Optimal Selection of Diversity Types for Safety-Critical Computer Systems

Vyacheslav Kharchenko[a, b], Tetyana Nikitina[a], and Sergiy Vilkomir[c]
[a]National Aerospace University named after N.E. Zhukovsky "KhAI", Kharkiv, Ukraine
[b]Centre for Safety Infrastructure-Oriented Research and Analysis, Kharkiv, Ukraine
[c]East Carolina University, Greenville, NC, USA

Abstract: An important task in the development of safety-critical computer systems is achieving high levels of reliability and safety. To protect such systems from common-cause failures that can lead to potentially dangerous outcomes, special methods are applied, including multi-version technologies operating at different levels and volumes of diversity. In this article, we solve the problem of finding an optimal design decision at minimum cost with the required diversity or maximum diversity level with assumed cost. The proposed multi-version model takes into consideration the dependencies among diversity types, diversity metrics, and costs. The model presents a decision for each version of a two-version system. The model can be used to make an optimal design decision with various types of diversity during software-based multi-version system development.

Keywords: diversity, multi-version system, safety-critical computer systems, common-cause failures.

1. INTRODUCTION

An important task in the development of safety-critical computer systems is achieving high levels of reliability and safety. To protect safety-critical systems from common-cause failures (CCF) that can lead to potentially dangerous outcomes, special methods are applied, including multi-version technologies operating at different levels of diversity and with different diversity types.

Safety-critical system failures are due to two factors: physical faults and design faults. Due to physical faults, hardware failures occur that are the result of the manifestation of degradation mechanisms. These failures may be tolerated by using different structure redundancy types. Design faults can be eliminated only by changes in the project or the production process, documentation, etc. To compensate for design faults, it is appropriate to apply versioning when the same system function is performed in different ways. However, in the case of version redundancy, faults may occur for both versions of systems.

According to various estimates, the number of faults in the two-version system can range from 5% to 90% of the number of faults in a single-version system [1-3]. It is therefore necessary to solve the multi-version development system problem of the optimal selection of version redundancy, which then provides the required diversity (safety) and minimal costs for the system.

2. RELATED WORKS

The diversity approach is one of the general principles used to decrease vulnerability against CCF and provide dependability of nuclear power plant instrumentation and control (I&C) systems, aerospace on-board systems, railway interlocking and block signal systems, etc. [3, 4]. Diversity is used jointly with structure and temporal redundancy to decrease the risks of the CCF. The IEC 60880 standard defines the diversity as “a means of enhancing the reliability of some systems and reducing the potential for certain CCF” [5].

Version is an option for the different realization of identical tasks (product or process); examples of versions are software, hardware or field-programmable gate array (FPGA) components performing the functions of I&Cs. Version redundancy is when different versions are used. There are many version

redundancy types and classification schemes [4-7]. The problems of CCF should be analyzed by taking into account the trends in computer technologies development, including the growing application of FPGAs [3, 7].

A key problem is the assessment of diversity metrics. Solve this problem may require information about diversity types and an expert assessment of their preferences [1-4]. The most probable sources of CCF are design faults or the multiple physical faults of different channels. The probability of CCF of safety critical systems may be essentially decreased by the application of different types of diversity if failures of redundant channels (versions) are maximally independent [3].

The problems of software and FPGA-based multi-version system development are described and analyzed in [4, 6, 7]. The complexity of diversity type choices occurs for two reasons. First, the number of diverse version pairs is very large. This number may be determined as the multiplication of cardinalities of sets for every attribute. Second, dependencies exist between different types of diversity (e.g., between different manufacturers of chips and technologies of chips, between technologies and families of chips, etc.) [8].

Therefore, these dependencies essentially complicate the task of diversity type selection and lead to the necessity of developing a model that allows for the systematization of the generation and choice of diversity type pairs.

3. GOAL

The earlier proposed graphical model [8] takes dependencies among diversity types into consideration and simplifies the choice of diversity options. However, as this model does not consider the metrics of diversity or costs, we attempt to extend and improve this model.

We propose a selection model based on the required metrics of diversity and the costs for safety-critical I&C systems. This model takes dependencies among diversity types, metrics, and the costs into consideration and allows for optimizing the choice of diversity options during system development.

