Chapter 1

Combinatorial Optimization Problems in Multi-Function System Testing: Outline

Mark Sh. Levin, Mark Last
Ben-Gurion University, Israel
{marksh,mlast}@bgumail.bgu.ac.il

A 5-layer hierarchy of system characteristics is proposed: (1) input-output, (2) input-output relationships, (3) system functions, (4) system function clusters; (5) digraph of system function clusters. The following problems are examined for layers 3, 4, and 5: synthesis of a test case for a system function cluster; scheduling of test cases; design of test case sequence for a chain of system function clusters, designing a collection of test case sequences as chain covering of function cluster digraph, structural fusion of ordinal unit test results.

1.1 Introduction

This article addresses a framework for a black-box testing process under multi-function approach that is based on combinatorial problems ([13], [14], [15]). Our current research corresponds to basic contemporary tendencies in system studies and systems engineering: ([2], [10], [12], [17], [18], etc.): *1. FROM* the designer's viewpoint *TO* the user's viewpoint; *2. FROM* serial processes *TO* concurrent processes; *3. FROM* analysis of an existing / designed system *TO* analysis / forecasting of system development / evolution processes; and *4. FROM* traditional system design methodology *TO* modular approaches in the design of products, systems, and multi-product systems (platforms).

Contemporary steps in system testing are the following ([1], [4], [5], [8], [19], etc.): *1. FROM* input-output analysis *TO* system functions investigation; *2. FROM* serial execution *TO* concurrent scheme including extraction of system properties (e.g., data on system structure, input-output relationships) and usage of the properties for advanced system testing; *3. FROM* simple input vectors *TO* the following: (i) combinations of inputs (pairs, triples, etc.), (ii) sequences of inputs, and (iii) analysis of differences between inputs; and *4. FROM* a formal input-output analysis / testing *TO* context-based analysis / testing processes.

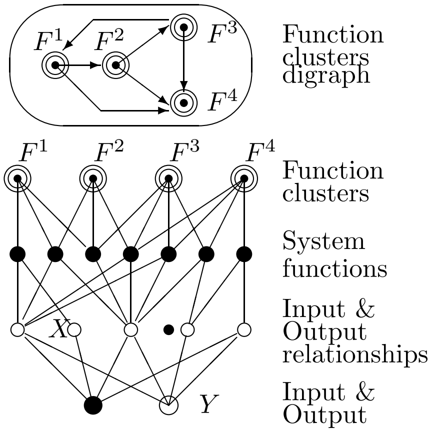

Fig. 1. System hierarchy

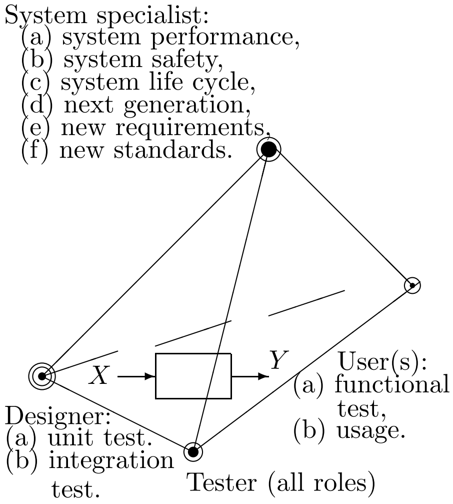

Fig. 2. Roles and responsibility

Here the extended system hierarchy is used (Fig. 1). Fig. 2 depicts main roles and tasks in the system testing framework. The following test units are examined: *1.* Individual test cases; *2.* Sequences of individual test cases (multi-stage test cases): $z =< t_{s[1]}, t_{s[2]}, ..., t_{s[k]} >$; *3.* Collections of test case sequences $Z = \{z_i | i = 1, ..., L\}$. In addition, quality estimates of system units on ordinal scale are used for the analysis of possible structural fusion of "quasi-good" unit test results from the multi-functional viewpoint. Mainly, the problems are based on well-known and new data mining techniques, combinatorial optimization models and algorithms ([7], [11], [12], etc.]. Fig. 3 depicts a test case space and corresponding relations / operations.

