

New techniques for selecting test frequencies for linear analog circuits

Mohand Bentobache, Ahcène Bounceur, Reinhardt Euler, Yann Kieffer,
Salvador Mir

► **To cite this version:**

Mohand Bentobache, Ahcène Bounceur, Reinhardt Euler, Yann Kieffer, Salvador Mir. New techniques for selecting test frequencies for linear analog circuits. International Conference on Very Large Scale Integration (VLSI-SoC), Oct 2013, Istamboul, Turkey. 2013. <hal-00855154>

HAL Id: hal-00855154

<http://hal.univ-brest.fr/hal-00855154>

Submitted on 29 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

New techniques for selecting test frequencies for linear analog circuits

Mohand Bentobache*, Ahcène Bounceur†, Reinhardt Euler†, Yann Kieffer‡ and Salvador Mir§

*LAMOS Laboratory, University of Bejaia, 06000, Algeria, Email: mbentobache@yahoo.com

†Lab-STICC Laboratory, University of Brest, 20, Avenue Victor Le Gorgeu, 29238, Brest, France,

Email: Ahcene.Bounceur,Reinhardt.Euler@univ-brest.fr

‡LCIS Laboratory, University of Grenoble Alpes, 26900 Valence, France,

Email: Yann.Kieffer@esisar.grenoble-inp.fr

§TIMA Laboratory, 46, av. Félix Viallet, 38031 Grenoble, France,

Email: Salvador.Mir@imag.fr

Abstract—In this paper we show that the problem of minimizing the number of test frequencies necessary to detect all possible faults in a multi-frequency test approach for linear analog circuits can be modeled as a set covering problem. We will show in particular, that under some conditions on the considered faults, the coefficient matrix of the problem has the strong consecutive-ones property and hence the corresponding set covering problem can be solved in polynomial time. For an efficient solution of the problem, an interval graph formulation is also used and a polynomial algorithm using the interval graph structure is suggested. The optimization of test frequencies for a case-study biquadratic filter is presented for illustration purposes. Numerical simulations with a set of randomly generated problem instances demonstrate two different implementation approaches to solve the optimization problem very fast, with a good time complexity.

Index Terms—Set covering problem, Consecutive-ones property, Analog circuit testing, Linear programming, Interval graphs.

I. INTRODUCTION

The optimization of test sets for integrated circuits is critical for reducing the large costs of production testing. For digital circuits, fault-based test pattern generation techniques are used. This structural approach is less common for analog circuits that are still tested using specification-based approaches. The variety of analog and mixed-signal blocks and the lack of general fault models have prevented a widespread use of structural techniques. However, there is a pressing need to adopt fault-based approaches also for analog circuits, providing measures of fault coverage for optimized test sets, especially in the context of SoC devices that embed digital and mixed-signal blocks in a single chip.

Multi-frequency tests (i.e. multi-tone sinusoidal signals) have been classically considered for the test and diagnosis of linear analog devices such as analog filters. Since the effect of parametric and catastrophic faults varies as a function of frequency, it is possible to derive a minimal set of test frequencies either for the detection or for the diagnosis of all potential faults. To optimize the set of test frequencies, the approaches based on sensitivity analysis have typically addressed parametric faults [1], [2]. However, these approaches are not

accurate for very large deviations, such as those that result from catastrophic faults. On the other hand, the approaches based on fault simulation can handle catastrophic faults, but at the expense of very time consuming simulations when realistic faults at transistor-level are considered. Today, it is evident that fault simulation of analog circuits is becoming essential in order to optimize test sets relying on new techniques to accelerate fault simulation.

In this context, this paper proposes a new technique for the optimization of multi-frequency tests for linear analog circuits. Fault simulation is used to obtain the frequency intervals for the detection of each fault. New efficient algorithms are then presented for the selection of the optimal set of test frequencies within these intervals for the detection of all faults. A simple case-study is used to illustrate the algorithms. Numerical simulations with randomly generated problem instances demonstrate the good time complexity of the proposed algorithms, with a large improvement over previous approaches [3]. We notice that the test optimization algorithms are general, applicable to any case-study requiring an optimization of multi-frequency tests based on fault simulation data. This technique is today feasible for analog filters, but it is also applicable to other analog devices such as analog-to-digital converters or radio-frequency front-ends requiring multi-tone tests, provided that fault simulation data are made available.

