INFORMATICA, 2012, Vol. 23, No. 3, 391–404

Towards Clique-Based Fusion of Graph Streams in Multi-Function System Testing

Mark Sh. LEVIN
Institute for Information Transmission Problems, Russian Academy of Sciences
19 Bolshoj Karetny Lane, Moscow 127994, Russia
e-mail: mslevin@acm.org

Received: July 2011; accepted: March 2012

Abstract. The article describes multi-function system testing based on fusion (or revelation) of clique-like structures. The following sets are considered: (i) subsystems (system parts or units/components/modules), (ii) system functions and a subset of system components for each system function, and (iii) function clusters (some groups of system functions which are used jointly). Test procedures (as units testing) are used for each subsystem. The procedures lead to an ordinal result (states, colors) for each component (e.g., 'out of service', 'major faults', 'minor faults', 'trouble free service'). For each system function a graph over corresponding system components is examined while taking into account ordinal estimates/colors of the components. Further, an integrated graph for each function cluster is considered (this graph integrates the graphs for corresponding system functions). For the integrated graph structure revelation problems are under examination (revelation of some subgraphs which can lead to system faults). Numerical examples illustrate the approach and problems.

Keywords: modular systems, system testing, data fusion, data streams, graphs, clique, combinatorial optimization, heuristics.

1. Introduction

In many contemporary complex systems, the significance of function system testing is increasing (Chittarq and Rannon, 1999; Levin and Last, 2006). In two recent decades, a central role of multi-function system testing has been pointed out in many studies (Cohen *et al.*, 1996; Levin and Last, 2006; Offutt, 1992). This situation is based on integration (fusion) of local faults into a general system faults. Here a local fault corresponds to a system unit/components and each local fault is a result of a certain system function. Often coordination of concurrent system functions can be organized at a insufficient level. In many recent well-known system faults (e.g., power stations, offshore drilling platforms, airplane crashes), coordination among concurrent system functions was not OK (by computer systems, by maintenance documentations, etc.). In contemporary distributed complex systems, there exist many different users (they are not often coordinated), many different support teams (i.e., maintenance, management; they can be coordinated at insufficient level), and many different processes (they can be coordinated at insufficient level). Evidently, the applied situations for complex distributed systems

have to be examined via various approaches. Thus, a system analysis of combinations for concurrent system functions is a crucial direction in system testing (Cohen *et al.*, 1996; Levin and Last, 2006; Offutt, 1992).

In this article, the above-mentioned type of applied situations for complex distributed systems is considered and described (i.e., problem, models, solving scheme). A general framework of data streams integration is depicted in Fig. 1. Our framework is based on clique-based fusion of graph streams for multi-function system testing (Levin, 2007; Levin and Last 2004, 2006).

Now let us consider a simplified example for a modular system consisting of the following components: basic facility s_1, control subsystem s_2, safety subsystem s_3, utilization personnel s_4, maintenance/testing personnel s_5, and personnel for remote control s_6. Three functions are examined (Fig. 2): utilization function f_1: $\{s_1, s_2, s_3, s_4\}$, maintenance/testing function f_2: $\{s_1, s_2, s_3, s_5\}$, remote control function f_3: $\{s_1, s_2, s_6\}$. In the case when a coordination between the above-mentioned concurrent functions is wrong, the following situation can be met: (i) basic facility s_1 is out of service (by utilization personnel s_4, i.e., via function f_1); (ii) safety subsystem s_3 is out of service (by an action of maintenance/testing personnel s_5, i.e., via function f_2); and (iii) control subsystem s_2 is under a wrong mode (by a wrong action of personnel for remote control s_6, i.e., via function f_3). Figure 2 depicts the examined situation (ordinal estimates of graph vertices

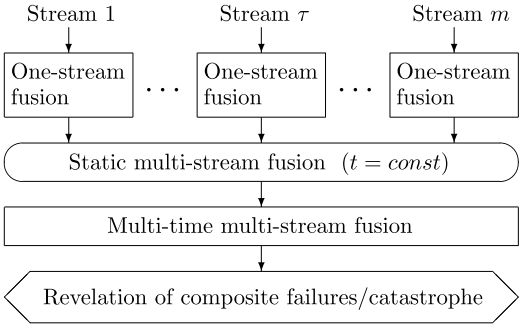


Fig. 1. Framework of data streams integration.

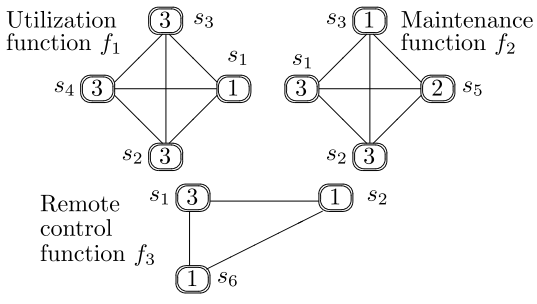


Fig. 2. Illustration for simplified example.

are shown in circles; the following estimates for nodes are used: 1 corresponds to a wrong mode, 2 corresponds to about 'OK', 3 corresponds to 'OK').

As a result, a combination of interconnected system components (as clique) $\{s_1, s_2, s_3\}$ will lead to a combined system fault.

