MINIMUM-COST SOLUTIONS FOR TESTING PROTOCOLS WITH TIMERS*

M. Ümit Uyar†
Department of Electrical Engineering
City College of New York, New York, NY

Mariusz A. Fecko, Adarshpal S. Sethi
Paul D. Amer
Computer and Information Science Department
University of Delaware, Newark, DE

Abstract

A method to generate a minimum-cost test sequence for a protocol with timers is presented. The protocol timers limit the number of consecutive self-loops that can be realized in a given state. The solution presented is applicable to test sequences that use any state identification method such as UIO sequences, distinguishing sequences, and characterizing sequences. If valid and inopportune transition testing are combined, or if only valid transitions are considered, a minimum-cost solution exists. In the case of testing inopportune transitions separately, however, finding a minimum-cost solution is shown to be NP-hard.

1 Introduction

As the complexity of communication protocols increases, testing implementations for conformance to their specifications has become an integral part of the product development cycle. Without the help of formal methods in protocol testing, the interoperability of devices is questionable. Various methods for automated test generation from protocol specifications have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], based on a deterministic finite-state machine (FSM) model of the specification.

Although existing test generation methods concentrate on optimizing the test sequence length, these methods place no restrictions on the order in which the tests can be applied to an implementation under test (IUT). Possibly the most common restriction stems from an IUT's timers that, during actual testing, may limit the duration that the IUT can remain in a particular state. During testing, each state transition takes a certain time to realize. Any test sequence that traverses many consecutive self-loops easily can exceed the timer limits for the given state, and therefore not be realizable in a test laboratory (a *self-loop* represents a state transition that starts and ends at the same state).

This paper presents a solution to optimize test sequence length (and cost) under the constraint that an IUT can remain only for a limited amount of time in a given state during testing. UIO sequences [11] are used for state verification throughout the paper. However, the results presented also are applicable to test generation that uses the distinguishing or characterizing sequences [12, 13].

Two different test suites are considered. The first suite combines the testing of valid and inopportune transitions [14]. An inopportune transition occurs when an IUT receives an input not expected in its current state. In general, an inopportune transition is modeled as a self-loop, the unexpected input is simply ignored. The second suite considers testing the valid and inopportune transitions separately. We show that there exists a minimum-cost solution for the first suite and the valid transition testing part of the second suite. Optimizing the cost of a test sequence for the inopportune transition testing part of the second suite is NP-hard. A heuristic solution based on the Rural Postman Problem [15] is introduced.

This paper is organized as follows. Section 2 presents the motivation behind the optimization problem formulated in the paper. The basics of test sequence generation and the practical restrictions due to the timers are addressed in Section 3. Section 4 formulates the problem and a minimum-cost solution is presented in Section 5. Section 6 discusses issues related to optimal testing of inopportune transitions separately.

*This work supported by ARO SPP administered by Battelle (DAAL03-91-C-0034), by ARO (DAAL03-91-G-0086), and by ATIRP Consortium sponsored by the ARL under the FedLab Program (DAAL01-96-2-0002).

†Dr. Uyar, a Research Professor with CCNY, is presently Visiting Associate Professor at University of Delaware.

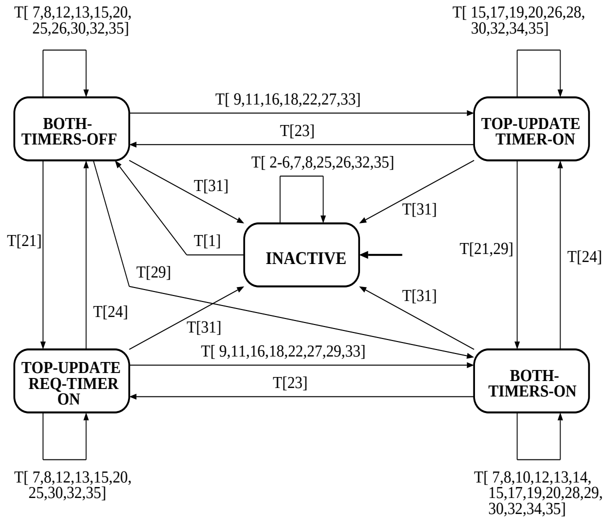

Figure 1: Extended FSM for Topology Update module of MIL-STD 188-220A.

2 Motivation

During testing, to realize a state transition takes a certain amount of time. A test sequence that traverses many consecutive self-loops in a state where a timer is running may not be realizable in a test laboratory. In this case, a timeout may disrupt the test sequence and move the IUT into a different state before all of the consecutive self-loops are exercised. If this unrealizable test sequence is not avoided during test generation, most IUTs will fail the test even when they meet the specification. Therefore, an optimization technique for generating realizable tests must consider the additional restriction that there is a limit on the number of self-loop transitions traversed consecutively.

In general, the majority of tests defined for an IUT are classified into two categories: valid and inopportune tests [5, 14]. Valid tests correspond to the "normal" or expected behavior of a protocol entity. Inopportune tests have inputs that are semantically and syntactically correct, but arrive at unexpected states (or, out of sequence). It is common practice that most inopportune messages are expected to be ignored by an IUT, which typically defines the edges representing inopportune messages as self-loops with a null or warning output.