The paper is organized as follows: in Section 2, the mathematical formulation of the Set Covering Problem (SCP) is reviewed and some definitions are given. In Section 3, we present the mathematical formulation of the problem of minimizing the number of frequency intervals necessary to detect the faults of an analog circuit, and we study the specific structure of the related coefficient matrix. In Section 4, we present two approaches for solving the problem: the Linear Programming (LP) approach and the interval graph approach. In Section 5, we present a case-study of testing a biquadratic filter. In Section 6, we carry out a large-scale numerical study in order to compare both approaches and to evaluate their time complexity. Finally, Section 7 concludes the paper and provides some perspectives.

II. THE SET COVERING PROBLEM

The Set Covering Problem is one of the most important models in combinatorial optimization. Indeed, a wide range of real-world problems are modeled as SCP, namely: railway crew scheduling, airline crew scheduling, facility location, etc. The practical importance of the SCP has motivated many researchers to develop efficient algorithms and heuristics for finding good solutions in reasonable time. We can cite: exact algorithms based on branch-and-bound or branch-and-cut [4], greedy heuristics [5], Lagrangian-based heuristics [6], genetic algorithms [7], etc. In order to test the efficiency of the proposed algorithms, there exists a library of SCP Benchmarks (Beasley's OR Library [8]).

The set covering problem is known to be NP-hard [9]. However, there exist some particular forms of the SCP which are polynomial-time solvable such as the SCP with a constraint matrix having the consecutive-ones property (i.e., the ones in each row appear consecutively). This particular case, denoted in the following by SCP-C1P, can be solved efficiently with LP algorithms.

A. Some definitions

Let $M = \{1, \dots, m\}$ and $N = \{1, \dots, n\}$ be two sets of indices. Let $A = (a_{ij}, i \in M, j \in N)$ be a binary $(m \times n)$ -matrix and $c = (c_j, j \in N)$ be an integer n -vector. The value $c_j, j \in N$, represents the cost of column j and we assume without loss of generality that $c_j > 0$ for all $j \in N$. We say that a column $j \in N$ covers a row $i \in M$ if $a_{ij} = 1$. A set $S \subseteq N$ is called a *cover* if each row $i \in M$ is covered by at least one column $j \in S$. The problem of finding a cover S of minimum cost can be formulated as an ILP (Integer Linear Programming) problem as follows:

$$\begin{aligned} \min z &= c^T x, \\ \text{subject to } Ax &\geq \mathbf{1}_m, \quad x \in \{0, 1\}^n, \end{aligned} \quad (1)$$

where $\mathbf{1}_m$ is the m -vector of ones. When $c_j = 1$, for all $j \in N$, problem (1) is called a *unicost set covering problem*.

Definition 1: ([10]).

- A *block* of 1's (*block* of 0's) in a row of a binary matrix A is a maximal set of consecutive 1-entries (0-entries) in this row.
- A binary matrix has the *strong consecutive-ones property* (*strong C1P*) if in every row the 1's appear consecutively, that is, if every row contains at most one block of 1's.
- A binary matrix has the *consecutive-ones property* (*C1P*) if its columns can be permuted in such a way that the resulting matrix has the strong C1P. If an ordering for the columns of a binary matrix yields the strong C1P, it is called a *C1-ordering*.
- A graph G is an *interval graph* if its vertices can be mapped to intervals on the real line such that two vertices are adjacent if and only if their corresponding intervals have non empty intersection.
- A binary matrix is *totally unimodular* if every square submatrix has determinant 0, 1, or -1.

III. MINIMIZATION OF TEST FREQUENCIES TO DETECT ALL THE FAULTS OF AN ANALOG CIRCUIT

In [3], the problem of selecting a minimum number of test frequencies is formally presented and solved using boolean expressions. However, when the number of faults is large, the technique used is not efficient. In this section, we present a mathematical formulation of the general problem as a set covering problem. We show that, under some special conditions on the considered faults, the coefficient matrix of the SCP will have the C1P. Moreover, we suggest an interval graph formulation of the problem.

