Minimizing Test Frequencies for Linear Analog Circuits: New Models and Efficient Solution Methods

Mohand Bentobache[1,2(✉)], Ahcène Bounceur[3], Reinhardt Euler[3], Salvador Mir[4,5(✉)], and Yann Kieffer[6(✉)]

[1] LAMOS Laboratory, University of Bejaia, 06000 Bejaia, Algeria
mbentobache@yahoo.com
[2] LMPA Laboratory, University of Laghouat, 03000 Laghouat, Algeria
m.bentobache@lagh-univ.dz
[3] Lab-STICC Laboratory, University of Brest, 20, Avenue Victor Le Gorgeu, 29238 Brest, France
{Ahcene.Bounceur,Reinhardt.Euler}@univ-brest.fr
[4] University of Grenoble Alpes, TIMA, 38000 Grenoble, France
Salvador.Mir@imag.fr
[5] CNRS, TIMA, 38000 Grenoble, France
[6] LCIS Laboratory, University of Grenoble Alpes, 26900 Valence, France
Yann.Kieffer@esisar.grenoble-inp.fr

Abstract. This work presents new approaches to minimize the number of test frequencies for linear analog circuits. The cases of single and multiple fault detection regions for multiple test measures are considered. We first address the case when the injected faults have a single detection region in the frequency band. We show that the problem can be formulated as a set covering problem with a matrix having the consecutive-ones property for which the network simplex algorithm turns out to be very efficient. A second approach consists in modeling the problem by means of an interval graph, leading to its solution with a specific polynomial-time algorithm. A case-study of a biquadratic filter is presented for illustration purposes. Numerical simulations demonstrate that the two different approaches solve the optimization problem very fast. Finally, the optimization problems arising from multiple detection regions are modeled and solution approaches are discussed.

Keywords: Set covering problem · Consecutive-ones property · Analog circuit testing · Linear programming · Interval graphs

1 Introduction

Testing of analog circuits to verify their functionality is a time consuming and expensive task. In order to reduce costs, testing for the presence of faults that affect the device structure leads to more structured and cheaper test sets. Although the number of faults to be tested may indeed be very large, a reduced

set of tests that are simple to apply may be sufficient to achieve high fault coverage values. There is today a pressing need from the Semiconductors Industry to provide these fault coverage measures and optimized tests, especially in the context of SoC (System-on-Chip) 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 work 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.

This work is organized as follows: in Sect. 2, the mathematical formulation of the Set Covering Problem (SCP) is reviewed and some definitions are given. In Sect. 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 Sect. 4, we present two approaches for solving the problem: the Linear Programming (LP) approach and the interval graph approach, and a numerical example is given for illustration purposes. In Sect. 5, we present a mathematical formulation of the problem to minimize the number of test measures necessary for detecting the faults of an analog circuit. In Sect. 6, we present a case-study of testing a biquadratic filter. In Sect. 7, we carry out a large-scale numerical study in order to compare both approaches and to evaluate their time complexity. Finally, Sect. 8 concludes the work and provides some perspectives.

2   The Set Covering Problem

In this section, we present the set covering problem, we cite the existing methods for solving this problem and some of its applications. Finally, we present the corresponding mathematical model and some definitions.

2.1   Overview

The Set Covering Problem is one of the most important models in combinatorial optimization. Indeed, a wide range of real-world problems can be 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.

2.2   Some Definitions

Let $M = \{1, \ldots, m\}$ and $N = \{1, \ldots, n\}$ 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 $c_j, j \in N$, may represent the cost of column j. 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:

$$\min \ z = c^T x, \tag{1a}$$

$$\text{subject to } Ax \geq \mathbf{1}_m, \ x \in \{0, 1\}^n, \tag{1b}$$

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 determi-
nant 0, 1, or -1.*

3   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.

3.1   Mathematical Formulation of the Problem as an SCP-C1P

Consider a given linear analog circuit **C**. Let $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ be the set
of all 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 $\mathcal{R}_i = \{R_{i1}, R_{i2}, \ldots, R_{in_i}\}$ 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, we sort in increasing order the bounds of the different detection
regions, we find the vector of frequencies:

$$f = (f_0, f_1, \ldots, f_n), \text{ with } f_0 < f_1 < f_2 < \cdots < f_{n-1} < f_n. \qquad (2)$$

Then, we compute a set of n frequency intervals $\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ as follows:

$$I_1 = [f_0, f_1[, \; I_2 = [f_1, f_2[, \ldots, \; I_n = [f_{n-1}, f_n[. \qquad (3)$$

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

$$a_{ij} = \begin{cases} 1, \text{ if } \exists k \in \{1, \ldots, n_i\} : I_j \subseteq R_{ik}; \\ 0, \text{ otherwise.} \end{cases} \qquad (4)$$