Our approach to system testing is based on a layered scheme (Fig. 3) that is adopted from Levin and Last (2004, 2006).

Generally, the system testing framework consists of several stages as follows (Levin, 2007):

Stage 1. Unit/component testing (Kaner *et al.*, 1999).

Stage 2. Analysis of system functions, their interconnections, and design of function clusters (functions which are executed jointly/concurrently, i.e., at the same time moment; ((?), (?)).

Stage 3. Design of an integrated graph over system components for each function cluster.

Stage 4. Revelation of clique (or quasi-clique) in the integrated graph (Osteen and Tou, 1973; Jiang and Pei, 2009; Pei *et al.*, 2005).

A simplified framework of the testing process is depicted in Fig. 4 (an integrated graph corresponds to function cluster $F' = \{f_1, \ldots, f_\xi, \ldots, f_\lambda\}$). Revelation of clique in the integrated graph is considered as a fusion problem or structure mining (Coble *et al.*, 2006; Inokuchi *et al.*, 2003).

Thus, our framework is based on revelation (mining) of cross-graphs cliques or quasi-cliques (Jiang and Pei, 2009; Levin, 2007; Levin and Last, 2006; Pei *et al.*, 2005). Many

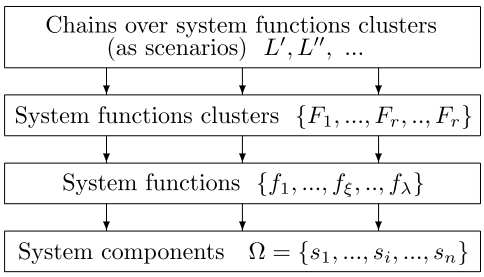


Fig. 3. Layered system testing scheme.

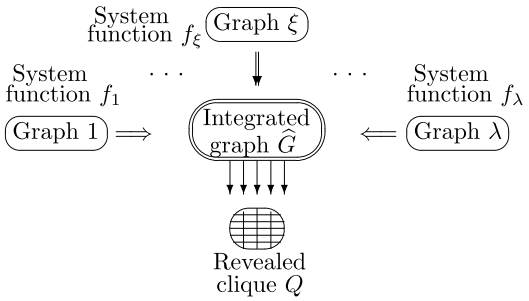


Fig. 4. Revelation of clique (quasi-clique).

well-known traditional methods may be used for the analysis and revelation of cliques (or quasi-clique) in graphs (Abello *et al.*, 2002; Bomze *et al.*, 1999; Garey and Johnson, 1979; Jiang and Pei, 2009; Johnson and Trick, 1996; Osteen and Tou, 1973; Pardalos and Xue, 1994; Pei *et al.*, 2005).

A preliminary version of the article has been published as an electronic preprint (Levin, 2011).

2. Preliminaries

Here the basic sets under examination are described (Levin, 2007; Levin and Last, 2006). Let $\Omega = \{s_1, \ldots, s_i, \ldots, s_n\}$ be a set of subsystems or main system components. Let $f = \{f_1, \ldots, f_\xi, \ldots, f_\lambda\}$ be a set of system functions. For each system function f_ξ there is the following: (i) a subset of $\Omega(f_\xi) \subseteq \Omega$ that consists of components which are used for system function f_ξ and (ii) a graph (usually a complete graph) over elements of $\Omega(f_\xi)$ as $G(f_\xi) = (\Omega(f_\xi), E(f_\xi))$.

There is a set of system function clusters $F = \{F_1, \ldots, F_r, \ldots, F_p\}$ where $F_r \subseteq f$; each system function cluster F_r ($r = \overline{1,p}$) is a subset of system function set F and for each system function cluster its elements (i.e., corresponding functions) are executed together at the same time moment. For each system function cluster F_r it is reasonable to examine the corresponding integrated graph as follows: $G(F_r) = G(\Omega(F_r), E(F_r)) = \cup_{f_j \in F_r} G(f_j)$, where $\Omega(F_r) = \cup_{f_j \in F_r} \Omega(f_j)$ and $E(F_r) = \cup_{f_j \in F_r} E(f_j)$.

For each subsystem s_i a system test procedure (as unit test) is used and the procedures lead to ordinal results (i.e., states, colors) for each system component, e.g., $[1, 2, 3, 4]$ (where 1 corresponds to "out of service", 2 corresponds to "major faults", 3 corresponds to "minor faults", 4 corresponds to "trouble free service"). As a result, the graphs with ordinal weights of vertices (or colored graphs) are obtained. Finally, for each function cluster F_r we can examine the corresponding colored integrated graph $\widehat{G}(F_r) = \widehat{G}(S(F_r), E(F_r))$.

In more complicated situation, the unit test results can be different for different system functions. Then the integration process is based on the following rule: in the case of difference of colors for the same vertex of graphs for different system functions (e.g., f_{j_1} and f_{j_2}) the 'worst' color is selected.