A. Mathematical formulation of the problem as an SCP-C1P

Consider a given linear analog circuit C . Let $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ be a set of all the possible faults which can occur in C . Let T be a test measure which will be used in the fault detection process. In order to detect a given fault F_i using the test measure T , test signals with maximum amplitude and at different frequencies in the interval $[f_{min}, f_{max}]$ are used as inputs of the analog circuit. The fault is detected if the test measure T exceeds a fixed threshold τ . Fault simulation allows then to compute, for each fault F_i , the frequency intervals for which the threshold is exceeded. These intervals are called the *detection regions* of fault F_i . Note that each fault can have one, two or more detection regions. Let n_i be the number of detection regions of the fault F_i and $\mathcal{R}_i = \{R_{i1}, R_{i2}, \dots, R_{in_i}\}$ be the set of all the detection regions of the fault F_i . Note that these detection regions are disjoint: for each two detection regions R_{ip} and R_{iq} , we have $R_{ip} \cap R_{iq} = \emptyset$. After that, a set of n frequency intervals $\mathcal{I} = \{I_1, I_2, \dots, I_n\}$ is computed as follows: we divide the interval $[f_{min}, f_{max}]$ into n disjoint intervals such that

$$[f_{min}, f_{max}] = [f_{min}, f_1] \cup [f_1, f_2] \cup \dots \cup [f_{n-1}, f_{max}], \quad (2)$$

where $f_k, k = 1, \dots, n-1$, represent the lower or upper bounds of all the detection regions sorted in increasing order ($f_{min} < f_1 < f_2 < \dots < f_{n-1} < f_{max}$).

We denote by $A = (a_{ij}, i = 1, \dots, m, j = 1, \dots, n)$ the matrix which is defined as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } \exists k \in \{1, \dots, n_i\}: I_j \subseteq R_{ik}; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The problem consists in finding a minimal cardinality set of frequency intervals $S \subseteq \mathcal{I}$ which can detect all the possible considered faults. One frequency for each selected frequency interval (typically in the middle of the interval) can be used in the optimized test set. This problem can be formulated as a unicast set covering problem: for each $S \subseteq \mathcal{I}$ and each frequency interval $I_j, j \in \{1, \dots, n\}$, we define a binary variable as follows:

$$x_j^S = \begin{cases} 1, & \text{if } I_j \text{ belongs to the set } S; \\ 0, & \text{otherwise.} \end{cases}$$

Now, if $x = (x_j, j = 1, \dots, n)$ is any vector of binary variables, then the mathematical model corresponding to this

problem will be given by:

$$\begin{aligned} \min r &= \mathbf{1}_n^T x, \\ \text{s.t. } Ax &\geq \mathbf{1}_m, x \in \{0, 1\}^n. \end{aligned} \quad (4)$$

Remark 1: The matrix A cannot have more than $2 \sum_{i=1}^m n_i$ columns.

Proposition 1: If $n_i = 1$, for $i = 1, \dots, m$, i.e., each fault has a unique detection region, then the matrix A will have the strong consecutive-ones property.

Proof: Let $F_i, i \in M$, be a given detected fault and $R_{i1} = [a_i, b_i]$ be the unique detection region of fault i . By construction, R_{i1} can be written as a union of k consecutive intervals $I_{j_1}, I_{j_2}, \dots, I_{j_k}$, where a_i is the lower bound of I_{j_1} and b_i is the upper bound of I_{j_k} . (Two intervals are consecutive if they have a common bound). Since the intervals I_{j_r} correspond to the consecutive columns a_{j_r} of A for $r = 1, \dots, k$, the detection regions correspond to the rows of A , and $I_{j_r} \subseteq R_{i1}$, i.e. $a_{ij_r} = 1$ for $r = 1, \dots, k$. ■

Note that the previous proposition is very important because it gives a simple sufficient condition for the matrix arising in our application to have the strong C1P.

In the following we consider only problems with faults having one detection region.

Following Proposition 1, ILP problem (4) is a unicast set covering problem with a coefficient matrix having the strong C1P (SCP-C1P). More precisely, it is given by:

$$\begin{aligned} \min z &= \sum_{j=1}^n x_j, \\ \text{s.t. } \sum_{j=lx(i)}^{rx(i)} x_j &\geq 1, i \in M, x_j \in \{0, 1\}, j \in N, \end{aligned} \quad (5)$$

where for each row i , $lx(i)$ denotes the leftmost index l for which $a_{il} = 1$ and $rx(i)$ the rightmost index r for which $a_{ir} = 1$. To get the LP-relaxation of the above problem, we simply exchange the integrality constraints against the non-negativity constraints [11]. Thus, we get the following LP problem:

$$\begin{aligned} \min z &= \sum_{j=1}^n x_j, \\ \text{s.t. } \sum_{j=lx(i)}^{rx(i)} x_j &\geq 1, i \in M, x_j \geq 0, j \in N. \end{aligned} \quad (6)$$

B. Graph formulation of the problem

Previously, we have formulated the problem of minimizing frequency intervals to detect all the faults of an analog circuit using an ILP model. In this subsection, we suggest a new formulation based on the concept of interval graph: we denote the detection region of fault i by the interval $[a_i, b_i]$ for $i \in M$. Let the interval graph $G = (F, E_F)$ be defined as follows:

$$F = \{[a_i, b_i], i \in M\}, E_F = \{F_i F_j : [a_i, b_i] \cap [a_j, b_j] \neq \emptyset\}. \quad (7)$$

In the next section we will suggest a polynomial algorithm using this interval graph for solving the considered problem.