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 unicost set covering problem: for each $S \subseteq \mathcal{I}$ and each frequency interval I_j, $j \in \{1, \ldots, n\}$, we define a binary variable as follows:

$$x_j^S = \begin{cases} 1, \text{ if } I_j \text{ belongs to the set } S; \\ 0, \text{ otherwise.} \end{cases} \tag{5}$$

Now, if $x = (x_j, j = 1, \ldots, n)$ is any vector of binary variables, then the mathematical model corresponding to this problem will be given by:

$$\min \ r = \mathbf{1}_n^T x, \tag{6a}$$

$$\text{s.t. } Ax \geq \mathbf{1}_m, \ x \in \{0,1\}^n. \tag{6b}$$

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

Proposition 1. *If $n_i = 1$, for $i = 1, \ldots m$, 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 $R_{i1} = [\alpha_i, \beta_i]$ be the unique detection region of fault i. By construction, R_{i1} can be written as a union of k consecutive intervals $I_{j_1}, I_{j_2}, \ldots, I_{j_k}$, where α_i is the lower bound of I_{j_1} and β_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 $r = 1, \ldots, k$, the detection regions correspond to the rows of A, and $I_{j_r} \subseteq R_{i1}$, i.e., $a_{ij_r} = 1$ for $r = 1, \ldots, k$.

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 (6) is a unicost set covering problem with a coefficient matrix having the strong C1P (SCP-C1P). More precisely, it is given by:

$$\min \ z = \sum_{j=1}^n x_j, \tag{7a}$$

$$\text{s.t. } \sum_{j=lx(i)}^{rx(i)} x_j \geq 1, \ i \in M, \ x_j \in \{0,1\}, j \in N, \tag{7b}$$

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:

$$\min \ z = \sum_{j=1}^{n} x_j, \tag{8a}$$

$$\text{s.t.} \ \sum_{j=lx(i)}^{rx(i)} x_j \geq 1, \ i \in M, \ x_j \geq 0, j \in N. \tag{8b}$$

3.2 Graph Formulation of the Problem

Previously, we have formulated the problem of minimizing the number of 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:

$$F = \{[a_i, b_i], \ i \in M\}, E_F = \{F_i F_j :]a_i, b_i[\cap]a_j, b_j[\neq \emptyset\}. \tag{9}$$

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

4 Approaches for Solving the Minimization Problem of Frequency Intervals

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

4.1 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 (8) represents a cover for the SCP-C1P (7). Therefore, we can find an optimal solution using LP algorithms such as the primal or dual simplex method [14], the support method [15], the hybrid direction algorithm [19], 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 [20] for the variables of problem (8): we introduce the variables $y_j, \ j = 1, \ldots, n+1$ such that $x_j = -y_j + y_{j+1}, \ j = 1, \ldots, n$. Hence, we obtain the following equivalent LP problem:

$$\min \ z = -y_1 + y_{n+1}, \tag{10a}$$
$$\text{s.t.} \ -y_{lx(i)} + y_{rx(i)+1} \geq 1, i \in M, \tag{10b}$$
$$-y_j + y_{j+1} \geq 0, \ j \in N. \tag{10c}$$

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

$$\min \ w = -\sum_{j=1}^{m} v_j, \tag{11a}$$
$$\text{s.t.} \ \sum_{j=1}^{m+n} a'_{1j} v_j = -1, \tag{11b}$$
$$\sum_{j=1}^{m+n} a'_{ij} v_j = 0, \ i = 2, \ldots, n, \tag{11c}$$
$$\sum_{j=1}^{m+n} a'_{(n+1)j} v_j = 1, \ v_j \geq 0, \ j = 1, \ldots, m+n. \tag{11d}$$

where $A' = (a'_{ij}, i = 1, \ldots, n+1, j = 1, \ldots, m+n)$, and A' has exactly one 1 and one -1 in each column. Remark that the LP problem (11) is a min-cost network flow problem. Thus, it can be solved by the network simplex algorithm. 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 (4);

Step 3. Make the Veinott-Wagner transformations. Let V be the constraint matrix of the LP problem (10);

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 (11):
$A' = V^T, \ b' = (-1, 0, \ldots, 0, 1)^T, \ c' = \begin{pmatrix} -\mathbf{1}_m \\ \mathbf{0}_n \end{pmatrix};$

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

4.2   Interval Graph Approach

In order to solve the problem using the interval graph G defined by (9), we suggest the following algorithm:

Algorithm 1.
● Sort the vertices F_i, $i = \overline{1, m}$ by increasing order of their
upper bounds b_i;
● Compute the frequencies f_k, $k = \overline{0, n}$ with (2);
● Set $S = \emptyset$, label$(i) = 0, i = \overline{1, m}$;
● For $i = 1$ to m
    ● If label$(i) = 0$, then
        ● Find the index k such that $b_i = f_k$ and set
          $S = S \cup \{[f_{k-1}, f_k[\}$;
        ● For $j = i + 1$ to m
          If F_j is adjacent to F_i in G, then set $label(j) = 1$;
          endif
          endfor
      endif
  endfor

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 $a_j < b_i \le b_j$".