The problem of making optimal decisions with various diversity types is the task of finding a pair of compatible elements for each type with required degree of diversity. We will use two criteria to obtain an optimal design decision of the system:

1. Find an optimal design decision with various types of diversity that provides minimum cost with the required degree of diversity.
2. Find an optimal design decision with various types of diversity that provides maximum diversity with assumed cost.

The suggested model simplifies the choice of an optimal design decision for a multi-version system.

4. DIVERSITY TYPE SELECTION MODEL

Let a set of diversity (or version redundancy) types

$$MD = \{d_1, d_2, \ldots, d_D\} \tag{1}$$

be used to develop a multi-version system. It may be a diversity of chip technologies (d_1), manufacturers of chips (d_2), families of chips (d_3), and others [4].

Let a set of diverse elements

$$MDE_j = \{e_{j1}, e_{j2}, \ldots, e_{jmj}\} \tag{2}$$

correspond to diversity type d_j. For example, the set MDE_1 consists of technologies FPGA (e_{11}), program logic controllers (e_{12}), microprocessors (e_{13}) and their subtechnologies.
A set

$$MDD = MDE_1 X MDE_2 X \ldots X MDE_D \tag{3}$$

forms all variants of project decisions (versions) for a multi-version system (X is the operation of Cartesian product).

In general, we have

$$m = m_1 \cdot m_2 \ldots m_D \tag{4}$$

versions to develop a multi-version system. But taking into account existing of dependencies between diversity types, the number of versions will be less [8].

Version L_t is described as a vector of elements

$$L_t = (e_{(t)1}, e_{(t)2}, \ldots, e_{(t)D}) \tag{5}$$

where $e_{(t)j} \in MDE_j$. This vector corresponds to one path in a direct acyclic graph G consisting of m + 2 nodes including the nodes "Enter" and "Exit." This graph may be represented in a compressed form GC where all elements $e_{j1}, e_{j2}, \ldots, e_{jmj}$ of j-th diversity level are joined in one node. Graph GC is step by step transformed into a special form of the graph GS in the case of dependencies between diversity types [8]. The graph GS allows for searching sequentially all paths (and vectors L) by taking into consideration such dependencies.

If a two-version system is developed according to project requirements, a pair $PL(L_t, L_k)$ of vectors L_t and L_k, $t \neq k$, should be selected. In general vectors L_t and L_k may differ in one, two, or D elements.

A set of pairs

$$MPL = \{PL(L_1, L_2), PL(L_1, L_3), \ldots, PL(L_{m-1}, L_m)\} \tag{6}$$

contains $r = C_m^2$ elements.

The version L_t and pair $PL(L_t, L_k)$ can be described by one and two ways correspondingly in a special graph that is called a graph of multi-version technologies [7].

Each pair PL_i is characterized by a metric of diversity DPL_i and cost CPL_i. The values of DPL_i and CPL_i depend on pairs of elements for selected vectors L_t, and L_k:

$$PL(L_t, L_k) = (e_{(t)1}, e_{(k)1}), (e_{(t)2}, e_{(k)2}), \ldots, (e_{(t)D}, e_{(k)D}) \tag{7}$$

The diversity metric for the pair PL_i is calculated in the following way

$$DPL_i = \sum_{j=1}^{D} \omega_{i,j} \cdot DPL_{i,j} \tag{8}$$

where $\omega_{i,j}$ is the weighting coefficient, $0 \leq \omega_{i,j} \leq 1$, and the sum of weighting coefficients equals 1.

The model assumes D diversity levels, and the sum of the weighting coefficients for them should be 1. To add a new level of diversity and in order to preserve the normalization condition, we need to recalculate the value of weighting coefficients and the new sum also should be 1. If a part of the specified diversity types is not used, i.e., the pair of elements $(e_{(t)h}, e_{(k)h})$ consists of identical elements, then the corresponding weighting coefficient $\omega_{t,k}$ will equal zero and the sum of ones will be less than 1.

It is not always possible to evaluate the project by simply summing the cost value of each element. In reality, the model should consider some variants of the elements' assessment. Each element on j-th level of diversity can include elements by default from other levels and we need to consider this property for the cost. For example, technologies of chips (SRAM, Flash, Antifuse for FPGAs, etc.) or manufacturers of chips (companies Altera, Xilinx, Microsemi, etc.) can include families of chips (Cyclone, Aria, Stratix, Virtex) and there is no need to consider the cost. In this case, we should set the cost to zero for each element of the diversity level families of chips.

