzero values of matrix  $\mathbf{Y}$  to determine  $v_t^*$ . Each time we make a new guess of the threshold, we update the adjacency graph starting from the smallest graph we have stored, adding the new edges and updating the CCs. The execution times  $e_{ih}$  are then evaluated and the heuristic to determine the makespan is applied. If the makespan is larger than the desired one, we have to increase the current threshold; on the contrary, if the makespan is larger than the desired one, we have to decrease it. The binary search algorithm is summarized as follows.

```

Procedure bin-search;
begin
  let  $A[i]$  be the array of  $\mathbf{Y}$  nonzero entries sorted in non increasing order,
   $i = 1, \dots, m$ ;
   $v_t' = 1$ ;  $i' = 1$ ;
  initialize the FIND-UNION data structure
  for graph  $AG' = AG(1)$ ;
   $v_t'' = 0$ ;  $i'' = m$ ;
  initialize the FIND-UNION data structure
  for graph  $AG'' = AG(0)$ ;
  repeat
     $i^* = \lfloor (i' + i'')/2 \rfloor$ ;
     $v_t^* = A[i^*]$ ;  $AG^* = AG'$ ;
    for each  $(i, j) : v_t^* \leq y_{ij} \leq v_t'$ ;
    begin
      add edge  $(i, j)$  to  $AG^*$ ;
      if FIND( $i$ )  $\neq$  FIND( $j$ ) then UNION( $i, j$ );
    end;
    evaluate  $e_{ch}$  for each CC  $c$  of  $AG^*$ ;
     $z = \text{MIN} - \text{MAKESPAN}(e_{ch})$ ;
    if  $z \leq z^*$ 
    begin
       $AG' = AG^*$ ;  $i' = i^*$ ;  $v_t' = v_t^*$ ;
    end;
    else begin  $i'' = i^*$ ;  $v_t'' = v_t^*$ ; end;
  until  $i'' - i' \leq 1$ 
end;

```

The overall complexity of the algorithm is  $O(m \log N \alpha(m) M)$ , where  $M$  is the complexity of the algorithm computing the minimum makespan.

#### IV. RESULTS

The accuracy of the AG-RT GAM implementation is first of all tested on two different circuits, as reported in Figs. 2 and 3. As is noticeable from these figures, the accuracy is satisfactory.

The results for the parallel implementation of the AG-RT GAM, on the contrary, are discussed referring to the above mentioned example of Fig. 1, whose main characteristics have already been described. For a reliable analysis of the parallel performance of the strategy, a discussion must be presented on its serial performance, which is proposed in Section IV-A.

##### A. Serial Results

As stated before, the admittance matrix for the circuit of Fig. 1 is partitioned into 579 subsystems, ranging from one system of

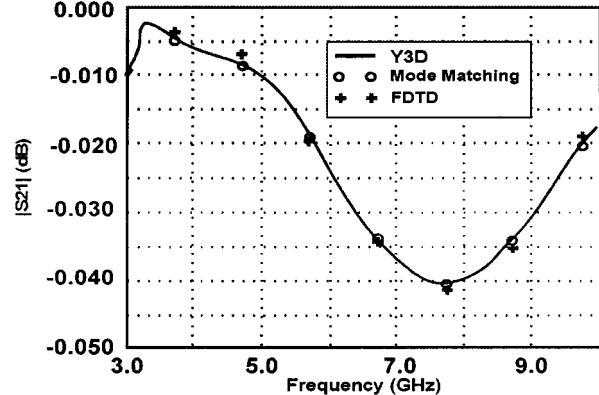


Fig. 2.  $E/H$ -step cascade analyzed by applying the AG-RT GAM approach (Y3D), compared with a generalized scattering matrix simulation (MM) and a licensed finite-difference time-domain software (XFDTD).