IV. APPROACHES FOR SOLVING THE MINIMIZATION PROBLEM OF FREQUENCY INTERVALS

In this section, we suggest two approaches for solving the problem of minimizing frequency intervals necessary to detect all the faults of an analog circuit.

A. LP Approach

Let us recall the following results:

Theorem 1: ([12]).

An $(m \times n)$ -matrix A with entries 0, 1 and -1 is totally unimodular if and only if each collection of columns from A can be partitioned into two column sets such that in each row the sum of the entries of the first set and the sum of the entries of the second set differ by at most 1.

Theorem 2: ([13]).

Let A be an $m \times n$ integral matrix. Then the polyhedron defined by $Ax \leq b$ and $x \geq 0$ is integral for every integral vector b if and only if A is totally unimodular.

Remark 2: ([10]).

Any matrix A having the C1P fulfills the conditions of Theorem 1 and, hence, is totally unimodular.

Following Theorem 2 and Remark 2, any basic feasible solution of the LP problem (6) represents a cover for the SCP-C1P (5). Therefore, we can find an optimal solution using LP algorithms such as the primal or dual simplex method [14], the support method [15], etc. However, it is more efficient to transform the SCP-C1P into a min-cost network flow problem [10].

Let us make the Veinott-Wagner transformations [17] for the variables of problem (6): we introduce the variables $y_j, j = 1, \dots, n+1$ such that $x_j = -y_j + y_{j+1}, j = 1, \dots, n$. Hence, we obtain the following equivalent LP problem:

$$\begin{aligned} \min z &= -y_1 + y_{n+1}, \\ \text{s.t. } -y_{lx(i)} + y_{rx(i)+1} &\geq 1, i \in M, \\ -y_j + y_{j+1} &\geq 0, j \in N. \end{aligned} \quad (8)$$

The dual of the above problem has $m+n$ variables v_1, v_2, \dots, v_{m+n} and $n+1$ constraints and it is given by:

$$\begin{aligned} \min w &= -\sum_{j=1}^m v_j, \\ \text{s.t. } \sum_{j=1}^{m+n} a'_{1j} v_j &= -1, \\ \sum_{j=1}^{m+n} a'_{ij} v_j &= 0, i = 2, \dots, n, \\ \sum_{j=1}^{m+n} a'_{(n+1)j} v_j &= 1, v_j \geq 0, j = 1, \dots, m+n. \end{aligned} \quad (9)$$

where $A' = (a'_{ij}, i = 1, \dots, n+1, j = 1, \dots, m+n)$; and A' has exactly one 1 and one -1 in each column. Remark that the LP problem (9) is a min-cost network flow problem. Thus, it can be solved by the network simplex method.

The scheme of the LP approach to solve the problem is described in the following steps:

- Step 1.** Compute the frequency intervals necessary to detect the different faults using relation (2);
- Step 2.** Compute the constraint matrix of the SCP-C1P using relation (3);
- Step 3.** Make the Veinott-Wagner transformations. Let V be the constraint matrix of the LP problem (8);
- Step 4.** Compute the constraint matrix A' , the $(n+1)$ -vector of right-hand-sides b' and the cost $(m+n)$ -vector c' of the min-cost network flow problem (9): $A' = V^T, b' = (-1, 0, \dots, 0, 1)^T, c' = \begin{pmatrix} -\mathbf{1}_m \\ \mathbf{0}_n \end{pmatrix}$;