Functional system modeling is an important contemporary approach ([1], [3], [8], etc.). Let us consider a set of system functions $F = \{f_1, ..., f_l, ..., f_L\}$. System function f_l corresponds to a set of requirements [5]. System function $f_l \in F$ is based on two sets: inputs $X(f_l)$ and outputs $Y(f_l)$. Main properties of system functions are:

1. Structural parameters (e.g., corresponding system outputs and values), weights of "functional" importance, and significance of possible fault(s).

2. Relations between (among) system functions: (a) by inputs, (b) by outputs, and (c) by inputs & outputs. The binary relations are the following: structural relations: *independence*, *dependency* (of *intersection*) (e.g., common outputs, common inputs), *inclusion* (i.e., function f_p is a subfunction of function f_q); relations by usage: *concurrency* (i.e., joint usage, etc.), *precedence*.

Figs. 4 and 5 illustrate system function clusters, the most important test cases for some function clusters (Fig. 4), a test case sequence for a chain of function clusters, and a collection of test case sequences that covers a digraph of function clusters.

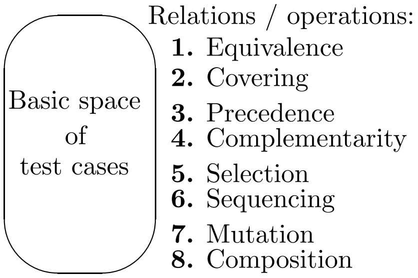

Fig. 3. Test cases space

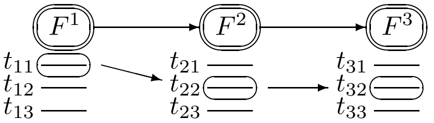

Fig. 4. Chain of function clusters

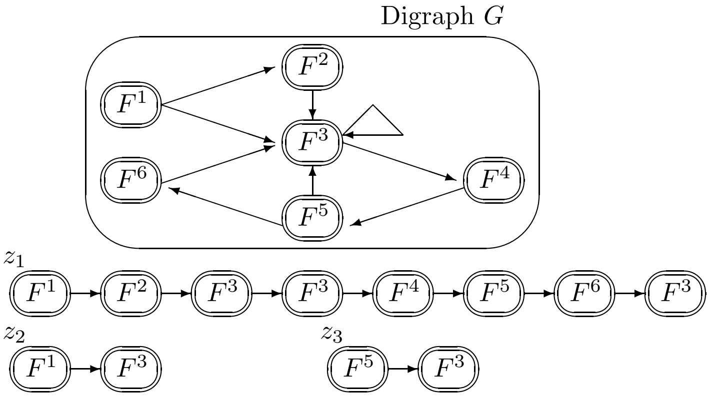

Fig. 5. Chain covering of function cluster digraph

1.2 Problems

1.2.1 Selection and Synthesis

Traditional approaches to testing processes are based on reducing an initial set of test cases. The selection process may be based on prioritizing the test cases and packing a knapsack of the most important test cases while taking into account resource constraints ([6], etc.).

Let $(X_1, ..., X_i, ..., X_n)$ be an n dimensional input and $D[x_i] = \{x_{i,1}, ..., x_{i,l_i}\}$ is a set of values for the input i $(1 \leq i \leq n)$. We extend the set $D[x_i]$ by an additional value $x_{i,0}$ that corresponds to the absence of the value (i.e., an empty input). Evidently, we can extend $D[x_i]$ by a composite value too, for example $(1 \leq i \leq n)$: (a) $x_{i,2}\&x_{i,4}$, (b) $x_{i,1}\&x_{i,3}\&x_{i,l_i}$. Our method deals with the composition of the most important test case(s) for a function cluster testing process (Fig. 6) [14]. We use Hierarchical Morphological Multicriteria Design HMMD [12]. Note compatibility of some two test cases can corresponds to binary relation *independence* or *complementarity.*