The problem of limiting the number of consecutive self-loop traversals in a test sequence becomes an important issue for protocols that have many self-loops such as MIL-STD 188-220A [16] and LAPD [17], and for protocols with self-loop state verification sequences such as ISDN Q.931 for supplementary voice services (uses status feature which is a self-loop) [18] and LAPD (uses UIO sequences) [17]. For example, in ISDN Q.931 protocol (Basic voice services, for the user side), each state has an average of 9 inopportune transitions, which requires the traversal of 18 self-loop transitions during testing. A Q.931 implementation has several active timers that are running in certain states, e.g., timer *T304* running in state *Overlap sending*, and timer *T310* in state *Outgoing call proceeding*.

Example: Timing constraints in MIL-STD 188-220A

The UD's Protocol Engineering Lab is developing test scripts to be used by the 188-220A Conformance Tester. The test scripts specify a logical sequence of test steps that must be performed by the Conformance Tester to individually test the Data Link Layer (Types 1, 2 and 4 procedures) and Intranet Layer.

The tests are derived from an EFSM model of the protocol. An EFSM modeling the Topology Update (TU) functionality of the Intranet Layer is shown in Figure 1 [19]. There are 5 states and 86 transitions in the EFSM. The four active states are defined based on the status of two timers: *Topol-*

ogy_Update and *Topology_Update_Request*. The former is started after a topology update message is sent by a station. According to 188-220A, a station is not allowed to send another topology update message until the timer expires. The latter timer performs the same role for topology update request messages.

Each timer may or may not be running in a given point in time - this gives four possible configurations. The timers' states determine the I/O behavior of the TU module, because a running timer prevents the station from sending certain messages. For example, when the topology information changes, the station is allowed to send a topology update message only if the *Topology_Update Timer* is not running.

As can be seen in Figure 1, there are 10 self-loop transitions defined for states *TOP-UPDATE-REQ-TIMER-ON* and *TOP-UPDATE-TIMER-ON*, and 16 self-loop transitions in state *BOTH-TIMERS-ON*. Depending on the timer expiration values for both timers, it may or may not be possible to test all self-loop transitions during one visit to a state. This constraint must be taken into account while generating realizable test sequences for the Intranet Layer of 188-220A. If the constraint is not considered, all valid implementations will fail the test sequence, which is not what the tester desires.

3 Preliminaries and practical restrictions on the test sequences

A *protocol* can be specified as a deterministic FSM [13, 20], which can be represented by a directed graph $G = (V, E)$. The set $V = \{v_1, \ldots, v_n\}$ of vertices correspond to the set of states S of the FSM. A directed edge from v_i to v_j with label $L_k = a_l/o_m$, and the *cost* to realize the edge during testing, corresponds to a state transition in the FSM from s_i to s_j by applying input a_l and observing output o_m. If the start and the end vertices of an edge are the same (i.e., $v_i = v_j$), the edge is called a *self-loop*. The *indegree* and *outdegree* of a vertex are the number of edges coming toward and directed away from it, respectively. If the indegree and outdegree of each vertex are equal, the graph is said to be *symmetric*.

A *tour* is a sequence of consecutive edges that starts and ends at the same vertex. An *Euler tour* is a tour that contains every edge of G exactly once. The so-called *Chinese Postman Problem* is defined as finding a minimum-cost tour of G that traverses every edge at least once [21]. The *Rural (Chinese) Postman Problem* is finding a (minimum-cost) tour for a subset of edges in G [15].

During conformance testing of a protocol implementation, the IUT is viewed as a *black box*, where only the inputs applied to the IUT and the outputs generated by the IUT can be controlled and observed, respectively. An IUT *conforms* to its specification if all state transitions defined in the specification are tested successfully. To test a single transition defined from state v_i to v_j, the following steps are needed: **1)** bring the IUT into state v_i; **2)** apply the required input and compare the output(s) generated with those defined by the specification; **3)** verify that the new state of the IUT is v_j by applying a state verification sequence.

Aho et al. [1] introduced an optimization for the test sequence length (and cost) using UIO sequences to perform the last step of the above single transition test. A UIO sequence of a state s_i is a sequence of edges starting at v_i such that the output sequence generated by these edges is unique for v_i.

The existing methods for conformance test generation [1, 6, 7, 11, 12, 13, 20, 22, 23, 24, 25] emphasize optimizing the test sequence length and its cost, without considering any restrictions on the order in which the tests can be applied to an IUT. However, an optimization technique for generating realizable tests must consider the additional restriction that there is a limit on the number of self-loop transitions traversed consecutively.

Two test suites are considered in this paper. In the first test suite, valid and inopportune tests are handled together. In the second test suite, the valid and inopportune tests are performed separately. Two test sequences are generated in this case - the first one tests valid transitions, whereas the second one only inopportune transitions. A minimum-cost test sequence generation method for the first test suite is presented in Section 5. In most cases, the generated test sequence will be longer than one without the constraint since limiting the number of self-loop traversals may require additional visits to a state which otherwise would have been unnecessary.

The solution given in Section 5 also is applicable to the valid edge tests in the second test suite. However, to generate a minimum-cost test sequence for *inopportune edges only* is shown to be NP-hard in Section 6. Heuristics based on the Rural Chinese Postman Problem is introduced for this case.