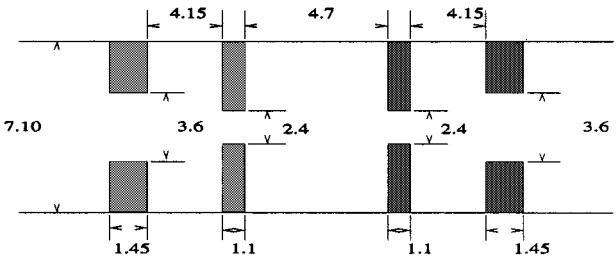
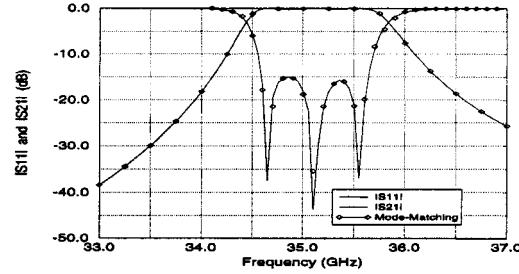


Fig. 3. Filter analyzed by applying the AG-RT GAM approach, compared with experimental data from [24].

size 174 to many others of size 1. The original matrix has a bad condition number so that the analysis could not be performed. The proposed approach substantially improves the numerical stability of the method so that the several linear subsystems can all be solved by using very efficient banded solvers.

Moreover, the analysis of the circuit and the solution of the relative linear system can be performed with a significantly reduced computational complexity. In order to benchmark this, we have generated a 1308 sparse matrix with the same pattern of the original matrix and an improved condition number so that the sparse solver could be used. Afterwards, we have compared the computing times using the standard sparse solver (SP) versus the

TABLE I  
SERIAL COMPUTING TIMES (IN SECONDS) AS REFERRED TO AN IBM RS6000 390. DATA REFER TO BOTH THE USE OF A SPARSE ITERATIVE SOLVER AND TO THE USE OF THE PROPOSED AG-RT STRATEGY IN CONJUNCTION WITH A BANDED SOLVER. BOTH SOLVERS ARE TAKEN FROM SLATEC LIBRARY

Strategy	Computing Time (s)
SP	2400
AG-RT	145+21

AG-RT strategy. In Table I, results are given on an IBM RS6000 390. The AG-RT strategy computing time is composed of the time needed to solve all the subsystems (145 s) plus the time needed to fragment and arrange all data (21 s).

As easily noticed, the global speed-up is around 14 times.

### B. Parallel Implementation—Results

Once the serial implementation has been designed, the fraction of intrinsically serial  $f_s$  and parallel  $f_p$  work must be estimated in order to evaluate the maximum achievable speed-up  $S$  in a parallel environment. In fact, in accordance with Amdhal's law [21], if  $NP$  is the number of processors, we have

$$S = \frac{1}{f_s + \frac{f_p}{NP}}. \quad (8)$$

The parallel task is, in our case, the solution of the many independent subsystems derived from the starting one. Of course,  $f_s$  and  $f_p$  are strongly correlated with the original matrix and with the several parameters influencing the system decomposition. Referring to the circuit of Fig. 1,  $f_s$  is approximately 0.13, and the maximum achievable speed-up is approximately 7.7.

The implementation of the strategy in a parallel environment can be performed in several different manners. In this paper, we focus our attention on two main scenarios, i.e., an MS implementation, and an SPMD approach. Both are supported by a parallel virtual machine (PVM)/MPI environment. In the MS implementation, a master program coordinates a variable number of slave programs running on different processors. Two different kinds of programs must be written down. In the SPMD implementation, a single program is written and, on each processor, an instance of the same program is running in the meantime.

The referenced platform is an eight-node IBM Scalable Parallel 2. Each node is thin, with an IBMRS6000 390 Power2 processor (66 MHz), and the point-to-point connection has a bandwidth of 9.9 MB/s and a latency of 312  $\mu$ s.

