Acta Cybernetica 14 (1999) 229-238.

Test Suite Reduction in Conformance Testing

Tibor Csöndes * Sarolta Dibuz * Balázs Kotnyek †

Abstract

Conformance testing is based on a test suite. Standardization committees release standard test suites, which consist of hundreds of test cases. The main problem of conformance testing is that we do not have enough time to execute them all. Therefore, test selection is required to maximize the test coverage. In our earlier papers [6,7] we outlined a new method of selecting an optimal test suite which can detect the errors with better probability and reduce the time of testing. In this paper we will expound the mathematical optimization method for test suite optimization based on cost and test coverage, and we will apply this method to an ISDN protocol.

1 Introduction

The main aim of conformance testing is to check whether the protocol implementation conforms to the standard. The procedure of conformance testing as well as the protocols are standardized [1]. The two main terms of testing are the test purpose (TP) and the abstract test suite (ATS). The test purpose is the description of the well defined objective of testing to focus on a single conformance requirement or a set of related conformance requirements. The ATS consists of several test cases (TC) created to test one or more TPs. In real-life conformance testing, the testers choose some of the TPs and execute all the TCs that are related to the chosen set of TPs. The challenge in conformance testing is this selection, choosing this set so that the coverage, the fault detection capability, be maximal. Of course, the best selection is when we choose all the TPs or - what means the same - all the TCs. The problem that arises here is the time limitation. Usually we do not have enough time to do this, so we can only execute some of the TCs.

The existing approach of handling this problem is the test generation. The goal of such procedures is briefly to generate optimal ATSs from the protocol specification , i.e. that contain as few parallelisms as possible so they can be entirely executed within the time limit. The theoretical background of this kind of optimization is the finite state machine (FSM). If a FSM model of the protocol is already given, there are several algorithms for generating good or better test suites (Transition Tour, Unique Input/Output method, Distinguishing Sequence) [4]. We

*Conformance Center, Ericsson Ltd.,Laborc u. 1., H-1037 Budapest, Hungary
†Eötvös Loránd University, Budapest

need this FSM model, however and real life protocols are so complicated that it is not possible to create their usable FSM model.

Our approach, which we outlined in our earlier papers [6,7], is based on practice. We suppose that we are given an ATS and we cannot generate any new TCs. That is what test laboratories do, they use only those test cases which are provided by the standards. By now, the selection of TCs from ATS is based on the subjective decision of the test laboratory. Our aim is to create a theoretical background of such selection. We found that mathematical optimization could be a suitable one.

The rest of the paper is organized as follows: first, we introduce a model of conformance testing mathematics could operate on. In Section 3 we describe this operation, in Section 4 we present how this method could be applied to an existing protocol the ISDN DSS1 Layer 2 protocol[2]. We chose this protocol because it is widely used, well known, and its ATS exists.

2 The model

In our model, we create a mathematically formulated relation between the test cases and test purposes by introducing the purpose-test incidence matrix. The purposes are placed in the rows and the tests in the columns. As a result we get an A matrix of size $k \times n$ where $P_1, \ldots, P_k$ are the purposes and $T_1, \ldots, T_n$ are the test cases defined in the ATS (Figure 1). The j^{th} element in the i^{th} row is 1 if, and only if T_j is necessary to check P_i, otherwise it is 0. In other words, if we want to check a purpose (e.g. P_i) completely, we have to execute all test cases having 1 in the row of the purpose (namely in the i^{th} row). Let us introduce the number b_i designating the number of such test cases and let $b = (b_1, \ldots, b_k)^T$ be the vector made of these numbers.

$$\begin{array}{c|c|c|c|c|} & T_1 & T_2 & \cdots & T_n \\ \hline P_1 & 1 & 0 & \cdots & 1 \\ \hline P_2 & 1 & 1 & \cdots & 0 \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline P_k & 0 & 1 & \cdots & 0 \\ \hline \end{array} = A$$

Figure 1: Purpose-test incidence matrix

There are protocols the ATSs of which define one-to-one connection between the test cases and the purposes, hence their incidence matrix is diagonal. There exist, however, ATSs the matrices of which are not diagonal, thus there are more than one necessary test cases to check a test purpose. In this paper, we are dealing with these kind of protocols.