4 Problem formulation

Let the directed graph representing the FSM for a given protocol be $G(V,E)$. Let $\beta(v_i,v_j)$ and $\psi(v_i,v_j)$ be the capacity and cost of the edge $(v_i,v_j) \in E$, respectively, where $v_i, v_j \in V$. Each edge $e \in E$ is labeled as either valid (e_{valid}) or inopportune ($e_{inopportune}$). Let us divide edges in E into three disjoint sets:

- *valid self-loop transitions:* $E_{vsl} = \{(v_i,v_j) : v_i, v_j \in V \ \wedge \ (v_i,v_j)_{valid} \in E \ \wedge \ v_i = v_j\}$
- *remaining valid transitions:* $E_{vnsl} = \{(v_i,v_j) : v_i, v_j \in V \ \wedge \ (v_i,v_j)_{valid} \in E \ \wedge \ v_i \neq v_j\}$
- *inopportune transitions:* $E_{inop} = \{(v_i,v_j) : v_i, v_j \in V \ \wedge \ (v_i,v_j)_{inopportune} \in E \wedge \ v_i = v_j\}$

In general, besides the three sets being mutually disjoint, together they account for all transitions:

$$E = E_{vsl} \cup E_{vnsl} \cup E_{inop} \tag{1}$$

During testing, after traversing a given transition for the first time, state verification is performed. We first consider the special case where all UIO sequences are self-loops. The general case where UIO sequences may contain non-self-loops is presented in Section 5.1.

4.1 Constraints on the number of visits to a state

Let us consider a test suite where the valid and inopportune transition tests are combined and formally define the problem constraints. The case where the inopportune cases are tested separately is studied in Section 6.

Let $E_{self} \equiv E_{vsl} \cup E_{inop}$ be the set of self-loop edges to be tested. Let $d_{self}(v_i)$, the number of self-loops of vertex v_i, be defined as the number of edges in E_{self} incident on v_i. Let $d_{min\_self}(v_i)$ be the minimum number of times a tour covering all edges of $E_{vnsl} \cup E_{self}$ must include vertex $v_i \in V$.

Let $d_{state\_ver}(v_i)$ be the number of self-loop transitions used to verify whether an IUT is in state v_i. Suppose that during testing, a given vertex $v_i \in V$ can tolerate at most $max\_self(v_i)$ self-loops executed at one visit to vertex v_i. As indicated in Section 3, attempting to remain in state v_i to execute $1 + max\_self(v_i)$ self-loops would result in disruption of a test sequence. Testing a self-loop transition involves traversing the self-loop transition followed by applying the state verification self-loop sequence, which contains $d_{state\_ver}(v_i)$ transitions.

Due to space limitations, we are unable to include the detailed derivation of $d_{min\_self}(v_i)$. In [26], we prove that the minimum number of times vertex v_i must be visited in a test sequence is as follows:

$$d_{min\_self}(v_i) \stackrel{def}{=} \begin{cases} d_{in}(v_i) & \text{if } d_{self}(v_i) \leq (d_{in}(v_i) * \Delta_1(v_i)) \\ \Gamma(v_i) & \text{if } d_{self}(v_i) > (d_{in}(v_i) * \Delta_1(v_i)) \end{cases} \tag{2}$$

where $d_{out}(v_i)$ and $d_{in}(v_i)$ are respectively the out-degree and the in-degree of vertex v_i in E_{vnsl}, and where

$$\begin{aligned} \Gamma(v_i) &= d_{in}(v_i) + \left\lceil \frac{d_{self}(v_i) - (d_{in}(v_i) * \Delta_1(v_i))}{\Delta_2(v_i)} \right\rceil \\ \Delta_1(v_i) &= \left\lfloor \frac{max\_self(v_i) - d_{state\_ver}(v_i)}{1 + d_{state\_ver}(v_i)} \right\rfloor \\ \Delta_2(v_i) &= \left\lfloor \frac{max\_self(v_i)}{1 + d_{state\_ver}(v_i)} \right\rfloor \end{aligned}$$

Let g be a function of two arguments: an edge $e \in E_{vnsl}$ and an integer k. The value of g is a set of $k \geq 0$ copies of its first argument $e \in E$. The function g represents the duplications of an edge $e \in E$ in G.

Let $G^{'}(V^{'} \equiv V, E^{'} \equiv E_{vnsl})$ and its symmetric augmentation $G^{''}(V^{''} \equiv V, E^{''})$ be the graphs satisfying the following conditions:

$$E_g \stackrel{def}{=} \bigcup_{e_{dup} \in E^{'}} g(e_{dup}, f(e_{dup})) \tag{3}$$

$$E^{''} = E^{'} \cup E_g \tag{4}$$

$$(\forall v_i^{''} \in V^{''}) \quad d_{in}(v_i^{''}) = d_{out}(v_i^{''}) \tag{5}$$

$$(\forall v_i^{''} \in V^{''}) \quad d_{in}(v_i^{''}) \geq d_{min\_self}(v_i) \tag{6}$$

The function f is the maximum-flow minimum-cost function defining G'' as the *symmetric augmentation* of G', which will be discussed in Section 5. By definition, in G'', the in-degree of any vertex $v_i'' \in V''$ is equal to its out-degree. Also, equations (5) and (6) suggest that the in-degree of any vertex $v_i'' \in V''$ be greater or equal to the value defined by $d_{min\_self}(v_i)$, where v_i is the corresponding vertex in V. †

Our goal is to build a Chinese Postman tour in which the constraint set by (6) is satisfied for each vertex $v_i' \in V'$. A Chinese Postman tour is a minimum-cost tour covering each transition $e \in E_{vnsl} \cup E_{self}$ at least once, and is equivalent to an Euler tour in a minimum cost symmetric graph G''. In other words, our goal is to obtain the minimum-cost symmetric augmentation of the graph G' as the graph G''. Therefore, this goal is now reduced to finding the value of the function f in equation (3) above for all edges in E'.