Let Q_h be a clique over h vertices (e.g., Q_4, here the estimate of each clique vertex is: "$<=l$", $l = 1, 2, 3, 4$, and "quasiness" or "approximation" (by number of vertices or by estimates of vertices) will be denoted by "widetilde" (e.g., $\widetilde{Q}_4$). Thus, in the integrated colored graph $\widehat{G}(F_r)$ the following kinds of substructures (subgraphs) are under examination (by a rule: in the structure each vertex color "$= 1$", "$<= 2$", etc.). Figure 5 illustrates 4-vertex structure (ordinal estimates of graph vertices are pointed out in circles):

1. Clique, dimension of the clique equals (or more than) the number of functions in F_r (Fig. 5a): Q_r.
2. Quasi-clique by edges/interconnection (an edge is absent; Fig. 5b): Φ_r.
3. Quasi-clique by vertices (in the revealed subgraph not all vertices have estimate "$<=l$", $l = 1, 2, \ldots$ (Fig. 5c): Θ_r.

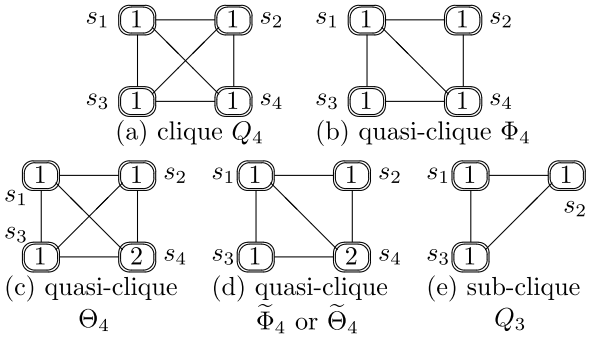


Fig. 5. Examples of clique and quasi-cliques.

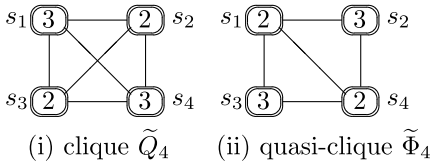


Fig. 6. Examples of system faults.

4. Quasi-clique by vertices and by edges (Fig. 5d): $\widetilde{\Theta}_r$ or $\widetilde{\Phi}_r$.
5. Sub-clique or clique with less dimension (i.e, the number of vertices in the clique v is less than the number of functions in F_r (Fig. 5e): $Q_v, v < p$).
6. Quasi sub-clique (structural approximation): (i) by vertices, (ii) by edges, (iii) by vertices and edges ($\widetilde{Q}$, etc.).

It is reasonable to point out the following:

I. A situation when the estimate of system components equals 1 (i.e., "out of service") is crucial and can lead to a system fault. This kind of situation is a "traditional" one in system testing.
II. A situation when several interconnected (by time and/or system work) system components have estimates at a "medium level" (e.g., "major fault" or "minor faults") can lead to a system fault in complex systems (Cohen *et al.*, 1996). Thus, our main efforts in this article are targeted to this kind of a system situation (Fig. 6, here estimates of vertices are: "<=3"). On the other hand, the further examination can be mainly based for estimates of system components at the levels 1 and 2 because after a shift of the ordinal evaluation scale this kind of mathematical problems will be obtained.

3. Basic Problem

Here a basic problem (*Problem 1*) is described. Structural fusion of quality estimates for system units/components upon ordinal scale is illustrated in Figs. 7, 8, and 9.

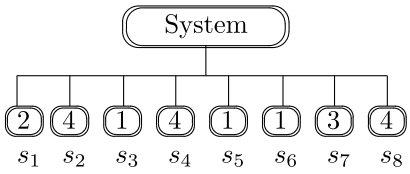


Fig. 7. System structure and estimates.

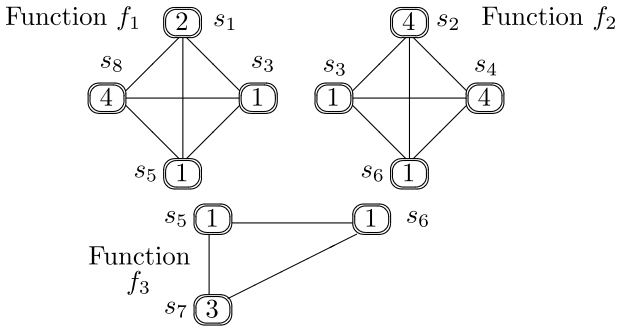


Fig. 8. Cluster of system functions.

The estimates for system components are depicted in Fig. 7 (components s_3, s_5, and s_6: 1; component s_1: 2; and component s_7: 3; other components: 4). The basic problem (*Problem 1*) is:

> *Find for multi-function situation (in the integrated graph for function cluster F_r) clique Q_h (number of nodes equals the number of functions in cluster F_r or more with the estimate level* "<=l" ($l = 1, 2, \ldots$).

Thus, the well-known clique problem is considered:

Find the largest clique (complete subgraph) Q in an undirected graph.

The obtained set of system vertices/units has to be examined as "critical unit subset" or "system syndrome" (analogue of "syndrome" in medicine). In Figs. 7 and 8, $F_r = \{f_1, f_2, f_3\}$. Further, in Fig. 9 the clique vertices are the following: $\{s_3, s_5, s_6\}$. Here estimates of the vertices above equal 1.