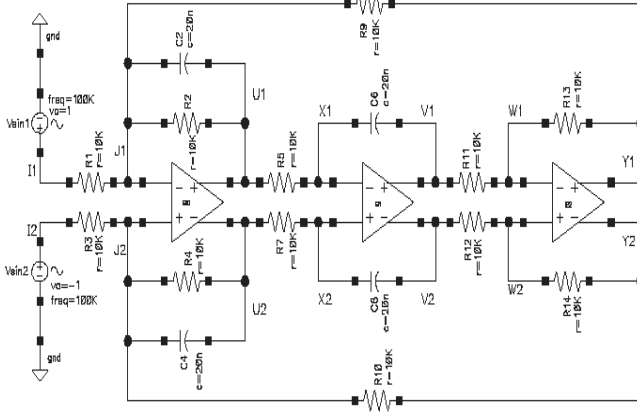


Figure 1. Biquadratic Filter

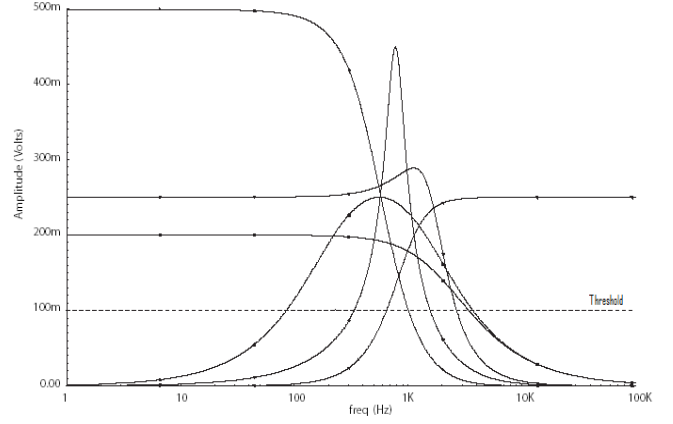


Figure 2. Output signals using test measure T_1

Step 5. Solve the min-cost network flow problem with the network simplex algorithm.

B. Interval graph approach

In order to solve the problem using the interval graph G defined by (7), we suggest the following algorithm:

Algorithm 1:

1. Let L be the list of vertices of the graph G . Sort the vertices F_i , $i = 1, \dots, m$ by increasing order of their upper bounds b_i ;
2. Set $s = m$ and $S = \emptyset$;
3. **For** $i = 1$ to s
 - 3.1. Let k be the index verifying $b_i = f_k$, where f_k is computed by (2);
 - 3.2. **For** $j = i + 1$ to s

If F_j is adjacent to F_i in G , then

 - delete F_j from L ;
 - set $s = s - 1$ and $S = S \cup \{[f_{k-1}, f_k]\}$;

endif

endifor

The input of this polynomial algorithm is the interval graph G and the output is a minimal cardinality set S of frequency intervals, which detects all the faults.

Note that it is not necessary to construct the whole interval graph, we can use only its vertices, i.e., detection regions of the different faults, and replace the test "If F_j is adjacent to F_i in G " by the test "If $a_j < b_i \leq b_j$ ".

Remark 3: Contrarily to the LP approach, the interval graph approach uses the detection regions of the considered faults directly, i.e. without transforming the problem.

V. CASE-STUDY

To illustrate our approach, similar to [3], we will now present a case-study biquadratic filter as shown in Figure 1. There are 6 test measures for this circuit that correspond to the common-mode signal at the input and at the output of each operational amplifier. For simplicity, only catastrophic (10 MOhm open and 1 Ohm short) faults in the passive components are considered. Due to the differential design, only 16 different faults need to be considered.

We denote the test measures by T_1, T_2, \dots, T_6 and the faults by F_1, F_2, \dots, F_{16} . Figure 2 shows the frequency behavior of test measure T_1 for some of these faults. The detection regions of each fault F_i , $i = 1, 2, \dots, 16$ using test measures T_k , $k = 1, 2, \dots, 6$ are computed using the fault simulator developed in [18], [19]. Note that the simulation results obtained in [18] indicate that test measures T_2, T_4 and T_6 do not detect any fault. Hence, we only consider test measures T_1, T_3 and T_5 . For simplicity also, we have only considered nominal simulations of the catastrophic faults. In practice, Monte Carlo simulations of each catastrophic fault should be considered, and worst-case detection regions be computed (that is, the intersection of the detection regions for each Monte Carlo instance).