Let us assign the value $cov(P_i)$, $(i = 1, \ldots, k)$ to every purpose describing its coverage. This value may be obtained from a theoretical consideration or we can simply mark the priority of the purpose with it. Similarly, let us designate the cost of test T_j with $c(T_j)$, $(j = 1, \ldots, n)$. This cost function can be defined to represent the resources (time) required to execute the test case. The cost of a set of test cases can be defined simply as the sum of the cost of the individual test cases in the set. Let us do the same with the coverage of a set of purposes.

Let us introduce the increasing functions $f_i : \{0, 1, \ldots, b_i\} \to [0, cov(P_i)]$, $(i = 1, \ldots, k)$ describing the coverage we get if we execute m tests among the tests that correspond to P_i $(m = 0, 1, \ldots, b_i)$. Of course $f_i(0) = 0$ and $f_i(b_i) = cov(P_i)$ for all $i = 1, 2, \ldots, k$. The different models differ from each other in choosing functions f_i. We introduce three models below.

1. **Linear model:** The coverage is in direct proportion to the number of executed tests i.e.

$$f_i(m) = \frac{m}{b_i} cov(P_i) \qquad m = 0, \ldots, b_i$$

2. **"All or nothing" model:** We only consider the purpose being checked when we executed all the necessary tests.

$$f_i(0) = f_i(1) = \ldots = f_i(b_i - 1) = 0 \quad \text{and} \quad f_i(b_i) = cov(P_i)$$

3. **"One is enough" model:** If only one test is executed among the ones that correspond to P_i, we get the whole coverage.

$$f_i(0) = 0 \quad \text{and} \quad f_i(1) = \ldots = f_i(b_i) = cov(P_i)$$

3 Optimization

We introduce two possible optimization problems. In the first one our aim is to select a test set from the test suite with **minimal cost** supposing a constraint bounding the coverage from below. Let $x \in \{0, 1\}^n$ be the decision vector, so $x_j = 1$ if T_j is executed and $x = 0$ if it is not. The minimization can be formalized in the following manner:

$$\begin{aligned} \min \quad & cx \\ \text{subject to} \quad & \sum_{i=1}^{k} f_i(a_i x) \geq K \\ & x \in \{0, 1\}^n \end{aligned} \tag{1}$$

where $c = (c(T_1), \ldots, c(T_n))$ is the cost vector, K is the lower bound for the coverage, and a_i is the i^{th} row of the matrix A. Furthermore, let $v = (cov(P_1), \ldots, cov(P_k))$ be the coverage vector. Let us see how this formula looks like in the case of the three introduced models.

1. **Linear model**

$$\sum_{i=1}^{k} f_i(a_i x) = \sum_{i=1}^{k} \frac{a_i x}{b_i} cov(P_i) = \left(\sum_{i=1}^{k} \frac{a_i cov(P_i)}{b_i} \right) x$$

Thus (1) turns into a binary minimization with a single linear constraint:

$$\begin{array}{rl} \min & cx \\ \text{subject to} & \left(\sum_{i=1}^{k} \frac{a_i cov(P_i)}{b_i} \right) x \geq K \\ & x \in \{0,1\}^n \end{array} \tag{2}$$

2. **"All or nothing" model**
Let us introduce a new variable vector $z = (z_1, \ldots, z_k)$ defined in the following manner:

$$z_i = \begin{cases} 1 & \text{if } a_i x = b_i \\ 0 & \text{if } a_i x < b_i \end{cases}$$

In other words, $z = \max\{Ax - b + e, 0\}$, where $e = (1, 1, \ldots, 1)$ and the maximization is made componentwise. Using this vector problem (1) can be written in the following manner

$$\begin{array}{rrcl} \min & cx & & \\ \text{subject to} & vz & \geq & K \\ & z & = & \max\{Ax - b + e, 0\} \\ & x & \in & \{0,1\}^n \end{array}$$

It is easy to see that this is equivalent to

$$\begin{array}{rrcll} \min & cx & & & \\ \text{subject to} & z_i & \leq & \frac{a_i}{b_i} x & i = 1, \ldots, k \\ & z & \geq & Ax - b + e & \\ & vz & \geq & K & \\ & x & \in & \{0,1\}^n & \\ & z & \in & \{0,1\}^k & \end{array} \tag{3}$$

This is a binary minimization with linear constraints. The number of the variables is $n + k$, the number of the constraints is $2k + 1$.