Let j_i – factor characterize the need to consider the cost of elements at the j-th level, where $j_i = \{0,1\}$. The value is zero $j_i = 0$ if the cost already included to the element at other level of diversity and the value is one if we need to consider the cost:

$$CPL_i = \sum_{j=1}^{D} j_i \cdot CPL_{i,j} \tag{9}$$

where $CPL_{i,j}$ – a cost of pair L_i and L_j.

To design one subsystem (version) of a multi-version system, it is necessary to choose a specific value from each set. If there are no dependencies among diversity types, then any combination of values is possible. An optimal design decision should contain a pair of elements according to the model (7), and we need to find one pair of decision according to selected criteria (task 1 or task 2):

Task 1. Find an optimal design decision with various types of diversity that provides minimum cost $CPL \to \min$ with the required degree of diversity DPL_{req}:

$$f = \begin{cases} DPL \geq DPL_{req} \\ CPL \to \min \end{cases} \tag{10}$$

The solution to this problem includes the following sequence of steps:

Step 1. Determine a set of versions. This task may be solved in the case of existing diversity type dependencies by the development of a direct acyclic graph GS according to the model [8].

Step 2. Determine diversity value $DPL_{i,j}$ and a cost $CPL_{i,j}$ for each pair of design decisions.

Step 3. Determine the weighting coefficient for each $j-th$ level of diversity.

Step 4. Calculate diversity value and a cost for each pair of decisions $PL_i = \langle DPL_i, CPL_i \rangle$.

Step 5. Determine the pairs of decisions PL_i that provide required degree of diversity DPL_{req}.

Step 6. Determine one pair of decisions PL_i that provides minimum cost $CPL \to \min$.

Task 2. Find an optimal design decision with various types of diversity that provides maximum diversity $DPL \to \max$ with assumed cost CPL_{assum}:

$$f = \begin{cases} CPL \leq CPL_{assum} \\ DPL \to \max \end{cases} \tag{11}$$

The solution of this problem includes following sequence of steps:

Step 1-4. Repeat steps 1-4 (task 1).

Step 5. Determine the pairs of decisions PL_i that provide assumed cost CPL_{assum}.

Step 6. Determine one pair of decisions PL_i that provides $DPL \to \max$.

5. EXAMPLE

According to the proposed model, it is necessary to choose two decisions for the optimal configuration of a two-version system based on (10) or (11) criteria. A basic example was taken from the article [8]. Based on diversity types presented at Fig. 1, the example of the diversity model is developed using

abstract sets of diversity values. This makes the example more general and applicable for various types of computer systems. There are diverse technologies of chips (TC) (SRAM, Flash and Antifuse for FPGAs; program logic controller-, microprocessor- and microcontroller-based technologies); manufacturers of chips (MC) (Altera, Xilinx, Microsemi, Intel, Motorola, etc.); families of chips (FC) (e.g., Cyclone, Aria, Stratix, Virtex, etc.); technologies of printed circuit board production (TP) based on different materials, dielectrics, technological processes, etc.; manufacturers of printed circuit boards (MP) (companies in different countries); languages (L) (VHDL, JHDl, C, C++, etc.); and technologies of development and verification (TO).

We consider these seven diversity types and dependencies among the values (Table 1), which are typical for many safety-critical systems. For example, the application of chips from Altera (MC) stipulates use of SRAM-FPGA technology-producing languages (L) and technologies and case tools of development and verification (TO). Dependencies between diversity types are shown in the Table 1 by arrows for corresponding types and subtypes.

Table 1: Diversity types, elements, and dependencies among diverse elements [8]

	Diversity type	Diverse elements
1	TC	TC1, TC2, TC3, TC4, TC5, TC6
2	MC	MC1, MC2, MC3, MC4, MC
3	FC	FC1, FC2, FC3, FC4, FC5, FC6
4	TP	TP1, TP2, TP3, TP4, TP5
5	MP	MP1, MP2, MP3, MP4
6	L	L1, L2, L3, L4, L5
7	TO	TO1, TO2, TO3