Clearly, the clique Q_λ may be absent. In this case it may be important to search for quasi-clique $\widetilde{Q}_\lambda$ or cliques which contain less number of vertices.

Thus, another problem of structural fusion is (*Problem 1a*):

> *Find quasi-clique $\widetilde{Q}_\lambda$ for the multi-function situation, i.e., without some interconnection/edges or/and with the estimate level*: "<= l", "1 <= l <= 3".

Examples of quasi-cliques (vertex sets) are the following (Fig. 9): (a) $\{s_1, s_3, s_5, s_6\}$, estimates: "<=2"; (b) $\{s_1, s_3, s_5, s_6, s_7\}$, estimates: "<=3". Note, it may be often possible (and reasonable) to reveal several cliques (or quasi-cliques).

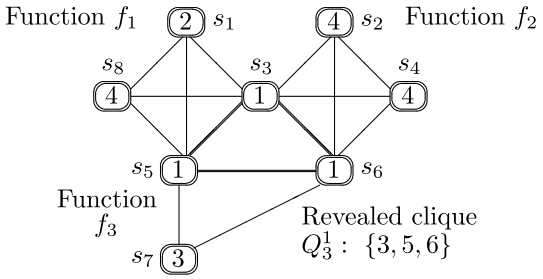


Fig. 9. Integrated graph for function cluster.

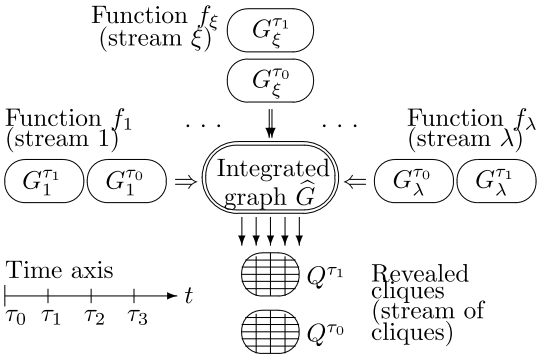


Fig. 10. Revelation of clique over graph streams.

4. Additional Problems

4.1. *Basic Sequences/Streams*

In the case of graph streams, an illustration is depicted in Fig. 10.

Here a time axis is considered as follows: $t = \{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \ldots\}$. As a result, the following is examined:

(i) sequence of states for system components ($j = \overline{1,k}$):
$s_j(t) = \{s_j^{\tau_0}, s_j^{\tau_1}, s_j^{\tau_2}, s_j^{\tau_3}, s_j^{\tau_4}, s_j^{\tau_5}, \ldots\}$, where $s_j(\tau_\eta) \in \{0,1\}$ ($\eta = 0,1,2,3,\ldots$), $s_j(\tau_\eta) = 1$ if function j is used at time τ_η and $s_j(\tau_\eta) = 0$ otherwise;

(ii) sequence of system functions ($\xi = \overline{1,\lambda}$):
$f_\xi(t) = \{f_\xi^{\tau_0}, f_\xi^{\tau_1}, f_\xi^{\tau_2}, f_\xi^{\tau_3}, f_\xi^{\tau_4}, f_\xi^{\tau_5}, \ldots\}$, where $s_j(\tau_\eta) \in \{0,1\}$, $s_j(\tau_\eta) = 1$ if function j is used at time τ_η and $s_j(\tau_\eta) = 0$ otherwise);

(iii) sequence of graphs for system functions ($\xi = \overline{1,\lambda}$):
$G_{f_\xi}(t) = \{G_{f_\xi}^{\tau_0}, G_{f_\xi}^{\tau_1}, G_{f_\xi}^{\tau_2}, G_{f_\xi}^{\tau_3}, G_{f_\xi}^{\tau_4}, G_{f_\xi}^{\tau_5}, \ldots\}$ (if $s_j^{\tau_\eta} = 0$ the corresponding graph $G_{f_\xi}^{\tau_\eta}$ is empty);

(iv) sequence of graphs for system function cluster ($r = \overline{1,p}$):
$G_{F_r}(t) = \{G_{F_r}^{\tau_0}, G_{F_r}^{\tau_1}, G_{F_r}^{\tau_2}, G_{F_r}^{\tau_3}, G_{F_r}^{\tau_4}, G_{F_r}^{\tau_5}, \ldots\}$ (if $s_j^{\tau_\eta} = 0$ the corresponding graph $G_{F_r}^{\tau_\eta}$ is empty).

Here a chain of system function clusters (e.g., $L = \langle F', F'', F''' \rangle$) is considered as a scenario (in general, scenario can have a more complicated type, e.g., tree, network).

4.2. *Other Problems*

The set of additional problems involves the following:

Problem 2. Revelation of clique when the number of vertices is less than the number of functions in the function cluster (i.e., sub-clique).

Problem 3. Dynamical problems (vertex colors are functions of time): *3.1.* existence of a time interval where clique exists; *3.2.* existence of a time interval where quasi-clique exists.

Problem 4. Analysis of time intervals when clique (or quasi-clique) exists and maintenance of the clique (quasi-clique) as some "critical" structure (substructure). As a result, a track for a special structure (e.g., clique Q) can be obtained: T_Q.