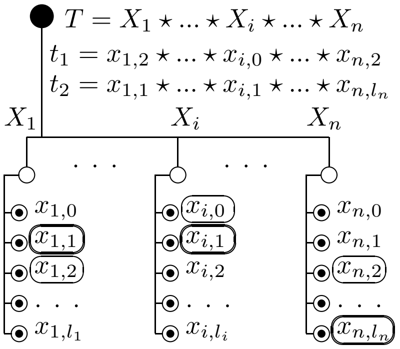

Fig. 6. Composite test case

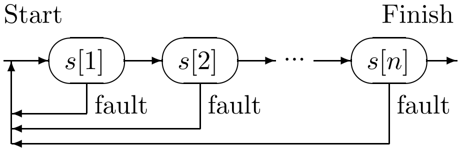

Fig. 7. Illustration for sequencing

1.2.2 Sequencing

Sequencing the test actions can be based on well-known scheduling problems [7], etc.). On the other hand, it is possible to meet here some specific situations. A special sequencing problem for serial manufacturing of a composite product (with corresponding manufacturing operations for each product part) has been proposed in [16]. In this case, each operation is connected with a possible failure that leads to the new start of the process. Thus, it is reasonable to consider the above-mentioned operations as some test actions (with possible feedback). Let $J = \{J_i | 1 \leq i \leq n\}$ be a set of the operations (jobs). Each operation J_i is characterized by an execution time τ_i and probability to find a fault (and to stop the process) η_i. The resultant ordering is: $S =< s[1], ..., s[l], ..., s[n] >$ where $s[l]$ is the number of the operation at position l. The objective function consists in minimization of the average cost (i.e., time) for all operations: $f(S) = \sum_{l=1}^{n} \tau_{s[l]} / \delta_{s[l]}$, $\delta_{s[l]} = \prod_{k=1}^{s[l]} (1 - \eta_k)$. Fig. 7 depicts the problem. This problem can be reduced (a change of variable) to the well-known scheduling problem with the linear penalty function. The algorithm is polynomial $(O(n \log n))$. In the case of precedence constraints for the operations, the algorithm for the problems were proposed too, for example [12]: (a) polynomial algorithm for parallel-series precedence constraints; (b) morphological heuristic for general (an acyclic digraph) precedence constraints (and alternative jobs / operations).

In [6], an analogical problem is examined as scheduling of test cases in an order that increases their effectiveness in meeting some performance goal (e.g., a time for fault detection). In multi-function system testing the sequencing methods can be used for ordering of test cases while taking into account possible test decisions.

1.2.3 Test Case Sequence

Fig. 7 depicts a chain of system function clusters and a corresponding test case sequence: $W =< t_{13} \rightarrow t_{22} \rightarrow t_{32} >$. Here the solving scheme consists of the following two stages: (i) composition of the best test cases for each function cluster (F^l) and (ii) design of the sequence as "the best path" [15].

1.2.4 Collection of Test Case Sequences

Here a set of test case sequences is considered to cover the function cluster digraph [13]. Graph covering is a traditional approach to planning test processes ([9], etc.). In our situation, the *trail* covering problem is used. For the digraph from Fig. 1, the following two solutions exist:

1: $z_1 =< F^1 \rightarrow F^2 \rightarrow F^3 \rightarrow F^1 \rightarrow F^4 >$, $z_2 =< F^3 \rightarrow F^4 >$, and $z_3 =< F^2 \rightarrow F^4 >$.

2: $z_1 =< F^1 \rightarrow F^2 \rightarrow F^3 \rightarrow F^4 >$, $z_2 =< F^1 \rightarrow F^4 >$, $z_3 =< F^2 \rightarrow F^4 >$, and $z_4 =< F^3 \rightarrow F^1 >$.

1.2.5 Structural Fusion

Structural fusion of quality estimates of system units on ordinal scale is illustrated in Figs. 8, 9, 10, and 11 [14]. It is intent the ordinal scale for testing of system units is: "excellent" (1), "quasi-good" (2), and "bad" (3). The figures above involve levels 1 and 2 (by stroke lines). The basic problem is: Find for multi-function situation subclique of nodes (system units) with the estimate level that equals 2. The found set of system units has to be examined as a possible bottleneck. Another problem of structural fusion is: Find for multi-function situation a quasi subclique (without some interconnection / edges). o with the estimate level that equals 2. On the other hand, an ordinal scale with 4 and more levels can be used. In this case, several levels of "quasi-good" results may be examined. This situation leads to other problem formulations (i.e., with other definitions of subcliques).