5 Minimum-cost solutions for constrained self-loop testing

We present the following solution to the problem of finding a symmetric G'' while satisfying the constraint set in (6). First, $G'(V', E')$ is converted to $G^*(V^*, E^*)$ by splitting each vertex $v_i' \in V'$ satisfying

$$d_{min\_self}(v_i) \quad > \quad max(d_{in}(v_i), d_{out}(v_i)) \qquad (7)$$

into the two vertices $v_i^{*(1)}$, $v_i^{*(2)} \in V^*$ (Figure 2).

Then, $v_i^{*(1)}$ is connected to $v_i^{*(2)}$ with a set of edges with cardinality of $d_{min\_self}(v_i)$: $E_1^* \stackrel{def}{=} \bigcup_{v_i' \in V'} g((v_i^{*(1)}, v_i^{*(2)}), d_{min\_self}(v_i))$. Each edge in E_1^* is assigned infinite capacity β and a zero cost ψ. Finally, the original edges in E' are added to E^* as follows: $E_2^* \stackrel{def}{=} \{(v_i^{*(2)}, v_j^{*(1)}) : (v_i', v_j') \in E'\}$. The last step of the conversion is the addition of the source and sink vertices (s and t, respectively): $E^* \stackrel{def}{=} E_1^* \cup E_2^* \cup \bigcup_{v_i' \in V'} \{(s, v_i^{*(1)}), (s, v_i^{*(2)}), (v_i^{*(1)}, t), (v_i^{*(2)}, t)\}$.

†Note that, unless stated otherwise, $v_i', v_i'', v_i^*, v_i^\delta$ and v_i^σ are used in this paper to denote the copies of a corresponding vertex $v_i \in V$ in graphs G', G'', G^*, G^δ and G^σ, respectively.

The problem of finding the minimum-cost augmentation of G' as G'' then can be reduced to finding the integer function $f : E \to N$ whose value $f(v_i^{*(2)}, v_j^{*(1)})$ determines the number of times an edge $(v_i', v_j') \in E'$ needs to be duplicated to make the graph G' symmetric (Figure 2).

Aho et al. [1] presented an efficient solution to this problem for FSMs with either a self-loop property or a reset capability. We now apply a similar approach to the problem of minimizing the test sequence with the above self-loop repetition constraints. We use network flow techniques to maximize the flow on graph G^* with minimum cost. Edges incident to the source and sink in G^* are assigned capacity β as follows:

$$\beta(s, v_i^{*(1)}) = \max(0, d_{in}(v_i') - d_{min\_self}(v_i)) \qquad (8)$$

$$\beta(s, v_i^{*(2)}) = \max(0, d_{min\_self}(v_i) - d_{out}(v_i')) \qquad (9)$$

$$\beta(v_i^{*(1)}, t) = \max(0, d_{min\_self}(v_i) - d_{in}(v_i')) \qquad (10)$$

$$\beta(v_i^{*(2)}, t) = \max(0, d_{out}(v_i') - d_{min\_self}(v_i)) \qquad (11)$$

Then, the edges of $(s, v_i^{*(1)})$, $(s, v_i^{*(2)})$, $(v_i^{*(1)}, t)$, and $(v_i^{*(2)}, t)$, are assigned a zero cost ψ. Each of the remaining edges in E^* (i.e., the edges corresponding to original edges in E') has infinite capacity with the cost of the original edge in E'.

f is the maximum-flow minimum-cost function defined on the graph $G^*(V^*, E^*)$ that saturates all edges incident to s and t. Formally, such an f needs to satisfy the following conditions $\forall v_i^* \in V^* - \{s, t\}$):

$$\begin{aligned} \sum_{v_j^* \in V^*} f(v_j^*, v_i^*) &= \sum_{v_j^* \in V^*} f(v_i^*, v_j^*) \\ \beta(s, v_i^*) &= f(s, v_i^*) \\ \beta(v_i^*, t) &= f(v_i^*, t) \end{aligned}$$

The function f satisfying the above conditions exists iff

$$\sum_{v_i^* \in V^* - \{s,t\}} \beta(s, v_i^*) = \sum_{v_i^* \in V^* - \{s,t\}} \beta(v_i^*, t)$$

which holds true for capacity assignments defined by (8) through (11) [27].