3. **"One is enough" model**
This model can be handled similarly to the previous one. Let now z be the following vector:

$$z_i = \begin{cases} 0 & \text{ha } a_i x = 0 \\ 1 & \text{ha } a_i x \geq 1 \end{cases}$$

namely $z = \min\{Ax, e\}$. In this case (3) can be transformed into the following problem:

$$\begin{array}{rrcll} \min & cx & & & \\ \text{subject to} & z_i & \geq & \frac{a_i}{b_i}x & i = 1, \ldots, k \\ & z & \leq & Ax & \\ & vz & \geq & K & \\ & x & \in & \{0,1\}^n & \\ & z & \in & \{0,1\}^k & \end{array} \tag{4}$$

Our second optimization problem is to find a **maximal coverage** test set supposing an upper bound for the cost (L). This optimization problem, as the previous one, can be formulated as a binary minimization problem with the functions f_i:

$$\begin{array}{rl} \max & \sum_{i=1}^{k} f_i(a_i x) \\ \text{subject to} & cx \leq L \\ & x \in \{0,1\}^n \end{array} \tag{5}$$

Without further details let us see the formulas for the three models.

1. **Linear model**

$$\begin{array}{rl} \max & \left(\sum_{i=1}^{k} \frac{a_i cov(P_i)}{b_i} \right) x \\ \text{subject to} & cx \leq L \\ & x \in \{0,1\}^n \end{array} \tag{6}$$

2. **"All or nothing" model**

$$\begin{array}{rrcll} \max & vz & & & \\ \text{subject to} & z_i & \leq & \frac{a_i}{b_i}x & i = 1, \ldots, k \\ & z & \geq & Ax - b + e & \\ & cx & \leq & L & \\ & x & \in & \{0,1\}^n & \\ & z & \in & \{0,1\}^k & \end{array} \tag{7}$$

3. **"One is enough" model**

$$\begin{array}{rrcll} \max & vz & & & \\ \text{subject to} & z_i & \geq & \frac{a_i}{b_i}x & i = 1, \ldots, k \\ & z & \leq & Ax & \\ & cx & \leq & L & \\ & x & \in & \{0,1\}^n & \\ & z & \in & \{0,1\}^k & \end{array} \tag{8}$$

4 The results of optimization

Having described the method, let us look at the experiments now. As it was mentioned in the introduction, we applied the method to the ISDN DSS1 Layer 2 protocol [2]. This TBR4 standard contains 27 test purposes (k=27) and 52 test cases (n=52) as well as the relation between the TCs and the TPs. The TCs that are necessary to check a TP are given for each purpose. Based on this standard the purpose-test incidence matrix can be easily constructed.

We fixed the coverage vector v in the value of $e = (1, \dots, 1)$ because we did not want to distinguish the TCs with respect to the coverage. We defined three different cost vectors:

- In the first, $c_1(T_j) = 1$ for $j = (1, \dots, n)$. This cost can be used if we are interested in only the number of the test cases; for example if their costs are all equal.
- The second cost vector (c_2) is based on the timers contained by the test cases. We estimated c_2 using the sum of the default times of the timers in the j^{th} test case. This time can be an upper bound for the execution time of T_j. The exact value of c_2 is as follows:

$$\begin{aligned} c_2 = \ & (6,3,33,3,3,8,4,5,3,9,6,35,3,13,2,4,34,6,6,2,34,5,3,5,2,5, \\ & 1,8,33,7,8,3,2,1,2,35,4,2,2,3,2,4,6,2,2,2,33,33,31,6,1,2) \end{aligned}$$

- The definition of the third cost vector (c_3) is based on the assumption that the main time consuming steps of testing are the preparation and, in case of fault, the search for its cause. That is why we added a constant value (100) to every $c_2(T_j)$ referring to this time, so $c_3 = c_2 + 100$.

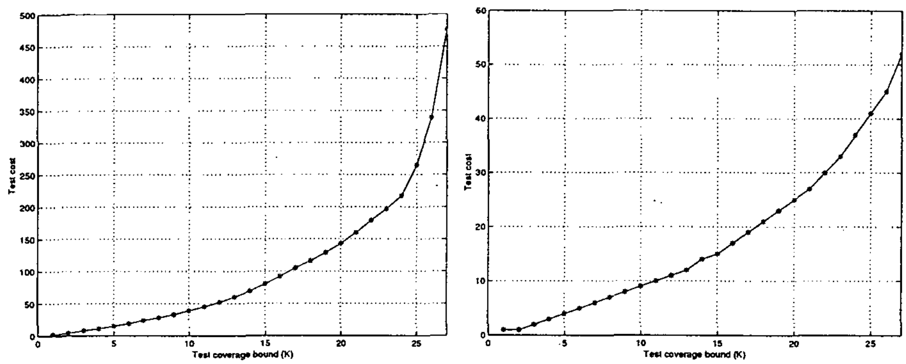
To solve the integer (binary) programming problems with the described parameters, we used the CPLEX program tool [5] .