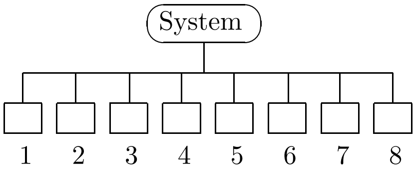

Fig. 8. System structure

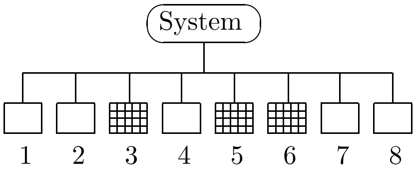

Fig. 9. Results of unit testing

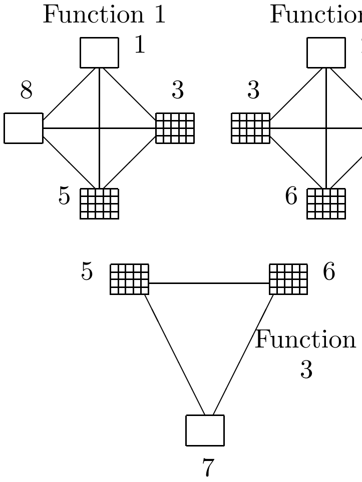

Fig. 10. System functions

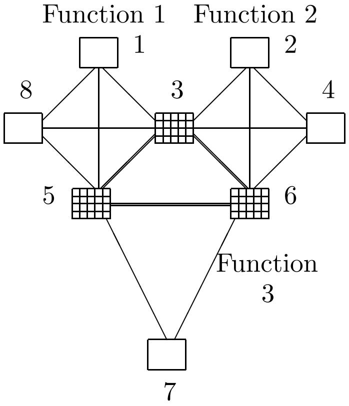

Fig. 11. Cluster of system functions

1.3 Illustrative Example

Let us consider the control center of a missile defense / anti-aircraft system (Fig. 12). Here an input structure is the following (i.e., basic target types and their properties): (i) ballistic missiles X_1; (ii) medium range missiles X_2; (iii) airplanes X_3; (iv) winged missiles X_4; (v) helicopters X_5; (vi) others target types (e.g., gliders) X_6; (vii) an environmental situation (snow, rain, etc.) X_7; and (viii) information on targets from other defense centers X_8. For each input type above, the following attributes are under study: (a) the cardinality of the target set; (b) coordinate sources of their movement; (c) directions; (d) maneuvers; and (e) additions (e.g., impediments, wrong targets). The set of outputs involves the following: (1) the number and types of dangerous targets Y_1; (2) the number and types of the processed (e.g., initialization, identification, maintenance, and knocking off) targets Y_2; and (3) the number and types of the targets which were not knocked off Y_3. Thus we get input and output hierarchies. The system functions are the following (for control center, targets): *1.* scanning the examined area (on the basis of a scheduling of the main radar in multi-function mode) f_1; *2.* initialization of targets f_2; *3.* identification of targets f_3; *4.* tracking and maintenance of targets f_4; *5.* multi-target multi-track assignment f_5; *6.* fair control (assignment of rockets into dangerous targets) f_6; and *7.* deletion of targets from the maintenance process (for non-dangerous targets) f_7. System function clusters are the following: $F^1 = \{f_1\}$; $F^2 = \{f_1, f_4\}$ (for phased array radar); $F^3 = \{f_2, f_3, f_4\}$; $F^4 = \{f_4, f_5\}$; $F^5 = \{f_4, f_7\}$; and $F^6 = \{f_5, f_6\}$. The corresponding function cluster digraph is depicted in Fig. 13. *Trail* covering of the digraph is: $z_1 =< F^1 \rightarrow F^2 \rightarrow F^3 \rightarrow F^4 \rightarrow F^6 >$, $z_2 =< F^2 \rightarrow F^4 \rightarrow F^5 >$, and $z_3 =< F^3 \rightarrow F^5 >$.

Here the defense system is used as an example. This system type can be considered for many domains (e.g., technical maintenance, human resource management, financial analysis of possible investment actions and investment decisions).