Aho et al. show that the maximum-flow minimum-cost function on a strongly-connected graph define its optimal symmetric augmentation. Let us first prove that the symmetric augmentation of G' as

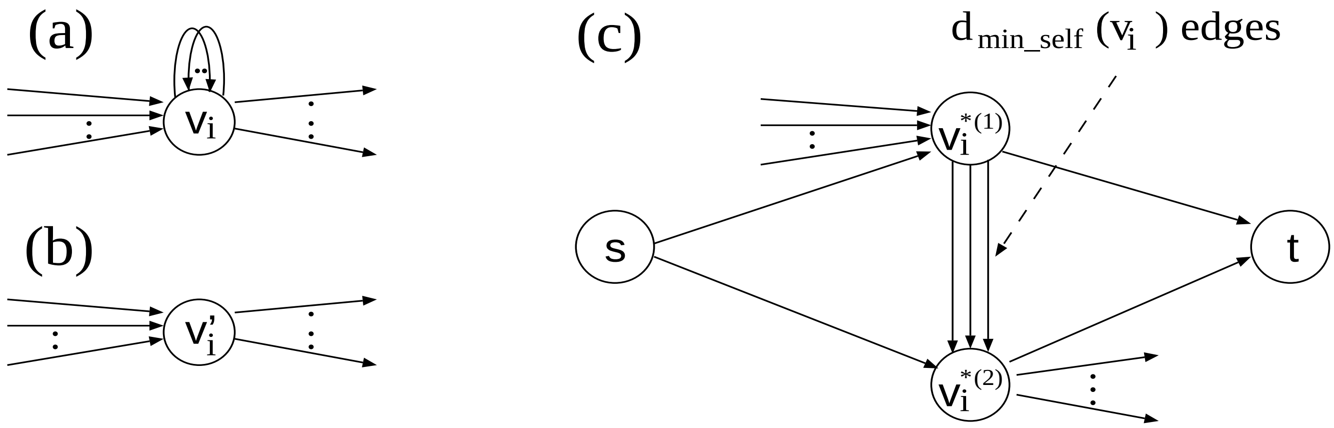

Figure 2: Conversion of v_i in G (part (a)), to $v_i^{'}$ in $G^{'}$ (part (b)) and to $v_i^{*(1)}, v_i^{*(2)}$ in G^* (part (c)).

$G^{''}$, defined by the maximum-flow minimum-cost f on G^*, satisfies constraint (6).

Proof: If a vertex $v_i^{'}$ is **not** split by the algorithm (i.e., each $v_i^{'}$ for which the condition (7) does not hold), then either $d_{in}(v_i) \geq d_{min\_self}(v_i)$ or $d_{out}(v_i) \geq d_{min\_self}(v_i)$. Then the flow defined by f satisfies the following condition:

$$d_{min\_self}(v_i) \leq \sum_{v_j^* \in V^* - \{s,t\}} f(v_j^*, v_i^*) + d_{in}(v_i) = \sum_{v_j^* \in V^* - \{s,t\}} f(v_i^*, v_j^*) + d_{out}(v_i) \quad (12)$$

If a vertex $v_i^{'}$ is split by the algorithm (i.e., each $v_i^{'}$ for which the condition (7) does not hold), then the following holds:

$$d_{min\_self}(v_i) \leq \sum_{v_j^* \in V^* - \{s,t\}} f(v_j^*, v_i^{*(1)}) + d_{in}(v_i) = \sum_{v_j^* \in V^* - \{s,t\}} f(v_i^{*(1)}, v_i^{*(2)}) + d_{min\_self}(v_i) \quad (13)$$

$$d_{min\_self}(v_i) \leq \sum_{v_j^* \in V^* - \{s,t\}} f(v_j^{*(2)}, v_i^*) + d_{out}(v_i) = \sum_{v_j^* \in V^* - \{s,t\}} f(v_i^{*(1)}, v_i^{*(2)}) + d_{min\_self}(v_i) \quad (14)$$

Equations (12), (13) and (14) imply that in both cases $v_i^{''}$ will have at least $d_{min\_self}(v_i)$ incident edges after duplication. □

Next, let us prove that T, an Euler tour of $G^{''}$ defined by f, is a minimum-cost tour of $G^{'}$ satisfying (6).

Proof: BWOC, suppose that $\check{T}$ is a minimum-cost tour of $G^{'}$ satisfying (6) such that

$$\psi(\check{T}) \quad < \quad \psi(T) \qquad (15)$$

Since $\check{T}$ satisfies (6), it visits each vertex $v_i \in V$ at least $d_{min\_self}(v_i)$ times. Then $\check{T}$ induces a flow $\check{f}$ on G^* defining a symmetric augmentation of $G^{'}$ as $\check{G}^{''}$. (15) implies that the value of $\check{f}$ is less than the value of f. This contradicts the fact that f is a minimum-cost flow on G^*. □

Example: Consider the FSM with self-loop transitions shown in Figure 3. Suppose that vertices v_0, v_2, and v_3 of the FSM can tolerate at most three, and v_1 at most two self-loop transitions during each visit. Let transitions $e10$ and $e11$ correspond to timeouts. After either $e10$ or $e11$ is triggered, the FSM is brought into state v_3. UIO sequences and the values of max_self, d_{state_ver} and d_{min_self} for vertices v_0, v_1, v_2, and v_3 are as follows:

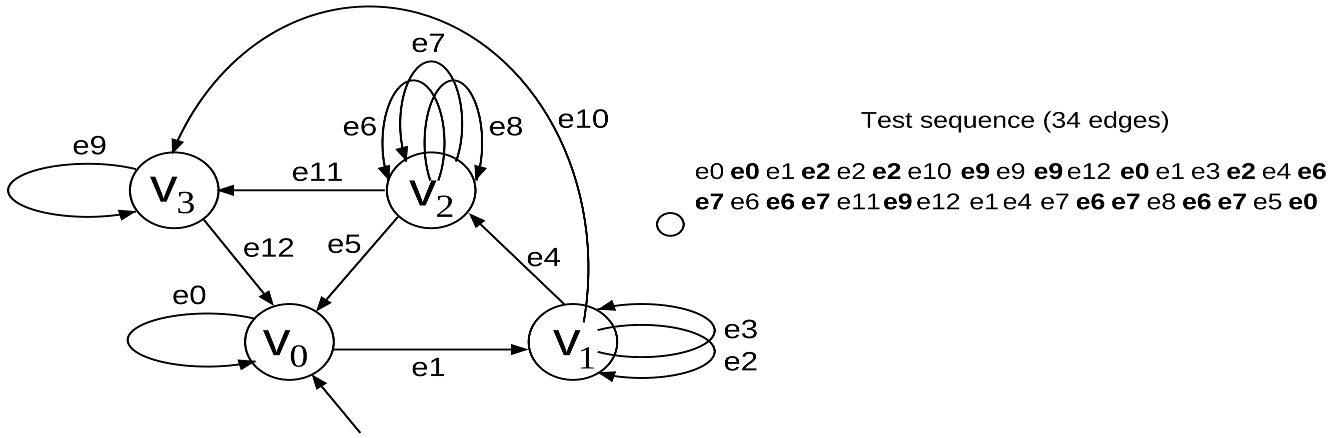

Figure 3: Minimum-cost test sequence without self-loop repetition constraint.

Vertex	*UIO*	max_self	d_{state_ver}	d_{min_self}
v_0	e0	3	1	2
v_1	e2	2	1	3
v_2	e6,e7	3	2	4
v_3	e9	3	1	2

The Chinese postman method [21] when applied to the graph without self-loop repetition constraint results in the test sequence

$$\begin{gathered} e0, \mathbf{e0}, e1, \mathbf{e2}, e2, \mathbf{e2}, e10, \mathbf{e9}, e9, \mathbf{e9}, e12, \mathbf{e0}, \\ e1, e3, \mathbf{e2}, e4, \mathbf{e6}, \mathbf{e7}, e6, \mathbf{e6}, \mathbf{e7}, e11, \mathbf{e9}, \\ e12, e1, e4, e7, \mathbf{e6}, \mathbf{e7}, e8, \mathbf{e6}, \mathbf{e7}, e5, \mathbf{e0} \end{gathered} \tag{16}$$

containing 34 edges (the edges used for the purpose of state verification appear in bold).

As can be seen from the beginning part of the above test sequence

$$e0, \mathbf{e0}, e1, \mathbf{e2}, e2, \mathbf{e2}, e10, ...$$

it is required that, after $e1$ is traversed, the IUT should stay in state v_1 for a time that allows at least three self-loop traversals. However, this part of the test sequence is not realizable in a test laboratory because the timeout edge $e10$ will be triggered after the second consecutive self-loop traversal (i.e., $max\_self(v_1) = 2$). The IUT will prematurely move into v_3 and the test sequence will be disrupted.

To address the problem of test sequence disruption due to timeouts, the graph of Figure 3 is converted by the method described in Section 5 to the graph shown in Figure 4. The vertices for which condition (7) holds, which are v_1 and v_2, are split and then connected by $d_{min\_self}(v_1) = 3$ and $d_{min\_self}(v_2) = 4$ edges, respectively. Considering the constrained self-loop problem, the test sequence for the graph of Figure 4 is obtained as

$$\begin{gathered} e0, \mathbf{e0}, e1, \mathbf{e2}, e10, \mathbf{e9}, e9, \mathbf{e9}, e12, \mathbf{e0}, \\ e1, e2, \mathbf{e2}, e4, \mathbf{e6}, \mathbf{e7}, e11, \mathbf{e9}, e12, e1, \\ e3, \mathbf{e2}, e4, e6, \mathbf{e6}, \mathbf{e7}, e5, \mathbf{e0}, e1, e4, \\ e7, \mathbf{e6}, \mathbf{e7}, e5, e1, e4, e8, \mathbf{e6}, \mathbf{e7}, e5 \end{gathered} \tag{17}$$

containing 40 edges.

Although the test sequence in Figure 4 is longer than that of Figure 3, it is minimum-length with the introduced self-loop constraint. During each visit to vertices v_0, v_1, v_2 and v_3, the number of consecutive self-loop edges traversed is less than or equal to the maximum allowed number of self-loop traversals. Therefore, this test sequence is realizable in the test laboratory.

5.1 Results for non-self-loop UIO sequences

In the case of self-loop UIO sequences, $d_{min\_self}(v_i)$, the minimum number of times vertex v_i must be included in a tour covering all edges of $E_{vnsl} \cup E_{self}$, is defined by equation (2). In the more general case, some UIO sequences may consist of both (or either) self-loop and non-self-loop edges.