First, we start by minimizing the frequency intervals under the test measure T_1 : the faults detected using test measure T_1 are F_1, F_2, \dots, F_8 . The detection regions of the different faults are:

$$R_{11} = [1, 10^5], R_{21} = [85, 3732], R_{31} = [85, 3732], \\ R_{41} = [1, 2685], R_{51} = [1, 3442], R_{61} = [336, 1566], \\ R_{71} = [1, 1014] \text{ and } R_{81} = [647, 10^5].$$

When we sort in increasing order the bounds of the different detection regions, we find the following series of numbers f_i , $i = 1, \dots, 10$:

$$1; 85; 336; 647; 1014; 1566; 2685; 3442; 3732; 10^5.$$

Therefore, the frequency intervals are:

$$I_1 = [1, 85[, I_2 = [85, 336[, I_3 = [336, 647[, \\ I_4 = [647, 1014[, I_5 = [1014, 1566[, I_6 = [1566, 2685[, \\ I_7 = [2685, 3442[, I_8 = [3442, 3732[, I_9 = [3732, 10^5].$$

The constraint matrix of the SCP (4) is then:

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

where each row $i, i = 1, \dots, 8$ corresponds to the detection region R_{i1} and each column $j, j = 1, \dots, 9$ corresponds to the frequency interval I_j .

Since the matrix A has the consecutive-ones property (each fault has a unique detection region), the optimal solution of the SCP (4) can be obtained using the network simplex method. In our example, this leads to the following optimal solution: $x^* = (0, 0, 0, 1, 0, 0, 0, 0)^T$ and $r^* = 1$, i.e., a minimal cardinality set of intervals detecting all the faults is $S = \{I_4\}$. Therefore, it is sufficient to use a single test frequency belonging to the interval $[647Hz, 1014Hz[$ to detect all the faults F_1, F_2, \dots, F_8 .

Similarly, when we use the test measure T_3 , we detect the faults F_9, F_{10}, F_{11} and F_{12} . The corresponding detection regions are:

$$R_{91} = [159, 7957[, R_{10,1} = [1, 1740[, \\ R_{11,1} = [1, 1739[\text{ and } R_{12,1} = [159, 7940[.$$

The different frequency intervals are:

$$I'_1 = [1, 159[, I'_2 = [159, 1739[, I'_3 = [1739, 1740[, \\ I'_4 = [1740, 7940[\text{ and } I'_5 = [7940, 7959[.$$

The constraint matrix of the SCP is

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

Hence, for the test measure T_3 , the solution of the SCP (4) is obvious, that is, a single test frequency belonging to the interval $I'_2 = [159Hz, 1739Hz[$ detects all the faults F_9, F_{10}, F_{11} and F_{12} .

Finally, with test measure T_5 , we detect the faults F_{13}, F_{14}, F_{15} and F_{16} . The corresponding detection regions are:

$$R_{13,1} = [1, 2798[, R_{14,1} = [1, 1413[, R_{15,1} = [1, 1412[, \\ \text{and } R_{16,1} = [1, 2794[.$$

The different frequency intervals are:

$$I''_1 = [1, 1412[, I''_2 = [1412, 1413[, I''_3 = [1413, 2794[, \\ \text{and } I''_4 = [2794, 2798[.$$

The constraint matrix of the SCP is

$$A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{pmatrix}.$$

Thus, for the test measure T_5 , the solution is obvious again, that is, the frequency interval which detects all the faults F_{13}, F_{14}, F_{15} and F_{16} is $I''_1 = [1Hz, 1412Hz[$, and a single test frequency belonging to this interval is required.

VI. NUMERICAL EXPERIMENTS

In order to compare the efficiency of the two approaches presented in Section IV (LP approach and the interval graph one), we have implemented them in C++ programming language and carried out large-scale numerical experiments on a set of randomly generated test instances using an Intel(R)

Core(TM)2 Duo CPU P8600 @ 2.40GHz machine with 4 GB of RAM.

We have generated 60 problems with number of faults $m = 500, 000; 600, 000; 700, 000; 800, 000; 900, 000; 1, 000, 000$.

The detection region bounds are generated in the interval $[1Hz, 10^5Hz]$. For each class of test problems with m faults, we generate ten problems. We have solved the different instances with the LP approach (LPA) using the network simplex method of the LP and ILP solver CPLEX [20] and the suggested interval graph algorithm (IGA). The CPU time of the two approaches IGA and LPA are reported in Table I.