4.1 Minimal costs

1. **Linear model**
   This model is less interesting than the others so we examined only the c_2 cost vector. The cost of the optimal test set for $K = 1, \dots, 27$ is shown in Figure 2-a.

2. **"All or nothing" model**
   We examined all the three cost vectors. The results of the optimization problem (3) for $K = 1, \dots, 27$ for the three cost functions are shown in Figure 2-b, Figure 3-a, Figure 3-b.

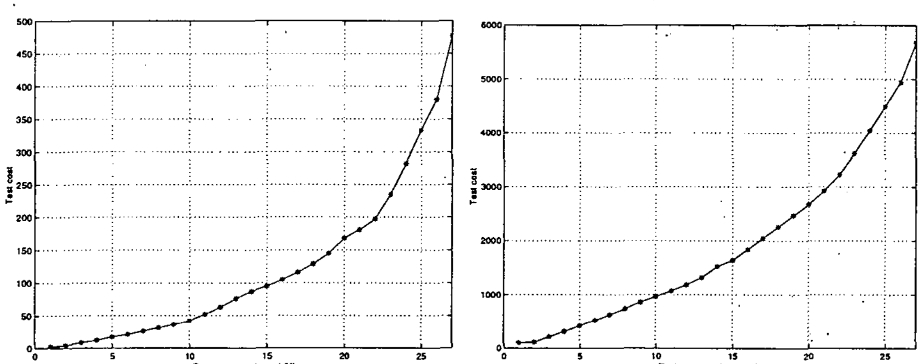
   We can see in all cases that increasing the coverage bound, the cost of the optimal test set does not increase linearly. This means that using our method we can obtain better (shorter in time) test sets to execute, than we would



(a) Linear model, c_2 cost vector

(b) "All or nothing" model, c_1 cost vector

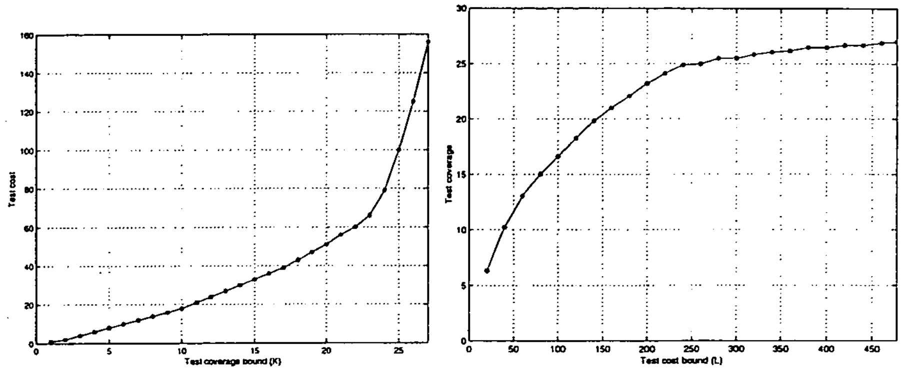
Figure 2: Minimal costs in Linear model and "All or nothing" model



(a) "All or nothing" model, c_2 cost vector

(b) "All or nothing" model, c_3 cost vector

Figure 3: Minimal costs in "All or nothing" model



(a) Minimal cost in "One is enough" model, c_2 cost vector

(b) Maximal coverage in Linear model, c_2 cost vector

Figure 4: Minimal cost in "One is enough" model and maximal coverage in Linear model

get if we chose at random. In fact, our method gives us the best possible test set within the constraints.