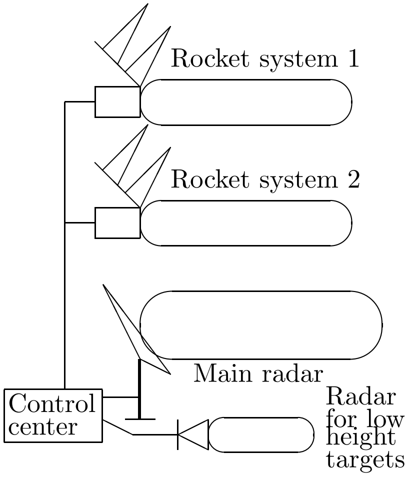

Fig. 12. Applied defense system

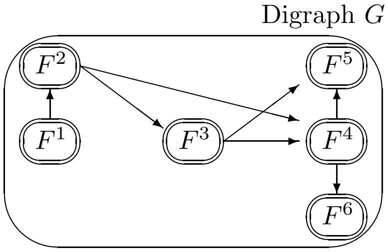

Fig. 13. Function cluster digraph

1.4 Conclusion

In the paper, we have described the potential usage of a set of combinatorial optimization problems in the framework of multi-function system testing. Future research includes additional models and experiments.

1.5 Acknowledgment

This work was partially supported by the National Institute for Systems Test and Productivity at University of South Florida under the USA Space and Naval Warfare Systems Command grant No. N00039-01-1-2248.

Bibliography

[1] Beizer, B., *Black-Box Testing: Techniques for Functional Testing of Software and Systems. Wiley* (1995).

[2] Buede, Dennis M., *The Engineering Design of Systems*, *Wiley* (1999).

[3] Chandrasekaran, B., "Functional representation and causal process", *Advances with Computers 38* (1994) 71-143.

[4] Chen, T.Y., Poon, P.-L., and T.H. Tse, "A choice relation framework for supporting category-partition test case generation", *IEEE Trans. on Software Engineering 29(1)* (2003) 577-593.

[5] Cohen, D.M., Dalal, SD., Parelius, J., and G.C. Patton, "The combinatorial design approach to automatic test generation", *IEEE Software* (Sept. 1996) 83-87.

[6] Elbaum, S., Malishevsky, A.G., and G. Rothermel, "Prioritizing test cases for regression testing", *ACM SIGSOFT Software Eng. Notes 25(5)* (2000) 102-112.

[7] Garey, M.R., and D.S. Johnson, *Computers and Intractability. The Guide to the Theory of NP-Completeness.* Freeman&Co. (1979).

[8] Jorgensen, P.C., *Software Testing. A Craftman's Approach*, 2nd ed., CRC Press (2002).

[9] Karpovsky, M.G., and E.A. Moskalev, "Covering of edges of graph by a minimal set of paths", *Discr. Math. 58(2)* (1986) 214.

[10] Kusiak, Andrew P., *Engineering Design: Products, Processes, and Systems.* Academic Press (1999).

[11] Last, Mark, and Abe Kandel, "Automated test reduction using an Info-Fuzzy Network", *Annals of Software Engineering* (2003) 235-258.

[12] Levin, Mark Sh., *Combinatorial Engineering of Decomposable Systems.* Kluwer (1998).

[13] Levin, Mark Sh., and Mark Last, "Collection of test case sequences: Covering of function cluster digraph", *IASTED Int. Symp. on Artificial Intelligence and Applications AIA'2004*, (2004) 806-811.

[14] Levin, Mark Sh., and Mark Last, "Multi-function system testing: Composition of Test Sets", *8th IEEE Int. Symp. on High Assurance Systems Engineering HASE'04*, IEEE Press (2004) 99–108.

[15] Levin, Mark Sh. and Mark Last, "Test case sequences in system testing: Selection of test cases for a chain (sequence) of function clusters", *17th Int. Conf. IEA/AIE 2004*, (2004) (in press).

[16] Livshitz, Emmanuil M., "Sequence of operations for the fabrication of complex parts", *Aut. and Remote Control 29(3)* (1968) 1807-1808.

[17] Prasad, B., *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*, Englewood Cliffs (1996).

[18] Sage A.P., and W.B. Rouse, (Eds.) *Handbook of Systems Engineering and Management*, Wiley (1999).

[19] Schroeder, P.J., and B. Korel, "Black-box test reduction using Input-Output analysis", *ACM SIGSOFT Software Eng. Notes 25(5)* (2000) 173-177.