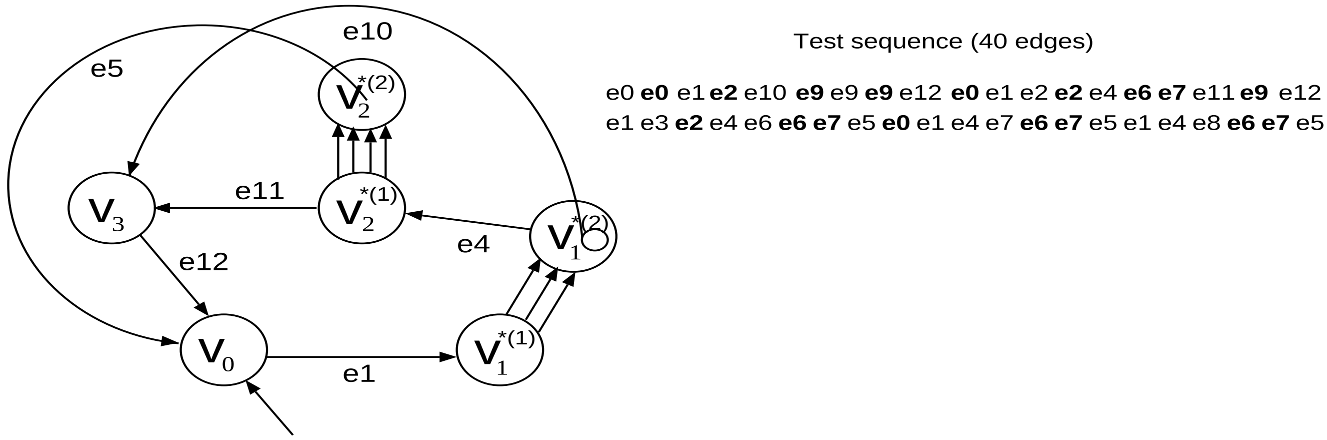

Figure 4: Minimum-cost test sequence with self-loop repetition constraint.

A UIO sequence for vertex v_i (which we denote as $UIO(v_i)$) can be viewed as a concatenation of a number of (some possibly empty) subsequences:

$$\begin{aligned} UIO(v_i) &= uio(v_i, v_i) \cdot uio(v_i, v_{j_0}) \cdot uio(v_{j_0}, v_{j_0}) \cdot \\ &\ldots \cdot uio(v_{j_{m-1}}, v_{j_m}) \cdot uio(v_{j_m}, v_{j_m}) \end{aligned} \quad (18)$$

where $\cdot$ is a concatenation operator. In (18), $uio(v_{j_k}, v_{j_k})$ denotes the subsequence that contains only self-loop edges of v_{j_k}. A subsequence of $uio(v_{j_{k-1}}, v_{j_k})$ is a path of non-self-loop edges starting at $v_{j_{k-1}}$ and ending at v_{j_k}.

Based on this definition, there are three possible forms that a UIO sequence can have [28]. The optimization method presented in this paper was generalized in our further work to account for different forms of UIO sequences. The results are available in [28].

6 Testing of inopportune transitions separately

We now consider the case where the valid and inopportune transitions are to be tested separately. The solution presented in the previous section is directly applicable to the valid transitions of this test suite. However, minimum-cost test generation for only inopportune transitions needs revisiting. Since, separated from the valid transitions, inopportune transitions alone cover only a subset of the edges, finding the minimum-cost solution becomes more difficult.

Let $V_{self} \subseteq V$ be the set of vertices that have at least one inopportune self-loop transition. During testing, we build a tour that contains all inopportune self-loop transitions, which are distributed over the vertices in V_{self}. To be able to bring an IUT into a state with self-loop transitions, we must include a subset of valid non-self-loop transitions in a transition tour. Let $d_{min\_inop}(v_i) > 0$ be the minimum number of times that the test sequence must visit $v_i \in V_{self}$ (see [26] for derivation) . The problem of testing inopportune transitions in a cost-effective way is now reduced to finding a minimum cost transition tour in $G^{'}(V^{'}, E^{'})$ that consists of a subset of E_{vnsl} and visits each vertex in $v_i \in V_{self}$ at least $d_{min\_inop}(v_i) > 0$ times.

We show [26] by reduction from the Rural Chinese Postman Problem (RCPP), which is known to be NP-complete for the general case [15], that the above defined problem is NP-hard. Heuristics based on RCPP is introduced to find an approximate solution to testing inopportune transitions in an optimal way.

7 Conclusion

This paper describes a method to generate a minimum-cost test sequence with the constraint that only a limited number of consecutive self-loops can be realized in a given state. This constraint for example may be active timers that are running in certain states. The final test sequence is longer than the absolute minimum length due to the additional constraints imposed on the optimization problem by the timers. The test length can be further shortened by the segment overlapping techniques of Chen et al. [23, 24], Yang and Ural [25], and Miller and Paul [6, 7].

The solution presented in this paper is applicable to test sequences that use various state identification methods such as UIO sequences, distinguishing sequences, and characterizing sequences.

If the valid and inopportune transition testing are combined, or only the valid transitions are considered, a minimum-cost solution exists. The minimum-cost test sequence generation for testing the inopportune transitions separately is shown to be NP-hard.