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.

4.3   Numerical Example

Example 1. We consider an analog circuit with five faults. The detection regions of these faults are: $R_1 = [1, 80]$, $R_2 = [160, 1700]$, $R_3 = [1, 1400]$, $R_4 = [1400, 2000]$ and $R_5 = [1000, 1500]$. Let us compute the minimal-cardinality set of frequency intervals necessary to detect all the five faults with the two approaches.

LP Approach: First, we compute the different frequency intervals $I_j, j = 1, \ldots,$ n: we sort the different bounds of the detection regions in increasing order, we get the vector $f = (f_i, i = 0, \ldots, 7) = (1, 80, 160, 1000, 1400, 1500, 1700, 2000)$. Thus, the different frequency intervals are computed as follows: $I_1 = [1, 80[$, $I_2 = [80, 160[$, $I_3 = [160, 1000[$, $I_4 = [1000, 1400[$, $I_5 = [1400, 1500[$, $I_6 = [1500, 1700[$ and $I_7 = [1700, 2000[$. The question is then: among all these seven intervals, which one must be included in a minimal-cardinality set? In order to answer this question, we solve the following ILP problem:

$$\min z = \mathbf{1_n}^T x, \tag{12a}$$

$$\text{s.t. } Ax \geq b, \; x \in \{0,1\}^n, \tag{12b}$$

where $x^T = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, $b^T = (1,1,1,1,1)$ and

$$A = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,1\,1\,1\,0 \\ 1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,1\,1\,1 \\ 0\,0\,0\,1\,1\,0\,0 \end{pmatrix}. \tag{13}$$

It is easy to remark that the constraint matrix has the consecutive-ones property. Using the Veinott-Wagner transformations $(x_j = -y_j + y_{j+1}, \; j = 1, 2, \ldots, 7)$, we get the following equivalent LP problem:

$$\begin{aligned} \min z &= (c')^T y \\ \text{s.t. } Vy &\geq b', \end{aligned} \tag{14}$$

where $y \in \mathbb{R}^8$, $c' = (-1, 0, 0, 0, 0, 0, 0, 1)^T$, $b' = (1,1,1,1,1,0,0,0,0,0,0,0)^T$ and

$$V = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}. \tag{15}$$

The dual problem of the LP problem (14) is

$$\begin{aligned} \min w &= -(b')^T v \\ \text{s.t. } V^T v &= c', \; v \geq 0, \end{aligned} \tag{16}$$

where $v \in \mathbb{R}^{12}$. Remark that the LP problem (16) is a min-cost network flow problem. The primal and dual optimal solutions obtained by the network simplex algorithm of CPLEX are:

$$v^* = (0,0,1,1,0,0,0,0,0,0,0,0)^T, \; y^* = (0,1,1,1,1,2,2,2)^T. \tag{17}$$

Hence $x^* = (1,0,0,0,1,0,0)^T$, which means that a minimal-cardinality set of frequency intervals is $S = \{I_1, I_5\} = \{[1, 80[, \; [1400, 1500[\}$.

Interval Graph Approach: Let $G = (F, E_F)$ be the graph defined by (9). First, we sort the vertices of the graph G in increasing order of their upper

bounds, we get:

$$F_1 = [a_1, b_1[= [1, 80[, \ F_2 = [a_2, b_2[= [1, 1400[, \ F_3 = [a_3, b_3[= [1000, 1500[,$$

$$F_4 = [a_4, b_4[= [160, 1700[\text{ and } F_5 = [a_5, b_5[= [1400, 2000[.$$

The vector of frequencies computed by (2) is then

$$f = (f_0, f_1, \ldots, f_7) = (1, 80, 160, 1000, 1400, 1500, 1700, 2000).$$

The different iterations of the interval graph algorithm are as follows:

We set $S = \emptyset$ and label$(i) = 0$, $i = \overline{1, 5}$.

For $i = 1$, $b_1 = 80$, $f_k = f_1 = 80$, $S = \{[f_0, f_1[\} = \{[1, 80[\}$.   The unique adjacent vertex of F_1 is F_2 (the edge $F_1 F_2 \in E_F$ because $a_2 < b_1 < b_2$), so we set label$(2) = 1$.

For $i = 2$, we have label$(2) = 1$, so we pass to $i = 3$.

For $i = 3$, $b_3 = 1500$, $f_k = f_5$, $S = \{[f_0, f_1[, [f_4, f_5[\} = \{[1, 80[, [1400, 1500[\}$. The adjacent vertices of F_3 are F_4 and F_5, so label$(4) = 1$ and label$(5) = 1$.

For $i = 4$, we have label$(4) = 1$, so we pass to $i = 5$.