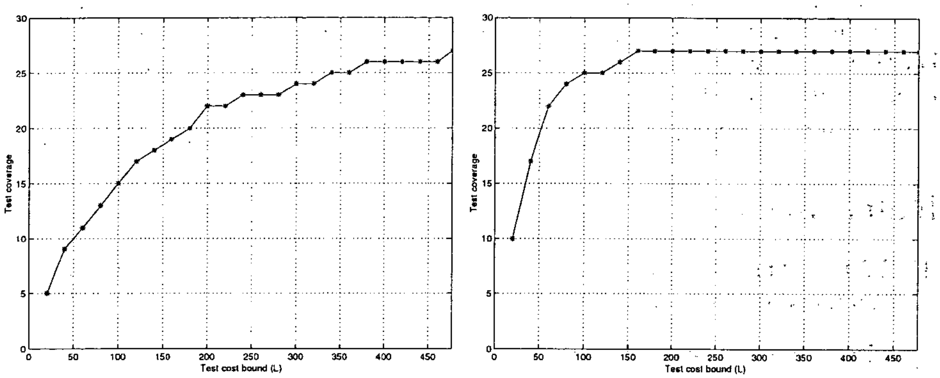
The figures also show that the biggest increase in the rate of the cost in coverage is in the case when the cost vector c_2 is used (Figure 3-a). This is because the variation of the cost values is the biggest in this case. That means we can reduce cost with only a small loss of test coverage. The cost jumps when, in order to reach the required test coverage, it is necessary to execute those test cases which have bigger cost values.

Where the variation of the cost vector is less, as in case c_3 and especially in c_1, the graph is smoother. This is quite logical as the execution of a given test case does not increase the total cost significantly regardless of which test case we select.

3. **"One is enough" model**
Figure 4-a shows the optimal cost in the "One is enough" model using cost vector c_2

4.2 Maximal coverages

1. **Linear model** When we are looking for a maximal coverage test set we have an upper bound for the cost (L). Different cost vectors have different maximal upper bounds (L_{max} =52, 477 or 5677). We present only one graph for this model, Figure 4-b, which shows the results using cost vector c_2 and



(a) "All or nothing" model, c_2 cost vector (b) "One is enough" model c_2 cost vector

Figure 5: Maximal coverage in "All or nothing" model and "One is enough" model

$L_{max} = 477$.

2. **"All or nothing" model**
   Figure 5-a shows the result of the maximal coverage problem for "All or nothing" model using the cost vector c_2.

3. **"One is enough" model**
   The results of the maximal coverage problem for the "One is enough" model using cost vector c_2 are shown in Figure 5-b. We can observe that the graph reaches the highest cost value at $L = 160$, so in this model only the third of the total cost is enough for the whole coverage.

4.3 Conclusions

The reduction of the time or effort put into conformance testing while keeping the test coverage under control is very important for those who perform conformance testing. If the time of executing conformance testing is limited, then we have to select the more efficient test cases from the whole test suite in order to make testing possible within a shorter period of time.

Our method is based on selecting test cases from the ATS of a protocol, when we can select what portion of the test coverage we are willing to devote to shorten the time of test execution. To achieve minimal testing time for a given lower bound of coverage the method determines which test cases have to be selected for execution. The method is protocol independent, so it can be used in the testing of any protocol. In the ATS of some protocols, however, the incidence matrix looks different. If there is a one-to-one relation between the test purposes and test cases, then the method cannot give us usable results.

Since the first step in the application of our method is the construction of the incidence matrix, it determines if the method is well applicable to a given protocol. For those protocols the incidence matrix is a diagonal one, we are working on other approaches to be able to give selection criteria for the test cases based on the coverage and the time constraints.

References

[1] X.290-X.296 OSI conformance testing methodology and framework for protocol recommendations for ITU-T applications

[2] ETSI Draft prTBR4 Integrated Services Digital Network (ISDN); Attachment requirements for terminal equipment to connect to an ISDN using ISDN primary rate access

[3] S. T. Vuong, Jinsong Zhu, Jadranka Alilovic-Curgus, "Sensitivity analysis of the metric based test selection", IFIP TC6 10th Int. Workshop on Testing of Communication Systems, 1997.

[4] K. Tarnay, "Protocol Specification and Testing", Akadémiai Kiadó, 1991.

[5] CPLEX, Version 3.0, 1989-1994, CPLEX Optimization, Inc.

[6] T. Csöndes, B. Kotnyek, "A Mathematical Programming Method in Test Selection", EUROMICRO'97 Short Contribution Session, 1997.

[7] T. Csöndes, S. Dibuz, B. Kotnyek, "Conformance Testing of Communication Protocols", IFIP TC6 Int. Symposium on Automation and Informatics'97, 1997.