Let $[a_i, b_i]$ be the detection regions of the faults $F_i, i = 1, \dots, m$. In the following, we give the implementation details for the LP approach:

Step 1. Compute the set of frequency intervals \mathcal{I} as follows:

- sort the bounds of the detection regions in increasing order: let d be the vector of these sorted bounds;
- delete duplicate elements from d , and let $n + 1$ be the dimension of d ;
- set $\mathcal{I} = \{I_j = [d_j, d_{j+1}[, j = 1, \dots, n\}$;

Step 2. Compute the constraint $(m \times n)$ -matrix of the SCP-C1P as follows:

- set $fmax = d(n + 1)$ and compute the vector t of dimension $fmax$ as follows:
- set $t(k) = 0$, for $k = 1, \dots, fmax$;
- set $t(d(j)) = j$, for $j = 1, \dots, n + 1$;
- set $lx(i) = t(a_i)$ and $rx(i) = t(b_i) - 1$ for $i = 1, \dots, m$;

Step 3. Make the Veinott-Wagner transformations: compute the constraint matrix V of the LP problem (8) as follows:

- for $i = 1, \dots, m$; for $j = 1, \dots, m + n$:

$$V_{ij} = \begin{cases} -1, & \text{if } j = lx(i); \\ 1, & \text{if } j = rx(i) + 1; \\ 0, & \text{otherwise;} \end{cases}$$
- for $i = m + 1, \dots, n + 1$; for $j = 1, \dots, m + n$:

$$V_{ij} = \begin{cases} -1, & \text{if } j = i - m; \\ 1, & \text{if } j = i - m + 1; \\ 0, & \text{otherwise;} \end{cases}$$

Step 4. Compute the constraint matrix A' , the $(n+1)$ -vector of right-hand-sides b' and the cost $(m + n)$ -vector c' of the min-cost network flow problem (9): $A' = V^T, b' = (-1, 0, \dots, 0, 1)^T, c' = \begin{pmatrix} -\mathbf{1}_m \\ \mathbf{0}_n \end{pmatrix}$;

Step 5. Solve the min-cost network flow problem with the network simplex algorithm.

Note that the efficient implementation presented above computes the constraint matrices of the SCP-C1P and the min-cost network flow problem of the LP approach in small CPU times (less than 1 second on average for all the test problems). That is why we have not reported the CPU times of computing the constraint matrices of the SCP-C1P and the min-cost network flow problem in Table I.

The optimal values found by the two approaches IGA and LPA for the different test problems are the same; the average optimal values for each class of test problems are shown in

m	OptVal	IGA	LPA	Ratio
500,000	560,80	4,10	3,18	1,29
600,000	611,90	5,30	3,52	1,51
700,000	672,50	6,70	3,85	1,74
800,000	704,90	9,40	4,30	2,19
900,000	762,90	12,00	4,53	2,65
1,000,000	809,10	16,30	4,76	3,42
Mean		8,97	4,02	2,13

Table I

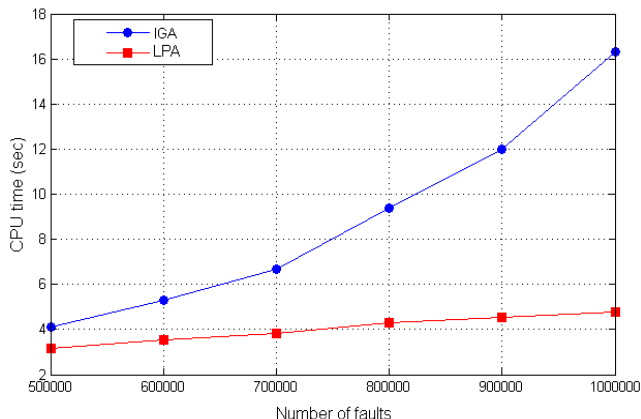


Figure 3. CPU time of the two approaches: IGA and LPA

column OptVal. We compute the ratios of the CPU time of IGA over LPA. These ratios are shown in column Ratio. Finally, the CPU time of the two approaches are plotted in Figure 3. The LPA shows a time complexity $O(m)$, while IGA shows a time complexity $O(m^2)$, where m is the number of faults.

The graphs of Figure 3 indicate that transforming the problem into a min-cost network flow problem and solving it with the network simplex method is more efficient than the interval graph algorithm. Indeed, LPA is, on average, two times faster than IGA. However, the computation time is very small even for problems with an extremely large number of faults. Therefore, the interval graph algorithm can also be used in practice by test engineers because of its easy implementation, the good CPU times (an average of 16 seconds for optimizing the test of analog circuits with 1,000,000 faults), and the fact that it solves the original problem directly.