1) *MS Implementation*: In the MS implementation, the master process is engaged with fragmenting the linear system, and scattering submatrices to all slave processes. It finally gathers solutions and builds up the whole solution vector. The optimum subsystem distribution, which is estimated by applying the optimization strategy previously described, is accomplished as described in Table II so that the best load-balancing and minimum execution time is achieved.

The performance achieved with the MS implementation is summarized in Table III (times are given in seconds).

As noticed, interprocessor communication plays a major role and limits the achieved speed-up to 3.5, as attained by comparing the solution time in Table III with the one in Table I.

TABLE II  
SUBSYSTEM PARTITIONING AND ASSIGNMENT TO DIFFERENT PROCESSORS. FOR EACH PROCESSOR, THE SIZE OF A SUBSYSTEM AND THE RELATIVE NUMBER OF SUBSYSTEM (IN BRACKETS) ARE GIVEN

Processor	Assigned Subsystems (size)
1	1 (174), 533 (1)
2	1 (68), 1 (16), 2 (2)
3	1 (35), 2 (16), 2 (12), 1 (4)
4	1 (20), 1 (16), 3 (12), 2 (8), 1 (6)
5	1 (18), 2 (16), 2 (12), 1 (8)
6	1 (16), 1 (12), 4 (10), 2 (8)
7	2 (16), 3 (10), 1 (8)
8	1 (16), 6 (12)

TABLE III  
TIMES (IN SECONDS) FOR THE DIFFERENT TASKS IN THE MS IMPLEMENTATION

System Solution	System Partitioning	Communication	Total
24	5	18	47

TABLE IV  
TIMES (IN SECONDS) FOR THE DIFFERENT TASKS IN THE SPMD IMPLEMENTATION

System Solution	System Partitioning	Communication	Total
24	3	0.3	27.3

2) *SPMD Implementation*: In the SPMD implementation, all the processes perform the same action, assuming that, on each processor, the  $\mathbf{Y}$  has been stored. Of course, each process performs the system fragmentation, but solves only a subset of all the generated subsystems (the partitioning policy is the same as described for the MS implementation in Table II). In the SPMD case, the data structures and operations to support the system fragmentation are less heavy than in the MS case, and this slightly reduces the corresponding computing time, as noticed from Table IV, where times for the SPMD code are given.

A speed-up of over six is, therefore, observed, with respect to the serial computing time reported in Table I.

## V. DISCUSSION AND CONCLUSIONS

The speed-ups achieved by applying the proposed strategy are attractive and turn it into a viable methodology for real industrial applications. In fact, times reported in the previous benchmarks refer to a single analysis. In real industrial problems, where tuning and trimming of circuits is attained after long optimization processes, usually deserving thousands of circuit analyses, the proposed strategy removes a major limitation of the GAM approach, i.e., its computational complexity. This can be achieved at low costs on affordable platforms: we have proven that, on serial platforms, it performs more than the standard GAM for a factor larger than 14 times.

The advantage of the proposed strategy is also more evident when referring to distributed/parallel platforms, where we have achieved substantial speed-ups: on an IBM SP2 with eight processors a 6.3 speed-up is observed, with respect to a theoretical limitation of 7.7. Furthermore, the approach proposed here is also quite amenable to be implemented in a low-cost Ethernet etherogenous distributed memory PVM/MPI environment, as

the low influence of communication tasks in the SPMD version also guarantees good performance on such a network.

Moreover, and of primary importance, the proposed approach is also useful in order to solve severe numerical problems of ill conditioning, which sometimes affect the GAM approach. We have proven on a real case that the AG-RT strategy is able to divide the starting system into independent well-conditioned subsystems, thus allowing the efficient analysis of the circuit. It has also been verified that the parallel optimized AG-RT GAM strategy is nearly four times faster than the standard generalized-scattering-matrix formulation of the same approach.

Finally, the strategy proposed here is general, and can be extended to many other numerical methods based on circuit theory, and its application is currently under evaluation for several other numerical methods for electromagnetic (EM) circuit simulation in distributed environments.