For $i = 5$, we have label$(5) = 1$. Since $i = m$, a minimal-cardinality set of frequency intervals is $S = \{[1, 80[, \ [1400, 1500[\}$.

4.4   Case of Multiple Detection Regions

Let us illustrate this case by the following numerical example:

Example 2.

Faults	Detection regions
F_1	$R_{11} = [1, 80]$
F_2	$R_{21} = [160, 200]$, $R_{22} = [300, 1700]$
F_3	$R_{31} = [1, 1400]$
F_4	$R_{41} = [1400, 2000]$
F_5	$R_{51} = [1000, 1500]$

The vector of frequencies computed by (2) is then

$$f = (f_0, f_1, \ldots, f_9) = (1, 80, 160, 200, 300, 1000, 1400, 1500, 1700, 2000).$$

The frequency intervals are:

$$I_1 = [1, 80[, \ I_2 = [80, 160[, \ I_3 = [160, 200[, \ I_4 = [200, 300[, \ I_5 = [300, 1000[,$$

$$I_6 = [1000, 1400[, \ I_7 = [1400, 1500[, \ I_8 = [1500, 1700[, \ I_9 = [1700, 2000[.$$

The constraint matrix of the SCP is then given by

$$A = \begin{pmatrix} 1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0\,1\,1\,1\,1\,0 \\ 1\,1\,1\,1\,1\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1\,1\,1 \\ 0\,0\,0\,0\,0\,1\,1\,0\,0 \end{pmatrix}. \tag{18}$$

Let us remark that this matrix loses the consecutive-ones property because the second row does not have consecutive ones. Therefore, we cannot apply the techniques suggested above to solve it. If this case occurs in practice, the branch-and-cut algorithm of CPLEX can be applied to solve the ILP problem. For this example, CPLEX gives us the optimal solution: $x^* = (1, 0, 0, 0, 0, 0, 1, 0, 0)^T$. So a minimal-cardinality set of frequency intervals is

$$S = \{I_1, I_7\} = \{[1, 80[, \ [1400, 1500[\}. \tag{19}$$

Remark 4. Although the constraint matrix of the previous SCP does not possess the consecutive-ones property, solving the LP relaxation gives us the optimal solution $x^* = (1, 0, 0, 0, 0, 0, 1, 0, 0)^T$. This means that for this type of problems the LP relaxation can give a good approximate solution for the optimal one (good upper bound for the optimal value). That is why we expect the branch-and-cut algorithm of CPLEX (or any other solver) to produce an optimal solution in a few iterations in case that the optimal solution of the LP-relaxation is not optimal for the SCP. Further numerical experiments have to be carried out in order to confirm this assertion.

5   Minimization of the Number of Test Measures Necessary for Detecting All the Faults of an Analog Circuit

Previously, we have assumed that only one test measure is used in the testing process. However, in practice, often we use several test measures. So if k is the number of the used test measures, we need to solve k SCPs for solving the problem of frequency interval minimization. Therefore, solving the problem of minimization of test measures beforehand can dramatically reduce the CPU time of solving the whole optimization process.

Consider a given analog circuit $\mathcal{C}$. Let $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ be the set of all faults which can occur in $\mathcal{C}$. Let $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ be the set of k test measures to be used in the fault detection process. We denote by $H = (h_{ij}, i = 1, \ldots, m, \ j = 1, \ldots, k)$ the Fault-Test-Measure (FTM) matrix which is defined as follows:

$$h_{ij} = \begin{cases} 1, \text{ if the test measure } T_j \text{ detects the fault } F_i; \\ 0, \text{ otherwise.} \end{cases} \tag{20}$$

The problem consists in finding the minimal-cardinality set of test measures which can detect all the considered faults. This problem can also be formulated as a unicost set covering problem: for each test measure T_j, $j = 1, 2 \ldots, k$ and any set $S \subseteq \mathcal{T}$, we define a binary variable as follows:

$$x_j^S = \begin{cases} 1, \text{ if the test measure } T_j \in S; \\ 0, \text{ otherwise.} \end{cases} \tag{21}$$

Hence, the mathematical model corresponding to this problem is given by:

$$\begin{aligned} \min p &= \mathbf{1}_k^T x, \\ \text{s.t. } Hx &\geq \mathbf{1}_m, \ x \in \{0,1\}^k. \end{aligned} \tag{22}$$

The optimal solution for this integer linear programming problem constitutes a desired minimal-cardinality set of test measures which detect all the faults. Note that, in general, the coefficient matrix of this problem does not have a specific structure, so the branch-and-cut algorithm of CPLEX can be used to solve this problem.

Once a minimal-cardinality set of test measures determined, we can solve the problem of frequency interval minimization for each test measure in this set.

The global optimization scheme for testing a linear analog circuit with multiple test measures is shown in Fig. 1.