VII. CONCLUSION

In this work, we have formulated as a SCP the problem of minimizing the number of test frequencies for detecting a set of faults injected into an analog circuit. We have shown that when the considered faults have a unique detection region, the constraint matrix of the SCP will have the strong consecutive-ones property. After that, we have reformulated this special case using interval graphs and an algorithm working directly with this graph is suggested. In order to solve the problem efficiently, two approaches are compared: an LP approach and an interval graph approach. The obtained numerical results show that the approach which transforms the problem into

a min-cost network flow problem to solve it by the network simplex method is the most efficient. However, the interval graph approach can also be used by test engineers because it solves the original problem directly; its implementation is very easy and it is extremely fast with CPU times of a few tenths of seconds even for large-scale problems. Future work will focus on the case when the faults have more than one detection region. We will study the complexity of the related problem and develop a branch-and-cut algorithm which takes into account the specific structure of the problem.

REFERENCES

- [1] C. Alippi, M. Catelani, A. Fort and M. Mugnaini. Automated selection of test frequencies for fault diagnosis in analog electronic circuits. *IEEE Transactions on Instrumentation and Measurement*, Vol. 54, No. 3, 2005, pp. 1033–1044.
- [2] F. Grasso, A. Luchetta, S. Manetti and M.-C. Piccirilli. A method for the automatic selection of test frequencies in analog fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, Vol. 56, No. 6, 2007, pp. 2322–2329.
- [3] S. Mir, M. Lubaszewski and B. Courtois. Fault-Based ATPG for linear analog circuits with minimal size multifrequency test sets. *Journal of Electronic Testing: Theory and Applications*, Vol. 9, No. 1/2, 1996, pp. 43–57.
- [4] E. Balas and M. C. Carrera. A Dynamic Subgradient-Based Branch-and-Bound Procedure for Set Covering. *Operations Research*, Vol. 44, 1996, pp. 875–890.
- [5] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, Vol. 4, No. 3, 1979, pp. 233–235.
- [6] J. E. Beasley. A Lagrangian Heuristic for Set Covering Problems. *Naval Research Logistics*, Vol. 37, 1990, pp. 151–164.
- [7] J. E. Beasley and P. C. Chu. A Genetic Algorithm for the Set Covering Problem. *European Journal of Operational Research*, Vol. 94, 1996, pp. 392–404.
- [8] J. E. Beasley. OR-Library: Distributing Test Problems by Electronic Mail. *Journal of Operational Research Society*, Vol. 41, 1990, pp. 1069–1072.
- [9] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, 1979.
- [10] M. Dom, *Recognition, Generation, and Application of Binary Matrices with the Consecutive-Ones Property*. PhD thesis, Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany, 2008.
- [11] J. Matoušek and B. Gärtner. *Understanding and Using Linear Programming*, Springer-Verlag, Berlin Heidelberg, 2007.
- [12] A. Ghouila-Houri. Caractérisation des matrices totalement unimodulaires. *C. R. Acad. Sci. Paris*, Vol. 254, 1962, pp. 1192–1194.
- [13] A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra, In H. W. Kuhn and A. W. Tucker, editors, *Linear Inequalities and Related Systems*, pp. 223–246, Princeton University Press, 1956.
- [14] G. B. Dantzig. *Linear Programming and Extensions*, Princeton University Press, Princeton, N.J., 1963.
- [15] M. Bentobache and M. O. Bibi. A two-phase support method for solving linear programs: Numerical experiments, *Mathematical Problems in Engineering*, Vol 2012, Article ID 482193, 28 pages doi:10.1155/2012/482193, 2012.
- [16] M. Bentobache, A. Bounceur and R. Euler. Une application efficace du problème de recouvrement au test de circuits analogiques, in *Proceedings of ROADEF 2012*, University of Angers, France, pp. 424–425, 11-13 April 2012.
- [17] A. F. Veinott and H. M. Wagner. Optimal capacity scheduling. *Operations Research*, Vol. 10, 1962, pp. 518–547.
- [18] A. Bounceur, CAO Platform for mixed circuit testing, PhD thesis, Grenoble INP, 2007 (in french).
- [19] A. Bounceur, S. Mir, L. Rolíndez and E. Simeu. CAT platform for analogue and mixed-signal test evaluation and optimization. Chapter in IFIP International Federation for Information Processing, Vol. 249, VLSI-SoC: Research trends in VLSI and Systems on Chip. G. De Micheli, S. Mir, R. Reis (Eds.), Springer, 2007, pp. 281-300.
- [20] IBM ILOG CPLEX, Software available at <http://www-01.ibm.com/software/integration/optimization/>