As future work, this method will be implemented as a software tool and be applied to MIL-STD 188-220A [16].

References

[1] A. V. Aho, A. T. Dahbura, D. Lee, and M. U. Uyar, "An optimization technique for protocol conformance test generation based on UIO sequences and rural Chinese postman tours," *IEEE Trans. on Communications*, vol. 39, pp. 1604–1615, Nov 1991.

[2] B. S. Bosik and M. U. Uyar, "FSM-based formal methods in protocol conformance testing: from theory to implementation," *Computer Networks and ISDN Systems*, vol. 22, Sep 1991.

[3] W. Y. Chan and S. T. Vuong, "An improved protocol test generation procedure based on UIOs," in *Proc. ACM SIGCOMM*, Sep 1989.

[4] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines - a survey," *Proc. of the IEEE*, vol. 84, pp. 1090–1123, Aug 1996.

[5] R. J. Linn, "Conformance testing for OSI protocols," *Computer Networks and ISDN Systems*, vol. 18, pp. 203–219, 1990.

[6] R. E. Miller and S. Paul, "Generating maximal fault coverage conformance test sequences of reduced length for communication protocols," in *Proc. Intn'l Conference on Network Protocols*, (San Francisco, CA), Oct 1993.

[7] R. E. Miller and S. Paul, "On the generation of minimal-length conformance tests for communication protocols," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 116–129, Feb 1993.

[8] R. E. Miller and S. Paul, "Structural analysis of protocol specifications and generation of maximal fault coverage conformance test sequences," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 457–470, Oct 1994.

[9] B. Sarikaya, G. Bochmann, and E. Cerny, "A test design methodology for protocol testing," *IEEE Trans. Software Engineering*, vol. 13, pp. 518–531, May 1987.

[10] H. Ural, "Formal methods for test sequence generation," *Computer Communications*, vol. 15, pp. 311–325, Jun 1992.

[11] K. K. Sabnani and A. T. Dahbura, "A protocol test generation procedure," *Computer Networks and ISDN Systems*, vol. 15, pp. 285–297, 1988.

[12] A. Bhattacharyya, *Checking Experiments in Sequential Machines.* New York, N.Y.: John Wiley & Sons, 1989.

[13] Z. Kohavi, *Switching and Finite Automata Theory.* New York, N.Y.: McGraw Hill, 1978.

[14] *Information Processing Systems - OSI Conformance Test Methodology and Framework*, 1991.

[15] J. K. Lenstra and A. H. G. R. Kan, "On general routing problems," *Networks*, vol. 6, pp. 273–280, 1976.

[16] *Military Standard - Interoperability Standard for Digital Message Device Subsystems (MIL-STD 188-220A), 27Jul95*, 1995.

[17] M. U. Uyar and M. H. Sherif, "Protocol modeling for conformance testing: Case study for the ISDN LAPD protocol," *AT&T Technical Journal*, vol. 69, Jan 1990.

[18] AT&T 5E4 Generic Program, *AT&T 5ESS^{TM} Switch - ISDN Basic Rate Interface Specification*, Sep 1985.

[19] M. A. Fecko, P. D. Amer, A. S. Sethi, M. U. Uyar, T. Dzik, R. Menell, and M. McMahon, "Formal design and testing of MIL-STD 188-220A based on Estelle," in *Proc. MILCOM'97*, (Monterey, California), Nov 1997.

[20] Y. N. Shen, F. Lombardi, and A. T. Dahbura, "Protocol conformance testing using multiple UIO sequences," *IEEE Trans. on Communications*, vol. 40, pp. 1282–1287, Aug 1992.

[21] M. U. Uyar and A. T. Dahbura, "Optimal test sequence generation for protocols: the Chinese postman algorithm applied to Q.931," in *Proc. IEEE GLOBECOM*, pp. 68–72, Dec 1986.

[22] H. Ural and Y. Lu, "An improved method for test sequence generation," Tech. Rep. TR-90-12, Dept. of CSI, University of Ottawa, Mar 1990.

[23] M. S. Chen, Y. Choi, and A. Kershenbaum, "Minimal length test sequences for protocol conformance," in *Proc. First Network Management and Control Workshop*, (New York, NY), 1989.

[24] M. S. Chen, Y. Choi, and A. Kershenbaum, "Approaches utilizing segment overlap to minimize test sequences," in *Proc. PSTV X*, pp. 85–98, 1990.

[25] B. Yang and H. Ural, "Protocol conformance test generation using multiple UIO sequences with overlapping," in *Proc. ACM SIGCOMM'90*, pp. 118–125, 1990.

[26] M. U. Uyar, M. A. Fecko, A. S. Sethi, and P. D. Amer, "Minimum-cost solutions for testing protocols with timers," Tech. Rep. TR-97-17, CIS Dept., University of Delaware, Newark, DE, 1997.

[27] L. R. Ford and D. R. Fulkerson, *Flows in Networks.* Princeton University Press, 1962.

[28] M. U. Uyar, M. A. Fecko, A. S. Sethi, and P. D. Amer, "Test generation for protocols with timing constraints," Tech. Rep. TR-98-07, CIS Dept., University of Delaware, Newark, DE, 1997. (Submitted for publication).