6   Case-Study

To illustrate our approach, similar to [3], we will now present a case-study biquadratic filter as shown in Fig. 2. 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 $1 \, \Omega$ 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 $T_1, T_2, \ldots, T_6$ and the faults by $F_1, F_2, \ldots, F_{16}$. Figure 3 shows the frequency behavior of test measure T_1 for some of these faults. The detection regions of each fault F_i, $i = 1, 2, \ldots, 16$ using test measures T_j, $j = 1, 2, \ldots, 6$ are computed using the fault simulator developed in [21,22]. Note, that the simulation results obtained in [21] 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 number of test measures necessary for the detection of the sixteen faults. Following simulation results, test measure T_1 detects the faults: $F_1, F_2, \ldots, F_8$; T_3 detects the faults $F_9, F_{10}, F_{11}, F_{12}$ and
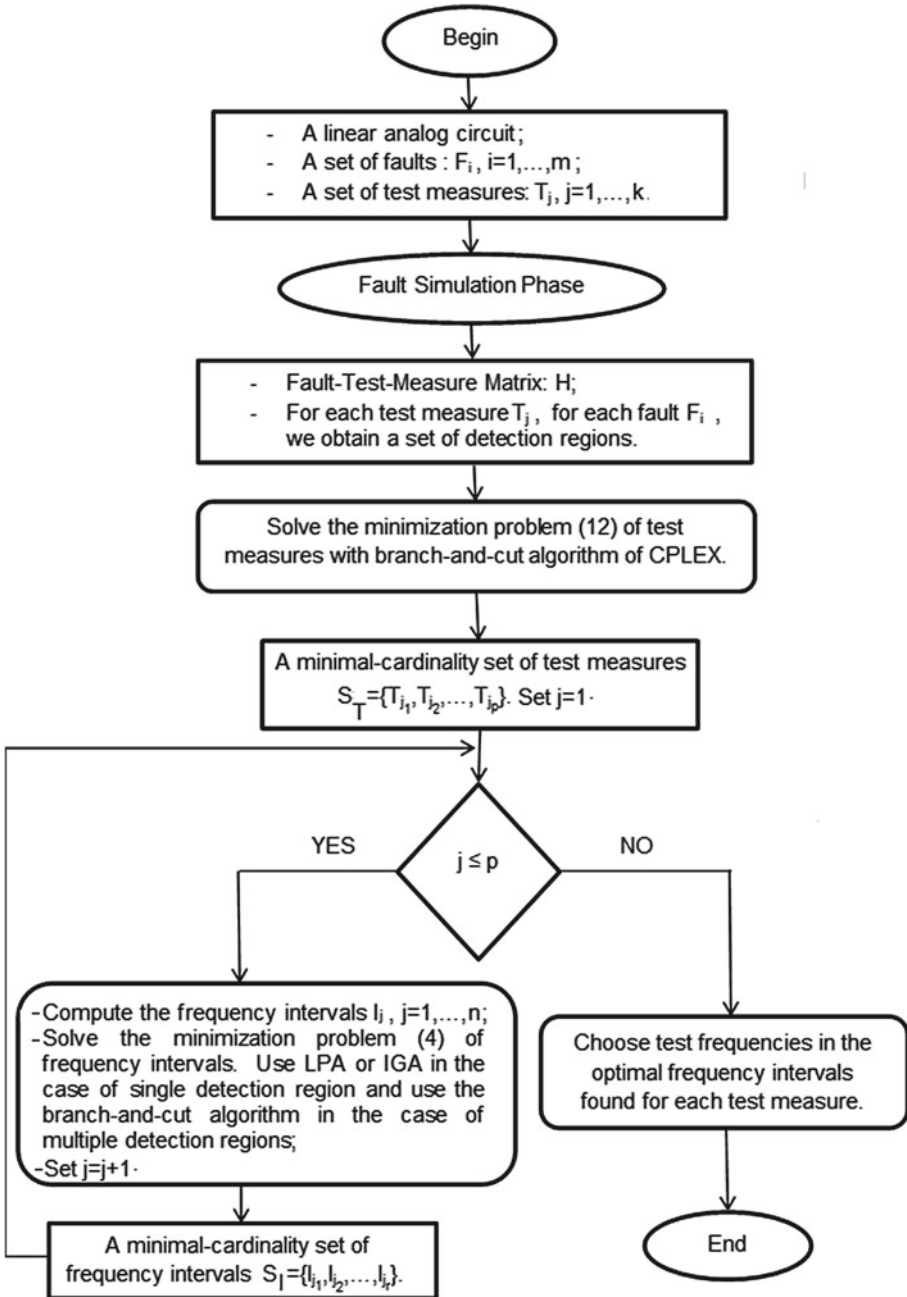
Fig. 1. Global optimization scheme for testing analog circuits with multiple test measures
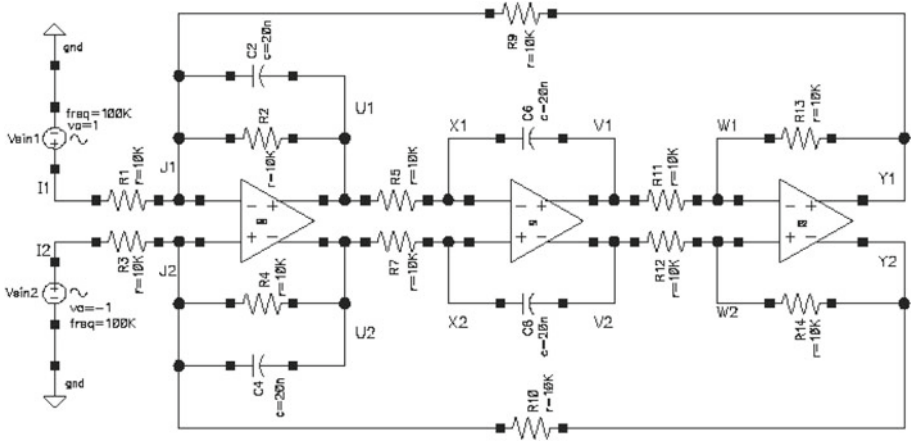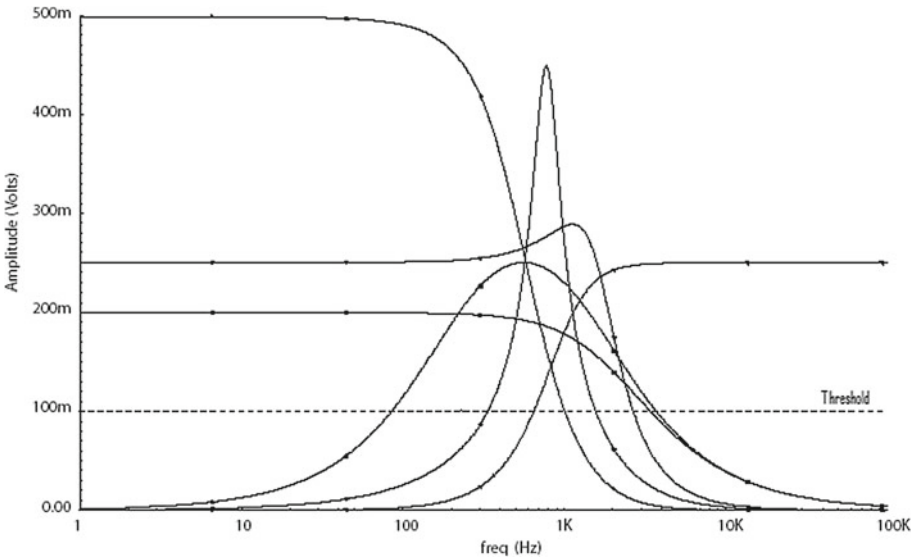
Fig. 2. Biquadratic filter