## REFERENCES

- [1] F. Alessandri, G. Bartolucci, and R. Sorrentino, "Admittance matrix formulation of waveguide discontinuity problems: Computer-aided design of branch guide directional couplers," *IEEE Trans. Microwave Theory Tech.*, vol. 36, pp. 394–403, Dec. 1988.
- [2] W. Hauth, R. Keller, U. Papziner, R. Ihmels, T. Sieverding, and F. Arndt, "Rigorous CAD of multiport coupled rectangular waveguide components," in *Proc. 23rd Eur. Microwave Conf.*, Madrid, Spain, 1993, pp. 611–614.
- [3] K. C. Gupta, R. Garg, and R. Chadka, *Computer Aided Design of Microwave Circuits*. Norwood, MA: Artech House, 1981.
- [4] F. Alessandri, M. Mongiardo, and R. Sorrentino, "A technique for the full-wave automatic synthesis of waveguide components: Applications to fixed phase shifters," *IEEE Trans. Microwave Theory Tech.*, vol. 40, pp. 1484–1495, July 1992.
- [5] ———, "New efficient full wave optimization of microwave circuits by the adjoint network method," *IEEE Microwave Guided Wave Lett.*, vol. 3, pp. 414–416, Nov. 1993.
- [6] A. Morini, T. Rozzi, and M. Mongiardo, "Efficient CAD of wideband contiguous channel multiplexers," in *Proc. IEEE MTT-S Int. Microwave Symp. Dig.*, San Francisco, CA, June 1996, pp. 1651–1654.
- [7] L. Tarricone and M. Mongiardo, "A stable and efficient admittance method via adjacency graphs and recursive thresholding," *IEEE Trans. Microwave Theory Tech.*, vol. 49, pp. 1750–1756, Oct. 2001.
- [8] J. M. Rebollar, J. Esteban, and J. E. Page, "Full-wave analysis of three and four-port rectangular waveguide junctions," *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 256–263, Feb. 1994.
- [9] A. A. Melcon, G. Connor, and M. Guglielmi, "New simple procedure for the computation of the multimode admittance or impedance matrix of planar waveguide junctions," *IEEE Trans. Microwave Theory Tech.*, vol. 44, pp. 413–418, Mar. 1996.
- [10] W. Wessel, T. Sieverding, and F. Arndt, "Mode-matching analysis of general waveguide multiport junctions," in *Proc. IEEE MTT-S Int. Microwave Symp. Dig.*, 1999, pp. 1273–1276.
- [11] J. A. Dobrowolski, *Introduction to Computational Methods for Microwave Circuit Analysis and Design*. Norwood, MA: Artech House, 1991.
- [12] F. X. Canning, "A new combined field integer," presented at the IEEE AP-S Int. Symp., Ann Arbor, MI, 1993.
- [13] E. Michielssen and A. Boag, "A multilevel matrix decomposition algorithm for analyzing scattering from large structures," *IEEE Trans. Antennas Propagat.*, vol. 44, pp. 1087–1093, Aug. 1996.
- [14] V. Rokhlin, "Rapid solution of integral equations of scattering in two dimensions," *J. Comput. Phys.*, vol. 86, pp. 414–439, 1990.
- [15] V. Rizzoli, A. Neri, F. Mastri, and A. Lipparini, "A Krylov-subspace technique for the simulation of integrated RF/MW subsystems driven by digitally modulated carriers," *Int. J. RF Microwave Computer-Aided Eng.*, vol. 9, pp. 490–505, 1999.
- [16] R. F. Remis and P. M. Van den Berg, "A modified Lanczos algorithm for the computation of transient electromagnetic wavefields," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 2139–2149, Dec. 1997.
- [17] K. Radhakrishnan and W. C. Chew, "Full-wave analysis of multiconductor transmission lines on anisotropic inhomogeneous substrates," *IEEE Trans. Microwave Theory Tech.*, vol. 47, pp. 1764–1770, Sept. 1999.
- [18] J. Arroyo and J. Zapata, "Subspace search iteration method for generalized eigenvalue problems with sparse complex unsymmetric matrices in finite-element analysis of waveguides," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1115–1123, Aug. 1998.
- [19] F. Robert, "Blocs- $H$ -matrices et convergences des méthodes itératives classiques par blocs," *Linear Algebra Appl.*, vol. 2, pp. 223–265, 1969.
- [20] M. R. Garey and D. S. Johnson, *Computers and Intractability*. San Francisco, CA: Freeman, 1971.
- [21] J. Gustavson, "Reevaluating Amdahl's law," *Commun. ACM*, pp. 532–533, 1988.
- [22] J. A. Hoogeveen, J. K. Lenstra, and S. L. Van de Velde, "Sequencing and scheduling," in *Annotated Bibliographies in Combinatorial Optimization*, M. Dell'Amico, F. Maffioli, and S. Martello, Eds. New York: Wiley, 1997.
- [23] R. E. Tarjan, *Data Structures and Network Algorithms*. Philadelphia, PA: SIAM, 1983.
- [24] M. Righi, J. L. Herring, and W. J. R. Hoefer, "Efficient hybrid TLM-mode matching analysis of packaged components," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1715–1724, Oct. 1997.