Problem 5. Design of actions as composite plans to destroy the critical substructure(s) (i.e., clique(s), quasi-clique(s)).

5. Example

Let us consider a numerical example. Tables 1 and 2 contain a description of the examined sets of system functions and function clusters: $\{f_1, f_2, f_3, f_4, f_5\}$ and $\{F_1, F_2, F_3\}$.

Table 1

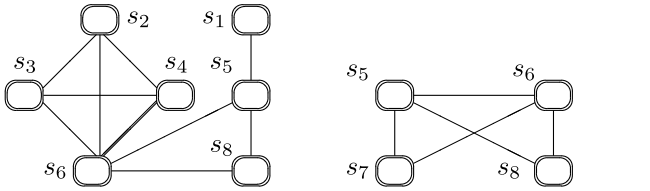
System functions

System function	System components							
	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
f_1	*		*		*			*
f_2		*	*	*		*		
f_3					*	*	*	
f_4	*				*			
f_5					*	*		*

Table 2

System function clusters

System function clusters	System functions				
	f_1	f_2	f_3	f_4	f_5
F_1	*	*	*		
F_2		*		*	*
F_3			*		*



(a) integrated graph $\widehat{G}(F_2)$ (b) integrated graph $\widehat{G}(F_3)$

Fig. 11. Integrated graphs for function clusters.

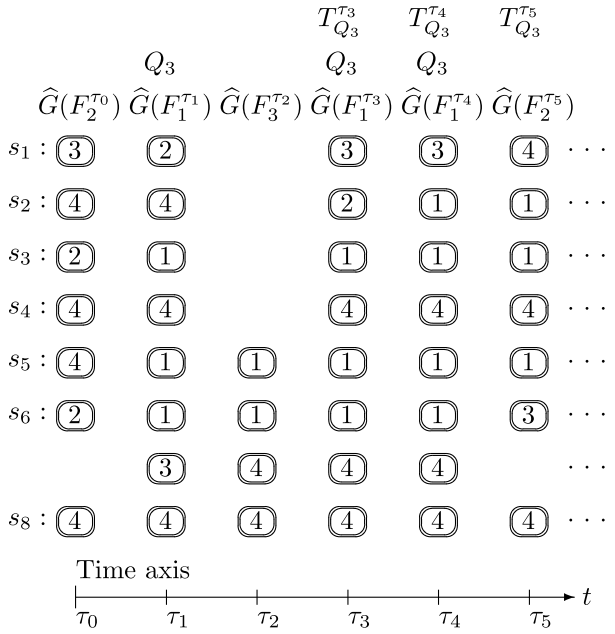


Fig. 12. Example of graph streams.

The following time axis is considered: $\{\tau_0, \tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$ (i.e., $\tau_\eta, \eta = \overline{0,5}$). The function cluster chain (a scenario) is as follows:

$$L' = \langle F_2^{\tau_0}, F_1^{\tau_1}, F_3^{\tau_2}, F_1^{\tau_3}, F_1^{\tau_4}, F_2^{\tau_5} \rangle,$$

where *upper index* corresponds to time moment. Figure 9 depicts integrated graph $\widehat{G}(F_1)$, Fig. 11 depicts two integrated graphs: $\widehat{G}(F_2)$ and $\widehat{G}(F_3)$.

Further, the following basic problems are under examination: Problems 3, 4, 5. Figure 12 depicts state streams for system components (vertices) s_1, s_2, s_3, s_4, s_5, s_5, s_6, s_7, and s_8 while taking into account time axis above. In addition, Fig. 12 contains the following: (i) integrated graphs $\{\widehat{G}\}$, (ii) revealed structures (here: Q_3), and (iii) obtained tracks of the revealed structures (here: T_{Q_3}).

Now the following solutions can be pointed out:

Problem 3.1. Clique Q_3: $\{s_3, s_5, s_6\}$ can be revealed for time: τ_1, τ_3, τ_4 (estimates of vertices equal 1).

Problem 4. For time interval $[\tau_1, \tau_5]$ there exists a structure with vertices: $\{3, 5, 6\}$ while taking into account a well-known engineering "track initiation rule: 2 from 3" (i.e., for time interval with length 3 clique is revealed 2 times). As a result, it is reasonable to initiate at time moment τ_3 clique Q_3: $\{s_3, s_5, s_6\}$ (estimates of vertices equal 1). Note, the rule above (the rules of these kind) is used as "track maintenance rule" as well. After that it is possible to maintain this structure (by the rule above) as an event (i.e., to check the initiation rule above at each discrete time moment, for example: τ_4, τ_5).

Problem 5. Clearly, to destroy the event above (i.e., $T_{Q_3}^{\tau_3}$, $T_{Q_3}^{\tau_4}$, $T_{Q_3}^{\tau_5}$) it is necessary to destroy clique Q_3 at time moment τ_3 by improvement of state for s_5 (or s_6) (i.e., improvement of the estimate: $1 \to 2$ or $1 \to 3$ or $1 \to 4$).

6. Generalized Testing Scheme

A generalized testing scheme is presented in Fig. 13. Evidently, the generation of test inputs as sequences $\{s_j(t),\ j = \overline{1,k}\}$ can be based on two methods: (a) previous practice, (b) special simulation approaches (e.g., Monte Carlo or quasi Monte Carlo methods (Sobol, 1994).