Fig. 3. Output signals using test measure T_1

T_5 detects the faults F_{13}, F_{14}, F_{15}, F_{16}. So the FTM matrix corresponding to the minimization problem of test measures is

$$H^T = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \end{pmatrix}. \tag{23}$$

Observe, that since test measures T_2, T_4 and T_6 do not detect any faults, the matrix H contains only three columns which correspond to the test measures

T_1, T_3 and T_5. Without solving the SCP problem (22), it is clear that a minimal-cardinality set of test measures which detects all the faults is $S = \{T_1, T_3, T_5\}$. Now, that a minimal set of test measures is at hand, we can proceed to solve the problem of minimization of test frequencies for each test measure belonging to this set.

First, we start by minimizing the frequency intervals under the test measure T_1: the faults detected using test measure T_1 are $F_1, F_2, \ldots, F_8$. 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 vector of frequencies:

$$f = (f_0, f_1, \ldots, f_9) = (1, \ 85, \ 336, \ 647, \ 1014, \ 1566, \ 2685, \ 3442, \ 3732, \ 10^5).$$

Therefore, the frequency intervals are

$$I_1 = [1, 85[, \ I_2 = [85, 336[, \ I_3 = [336, 647[, \ I_4 = [647, 1014[, \ I_5 = [1014, 1566[,$$

$$I_6 = [1566, 2685[, \ I_7 = [2685, 3442[, \ I_8 = [3442, 3732[\text{ and } I_9 = [3732, 10^5[.$$

The constraint matrix of the SCP (6) is then:

$$A = \begin{pmatrix} 1\,1\,1\,1\,1\,1\,1\,1\,1 \\ 0\,1\,1\,1\,1\,1\,1\,1\,0 \\ 0\,1\,1\,1\,1\,1\,1\,1\,0 \\ 1\,1\,1\,1\,1\,1\,0\,0\,0 \\ 1\,1\,1\,1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,1\,1\,0\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,1\,1\,1\,1\,1 \end{pmatrix}, \tag{24}$$

where each row i, $i = 1, \ldots, 8$, corresponds to the detection region R_{i1} and each column j, $j = 1, \ldots, 9$, 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 (6) can be obtained using the network simplex algorithm. 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 $[647\,Hz, 1014\,Hz[$ to detect all the faults $F_1, F_2, \ldots, F_8$.