**Alessandra Esposito** received the Laurea degree in electronic engineering from Naples University "Federico II," Naples, Italy, in 1990.

From 1994 to 1994, she was a Researcher with IBM, where she was involved in pattern recognition. In 1995, she was a system engineer with Sodalia, Trento, Italy. Since 1996, she has been a consultant involved in several research projects concerning high-performance computing and innovative information-technology solutions for industrial applications. She is currently with the Department of Innovation Engineering, University of Lecce, Lecce, Italy. She is also currently involved with the management of small and medium enterprise needs in Web-based environments on behalf of the Associazione degli Industriali, Perugia, Italy. She has authored over 40 scientific papers.

**Federico Malucelli** was born in Ferrara, Italy, on July 4, 1962. He received the Laurea degree and Ph.D. degree in computer science from Pisa University, Pisa, Italy, in 1988 and 1993, respectively.

From 1992 to 1998, he was a Research Associate with the University of Pisa and University of Perugia. He is currently an Associate Professor of operations research with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy. He has authored approximately 30 scientific papers appearing in international journals. His main research focuses on combinatorial optimization methods, with applications in the field of telecommunications, EM simulation, and transportations.

**Luciano Tarricone** was born in Galatone, Lecce, Italy, on May 24, 1966. He received the Laurea degree (with honors) in electronic engineering and the Ph.D. degree from Rome University "La Sapienza," Rome, Italy, in 1989 and 1994, respectively. Both his Laurea and Ph.D. theses were focused on the biological effects of EM fields and electromagnetic compatibility (EMC).

In 1990, he was a Visiting Researcher with the Italian National Institute of Health Laboratories, where he was in charge of European draft standards for electromagnetic interferences (EMIs) with implanted devices. From 1990 to 1992, he was a Researcher with the IBM Rome Scientific Centers. From 1992 to 1994, he was with the IBM European Center for Scientific and Engineering Computing, Rome, Italy, where he was involved in supercomputing for several scientific applications. Since 1994, he has been a Researcher with Department of Information Engineering, University of Perugia, Perugia, Italy, and since 1998, he has been Professore Incaricato of EM fields and EMC. Since November 2001, he has also been a faculty member with the Department of Innovation Engineering, University of Lecce, Lecce, Italy. He has authored approximately 150 scientific papers. His main contributions are in the modeling of microscopic interactions of EM fields and biosystems, and in numerical methods for efficient computer-aided design (CAD) of MW circuits and antennas. He is currently involved in the finite-difference time-domain (FDTD) analysis of human–antenna interaction, graph-theory methods for the enhancement of numerical EM techniques, novel CAD tools and procedures for MW circuits, and EM parallel computing.