The suggested system testing framework can be used for system maintenance as well (Fig. 14): (1) evaluation of system components $\{s_j\}$; (2) revelation of the executed system function clusters (i.e., F_r); (3) revelation of integrated graphs, corresponding "critical" clique-like structures, and the structure tracks; and (4) analysis and planning of system maintenance: (a) prediction: for system components and for whole system, (b) design of maintenance plan (e.g., improvement actions for system components).

7. Discussion and Conclusions

In the article, a new system framework for multi-function system testing has been suggested. The positive property of the framework consists in the following: concurrent system functions can be analyzed. The system framework is a basis for modification and adaptation. In general, it is reasonable to point out the significance of data stream systems (Babcock *et al.*, 2002; Coble *et al.*, 2006; Shoubridge *et al.*, 2002; Sun *et al.*, 2010) which are widely used in many domains (data/knowledge summarization, image processing, system reliability analysis, initiation of target tracks in sensor systems, etc.).

The key parameters for the systems above involve the following: (1) number of streams (one, many), (2) type of data, e.g., values (binary, ordinal, continuous), structures (i.e., preferences, graphs), (3) size of time window (i.e., number of series time moments which are jointly analyzed). A simplified typology of the systems may be considered as follows (Babcock *et al.*, 2002; Griffith, 1986; Shoubridge *et al.*, 2002; Levin, 2001; Levitin, 2003; Nyberg, 2006; Shoubridge *et al.*, 2002; Sun *et al.*, 2010):

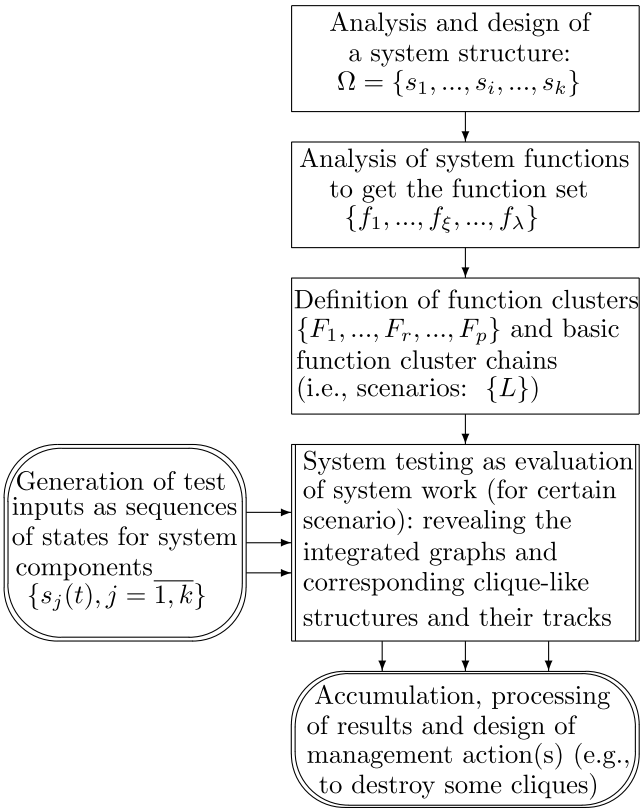


Fig. 13. Generalized testing scheme.

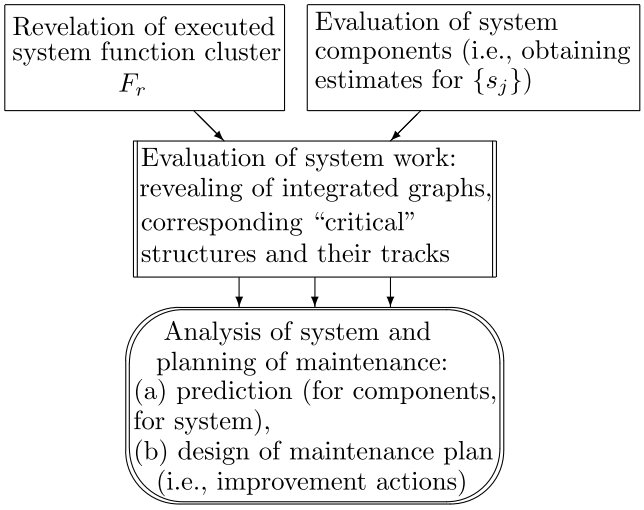


Fig. 14. Scheme of system maintenance.