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

$$A = \begin{pmatrix} 0\ 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 0\ 0 \\ 1\ 1\ 0\ 0\ 0 \\ 0\ 1\ 1\ 1\ 0 \end{pmatrix}. \tag{25}$$

Hence, for the test measure T_3, the solution of the SCP (6) is obvious, that is, a single test frequency belonging to the interval $I_2' = [159\,Hz, 1739\,Hz[$ 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

$$A = \begin{pmatrix} 1\ 1\ 1\ 1 \\ 1\ 1\ 0\ 0 \\ 1\ 0\ 0\ 0 \\ 1\ 1\ 1\ 0 \end{pmatrix}. \tag{26}$$

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 $I_1'' = [1\,Hz, 1412\,Hz[$, and a single test frequency belonging to this interval is required.

7   Numerical Experiments

In order to compare the efficiency of the two approaches presented in Sect. 4 (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 @ 2.40 GHz machine with 4 GB of RAM.

We have generated 60 problems with number of faults

$$m = 500,000; 600,000; 700,000; 800,000; 900,000; 1,000,000. \tag{27}$$

The detection region bounds are generated in the interval $[1\,Hz, 10^5\,Hz]$. 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 [23] and the suggested interval graph algorithm (IGA). The CPU time of the two approaches IGA and LPA are reported in Table 1.

Let $[a_i, b_i]$ be the detection regions of the faults F_i, $i = 1, \ldots, m$. 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 f be the vector of these sorted bounds;
- delete duplicate elements from f, and let $f = (f_0, f_1, \ldots, f_n)$;
- set $\mathcal{I} = \{I_j = [f_{j-1}, f_j[, \ j = 1, \ldots, n\}$;

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

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

Step 3. Make the Veinott-Wagner transformations: compute the constraint matrix V of the LP problem (10) as follows:

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

$$V_{ij} = \begin{cases} -1, & \text{if } j = lx(i); \\ 1, & \text{if } j = rx(i) + 1; \\ 0, & \text{otherwise;} \end{cases} \tag{28}$$

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

$$V_{ij} = \begin{cases} -1, & \text{if } j = i - m; \\ 1, & \text{if } j = i - m + 1; \\ 0, & \text{otherwise;} \end{cases} \tag{29}$$

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 (11): $A' = V^T$, $b' = (-1, 0, \ldots, 0, 1)^T$, $c' = \begin{pmatrix} -\mathbf{1}_m \\ \mathbf{0}_n \end{pmatrix}$;

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 1.

Table 1. CPU time of the different approaches

m	OptVal	IGA	LPA	CPU 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**
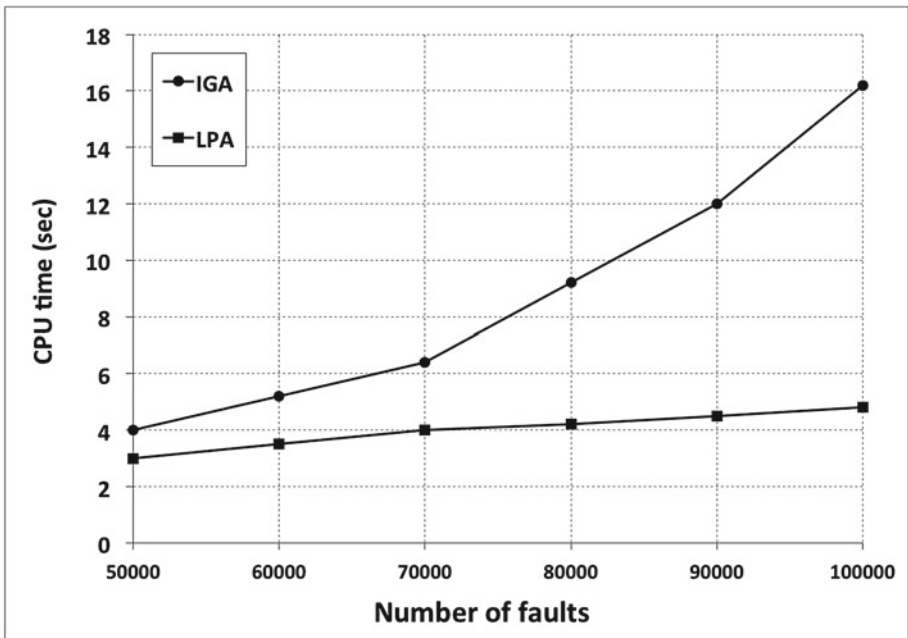
Fig. 4. CPU time of the two approaches: IGA and LPA

The average optimal values found by the two approaches IGA and LPA for the different test problems are shown in column OptVal. We compute the ratios of the CPU time of IGA over LPA. These ratios are shown in column CPU Ratio. Finally, the CPU time of the two approaches are plotted in Fig. 4. 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 Fig. 4 indicate that transforming the problem into a min-cost network flow problem and solving it with the network simplex algorithm 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 simple 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.