(a) static case for m streams: (i) summarization of values (binary, ordinal, continuous); processing methods: histograms, rule "k of m", diagnosis techniques (e.g., closeness to centers of specified clusters; Babcock *et al.*, 2002; Levitin, 2003;) (ii) aggregation of structures (e.g., decision making, knowledge engineering, image processing): building a maximum substructure (e.g., consensus, median), building a minimum superstructure (Cook and Kress, 1992; Ferrer *et al.*, 2010; Herrera-Viedma *et al.*, 2002; Jiang *et al.*, 2001; Kemeny and Snell, 1972; Levin, 2001; Levin, 2003; Shoubridge *et al.*, 2002; McGregor, 1982);

(b) dynamic case (one stream, window for n time moments); processing methods, for example: rule "k of m" (e.g., an engineering technique for initiation/maintenance of target tracks, analysis of system reliability, fusion of image sequences, revelation of patterns from time series of graphs) (Bar-Shalom and Portman, 1988; Coble *et al.*, 2006; Shoubridge *et al.*, 2002); and

(c) combined dynamic case (m streams and window for n time moments); composite processing methods (e.g., dynamic decision making based on Markov decision processes or dynamic decision networks) (Bar-Shalom and Portman, 1988; Blackman and Popoli, 1999; Brooks and Iyegar, 1998); Brooks *et al.*, 2002; Da Costa and Buede, 2000).

Three possible evident strategies can be used in Case c above:

Strategy 1: (i) Integration of data for each stream by a time window (Case b), (ii) summarization of results for m streams (Case a).

Strategy 2: (i) Summarization of data of m streams at each time moment (Case a); (ii) integration of results via a time window (Case b).

Strategy 3: combined scheme.

Our suggested framework implements *strategy 2* above: (i) fusion of graphs at each time moment with revelation of clique (or quasi-clique), and (ii) usage of rule "k of m".

Note, our material has only a preliminary character as the first step and it is targeted to problem(s) formulation and solving scheme description (i.e., a new system framework). Future research efforts could include various investigations, for example:

1. study of the problems with some weights of system components or/and their faults;
2. study of the problems while taking into account weights of interconnection of system components (i.e., weights of edges in integrated graph);
3. exploration of cliques (quasi-cliques) as special kinds of composite events (as "generalized system syndromes") (to generate a set of possible composite systems faults);
4. usage of probabilistic and/or fuzzy estimates;
5. study of stability issues for considered solving schemes;
6. special study of the suggested framework for system maintenance;
7. design of a special simulation computer environment based on the suggested framework;
8. consideration of real world examples for various application domains;
9. usage of the described framework as a basis for an educational laboratory environment; and
10. usage of the suggested approach for designing a logical computer game.

References

Abello, J., Resende, M.G.C., Sudarsky, S. (2002). Massive quasi-clique detection. In: Rajsbaum, S. (Ed.) *LATIN 2002: Theoretical Informatics*, LNCS, Vol. 2286. Springer, Heidelberg, pp. 598–612.

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J. (2002). Models and issues in data stream systems. In: *Proceedings of 21st ACM SIGMOD-SIGACT-SUGART Symposium Principles of Database Systems PDOS-2002*, Madison, WI, USA, pp. 1–16.

Bar-Shalom, Y., Portman, T.E. (1988). *Tracking and Data Association.* Academic, New York.

Blackman, S., Popoli, R. (1999). *Design and Analysis of Modern Tracking Systems.* Artech House, Boston.

Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M. (1999). The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (Eds.), *Handbook of Combinatorial Optimization* (Supp. Vol. A). Springer, New York, pp. 659–729.

Brooks, R.R., Iyengar, S.S. (1998). *Multi-Sensor Fusion: Fundamentals and Applications with Software*. Prentice Hall, Englewood Cliffs.

Brooks, R.R., Ramanathan, P., Sayeed, A.M. (2002). Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8), 1163–1171.

Chittarq, I., Rannon, R. (1999). Diagnosis of multiple faults with flow-based functional models: the functional diagnosis with efforts and flows approach. *Reliability Engineering and System Safety*, 64(2), 137–150.

Coble, J., Cook, D.J., Holder, L.B. (2006). Structure discovery in sequentially-connected data streams. *International Journal on Artificial Intelligence Tools*, 15(6), 917–944.

Cohen, D.M., Dalal, S.R., Parelius, J., Patton, G.C. (1996). The combinatorial design approach to automatic test generation. *IEEE Software*, 13(5), 83–87.

Cook, W.D., Kress, M. (1992). *Ordinal Information and Preference Structures: Decision Models and Applications*. Prentice Hall, Englewood Cliffs.

Da Costa, P.C.G., Buede, D.B. (2000). Dynamic decision making: a comparison of approaches. *Journal of Multi-Criteria Decision Analysis*, 9(6), 243–262.

Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., Bunke, H. (2010). Generalized median graph computation for means of graph embedding in vector space. *Pattern Recognition*, 43(4), 1642–1655.

Garey, M.R., Johnson, D.S. (1979). *Computers and Intractability. The Guide to the Theory of NP-Completeness*. Freeman and Company, New York.

Griffith, W. (1986). On consecutive k-out-of-n failure systems and their generalizaitons. In: Basu, A.P. (Ed.), *Reliability and Quality Control*. Elsevier, Amsterdam, pp. 157–165.

Herrera-Viedma, E., Herrera, F., Chiclana, F. (2002). A consensus model for multiperson decision making with different preference structures. *IEEE Transactions SMC, Part A*, 32(3), 394–402.

Inokuchi, A., Washio, T., Motoda, H. (2003). Complete mining of frequent patterns from data: mining graph data. *Machine Learning*, 50(3), 321–354.