8   Conclusion

In this work, we have formulated as an SCP the problem of minimizing the number of test frequencies necessary to detect 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 algorithm 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 simple and it is extremely fast with CPU times of a few tenths of seconds even for large-scale problems. Moreover, the optimization problem associated with multiple detection regions and multiple test measures is studied. Future work will focus on developing a branch-and-cut algorithm for solving efficiently the set covering problem corresponding to the case of multiple detection regions.

References

1. Alippi, C., Catelani, M., Fort, A., Mugnaini, M.: Automated selection of test frequencies for fault diagnosis in analog electronic circuits. IEEE Trans. Instrum. Meas. **54**(3), 1033–1044 (2005)
2. Grasso, F., Luchetta, A., Manetti, S., Piccirilli, M.-C.: A method for the automatic selection of test frequencies in analog fault diagnosis. IEEE Trans. Instrum. Meas. **56**(6), 2322–2329 (2007)
3. Mir, S., Lubaszewski, M., Courtois, B.: Fault-based ATPG for linear analog circuits with minimal size multifrequency test sets. J. Electron. Test. Theory Appl. **9**, 43–57 (1996)
4. Balas, E., Carrera, M.C.: A dynamic Subgradient-based Branch-and-bound procedure for set covering. Oper. Res. **44**, 875–890 (1996)
5. Chvatal, V.: A greedy heuristic for the set-covering problem. Math. Oper. Res. **4**(3), 233–235 (1979)
6. Beasley, J.E.: A lagrangian heuristic for set covering problems. Nav. Res. Logistics **37**, 151–164 (1990)
7. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. Eur. J. Oper. Res. **94**, 392–404 (1996)
8. Beasley, J.E.: OR-library: distributing test problems by electronic mail. J. Oper. Res. Soc. **41**, 1069–1072 (1990)

9. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco (1979)
10. Dom, M.: Recognition, Generation, and Application of Binary Matrices with the Consecutive-Ones Property. Ph.D. thesis, Institut für Informatik, Friedrich-Schiller Universität Jena, Germany (2008)
11. Matoušek, J., Gärtner, B.: Understanding and Using Linear Programming. Springer, Berlin (2007)
12. Ghouila-Houri, A.: Caractérisation des matrices totalement unimodulaires. CR Acad. Sci. Paris **254**, 1192–1194 (1962)
13. Hoffman, A.J., Kruskal, J.B.: Integral boundary points of convex polyhedra. In: Kuhn, H.W., Tucker, A.W. (eds.) Linear Inequalities and Related Systems, pp. 223–246. Princeton University Press, Princeton (1956)
14. Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press, Princeton (1963)
15. Bentobache, M., Bibi, M.O.: A two-phase support method for solving linear programs: Numerical experiments. Math. Probl. Eng., **2012**, Article ID 482193, 28 (2012). doi:10.1155/2012/482193
16. Bentobache, M., Bounceur, A., Euler, R., Kieffer, Y., Mir, S.: New techniques for selecting test frequencies for linear analog circuits. In: VLSI-SoC 2013, Istanbul, Turkey, pp. 93–98, 07–09 October 2013
17. Bentobache, M., Bounceur, A., Euler, R., Kieffer, Y., Mir, S.: Efficient minimization of test frequencies for linear analog circuits. In: 18th IEEE European Test Symposium, ETS 2013, Avignon, France, p. 1, 27–30 May 2013
18. Bentobache, M., Bounceur, A., Euler, R.: Une application efficace du problème de recouvrement au test de circuits analogiques. In: ROADEF 2012, University of Angers, France, pp. 424–425, 11–13, April 2012
19. Bibi, M.O., Bentobache, M.: A hybrid direction algorithm for solving linear programs. Int. J. Comput. Math. **92**(1), 201–216 (2015)
20. Veinott, A.F., Wagner, H.M.: Optimal capacity scheduling. Oper. Res. **10**, 518–547 (1962)
21. Bounceur, A.: CAO Platform for mixed circuit testing. Ph.D. thesis, Grenoble INP (2007) (in french)
22. Bounceur, A., Mir, S., Rolíndez, L., Simeu, E.: CAT platform for analogue and mixed-signal test evaluation and optimization. In: De Micheli, G., Mir, S., Reis, R. (eds.) VLSI-SoC: Research Trends in VLSI and Systems on Chip. IFIP, vol. 249, pp. 281–300. Springer, Boston (2007)
23. CPLEX, Software available at http://www-01.ibm.com/software/integration/optimization/