Jiang, D., Pei, J. (2009). Mining frequent cross-graph quasi-cliques. *ACM Transactions on Knowledge Discovery from Data*, 2(4), 16, 1–42.

Jiang, X., Munge, A., Bunke, H. (2001). On median graphs: properties, algorithms, and applications. *IEEE Transactions PAMI*, 23(10), 1144–1151.

Johnson, D.S., Trick, M.A. (Eds.) (1996). *Cliques, Coloring, and Satisfiability*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 26. AMS, Providence.

Kaner, C., Falk, J., Nguyen, H.Q. (1999). *Testing Computer Software*, 2nd edn., Wiley, New York.

Kemeny, J.G., Snell, J.L. (1972). *Mathematical Models in the Social Sciences.* MIT Press, Cambridge.

Levin, M.Sh. (2001). System synthesis with morphological clique problem: fusion of subsystem evaluation decisions. *Information Fusion*, 2(3), 225–237.

Levin, M.Sh. (2003) Common part of preference relations. *Foundations of Computing & Dec. Sciences*, 28(4), 223–246.

Levin, M.Sh. (2007) Clique-based structural fusion: multi-function system testing (position paper). In: *Complementary Volume of International Conference TestCom/FATES 2007*, Tallinn, Estonia, pp. 18–21.

Levin, M.Sh. (2011). Framework for clique-based fusion of graph streams in multi-function system testing. Electronic preprint, `http://arxiv.org/abs/1103.3807` [cs.DS]

Levin, M.Sh., Last, M. (2004). Multi-function system testing: composition of test sets. In: *Proceedings of 8th IEEE International Conference on High Assurance Systems Engineering (HASE'04)*, Tampa, USA, pp. 99–108.

Levin, M.Sh., Last, M. (2006). Design of test inputs and their sequences in multi-funciton system testing. *Applied Intelligence,* 25(1), 544–553.

Levitin, G. (2003). Linear multi-state sliding window systems. *IEEE Trans. Relaibility*, 52(2), 263–269.

McGregor, J.J. (1982). Backtracking search algorithms and the maximum common subgraph problem. *Software – Practice and Experience*, 12(1), 23–34.

Nyberg, M. (2006) A fault isolation algorithm for the case of multiple faults and multiple fault types. In: *Proceedings of 6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Safeprocess'06*, Beijing, China, Vol. 6, Part 1.

Offutt, A.J. (1992). Investigation of the software testing coupling effect. *ACM Transactions on Software Engineering and Methodology*, 1(1), 5–20.

Osteen, R.E., Tou, J.T. (1973). A clique-detection algorithm based on neighbourhoods in graphs. *International Journal of Computers and Information Sciences*, 2(4), 257–268.

Pardalos, P.M., Xue, J. (1994), The maximum clique problem. *Journal of Global Optimization*, 4(3), 301–328.

Pei, J., Jiang, D., Zhang, A. (2005). On mining cross-graph quasi-cliques. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD'05*, Chicago, USA, pp. 228–238.

Shoubridge, P., Kraetzl, M., Wallis, W., Bunke, H. (2002). Detection of abnormal change in time series of graphs. *Journal of Interconnnection Networks*, 3(1–2), 85–101.

Sobol, I.M. (1994). *A Primer for the Monte Carlo Method.* CRC Press, Boca Raton.

Sun, A., Zeng, D.D., Hsinchun, C. (2010). Burst detection from multiple data streams: a network-based approach, *IEEE Transactions SMC, Part C*, 40(3), 258–267.

M.Sh. Levin is with the Institute for Information Transmission Problems of the Russian Academy of Sciences He received the MS degree in radio engineering from Moscow Technical University of Communications and Informatics (1970), the MS degree in mathematics from Lomonosov Moscow State University (1975), the PhD degree in systems analysis from Russian Academy of Sciences (1982). Dr. Levin's interests include system design and testing, networking, combinatorial optimization, decision making. Dr. Levin is a member of IEEE, ACM(SM'06), SIAM, INCOSE. More information can be found at `http://www.mslevin.iitp.ru/`.

Apie klikų tipo grafų srautų suliejimo taikymą daugiafunkcinių sistemų teatavimui

Mark Sh. LEVIN

Straipsnyje pristatomas daugiafunkcinių sistemų testavimas suliejant klikų tipo struktūras. Nagrinėjamos šitokios aibės: (i) posistemės (sistemos dalys/komponentai/moduliai), (ii) sistemos funkcijos ir sistemos komponentų poaibis skirtas realizuoti kiekvienai sistemos funkcijai, (iii) funkcijų klasteriai (funkcijų, vykdomų kartu, grupės). Testavimo procedūros vykdomos kiekvienai posistemei. Procedūrų rezultatai kiekvienam komponentui išreiškiami ordinalioje skalėje (pvz., būviai, spalvos). Kiekvienai sistemos funkcijai analizuojamas sistemos komponentų grafas atsižvelgiant į jų ordinalius įverčius/spalvas. Vėliau nagrinėjamas integruotas klasterių grafas. Integruotų grafų struktūrose ieškoma subgrafų, kurie gali sąlygoti sistemos klaidas. Pasiūlyta metodika iliustruojama skaitmeniniais pavyzdžiais.