Dependencies among diverse elements	
TC → MC	TC1, TC2, TC3 → MC1, MC2, MC3
	TC4, TC5, TC6 → MC4, MC5
MC → FC	MC1, MC2 → FC1, FC2
	MC3, MC4, MC5 → FC3, FC4, FC5, FC6
FC → TP	FC1, FC2, FC4 → TP1, TP2
	FC3, FC5, FC6 → TP3, TP4, TP
TP → MP	TP1, TP3, TP5 → MP1, MP2
	TP2, TP4 → MP3, MP4
TC → L	TC1, TC3 → L1, L2, L3
	TC2, TC4, TC5, TC6 → L4, L5
L → TO	L1 → TO1
	L2, L3, L5 → TO2
	L4 → TO3
TC → TO	TC1, TC3, TC5, TC6 → TO1, TO2
	TC2, TC4 → TO3

According to the model [8], input data for the model were prepared and the following operations were performed:

1. The first step of the algorithm starts from the graph GC (Fig. 1), which describes possible diversity types, but it does not reflect any dependencies between these elements according to Table 1.
2. The next step includes splitting a subgraph, labeling ingoing and outgoing edges of split subgraphs, eliminating dead nodes and edges, and merging nodes. We repeated this for each dependency from Table 1. The final model of the complete example according to [8] is presented in Fig. 2. The model contains 26 different paths with 374 feasible diversity combinations, as shown in Table 2.

Figure 1: An example of the graph GC with weight coefficients

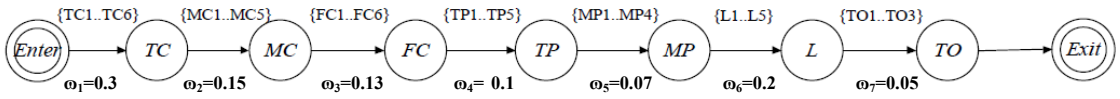


Figure 2: The graph GS (model of dependencies 1 – 7) [8]

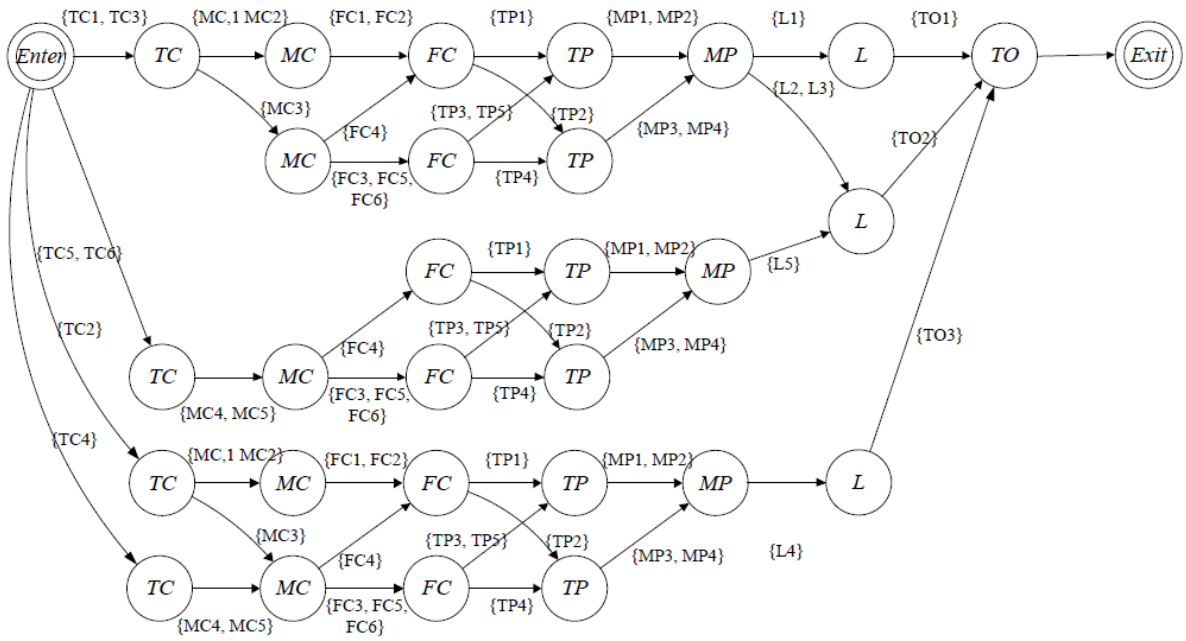


Table 2: Feasible combinations of diversity types

Path	TC	MC	FC	TP	MP	L	TO	Number of feasible comb.
1	TC1, TC3	MC1, MC2	FC1, FC2	TP1	MP1, MP2	L1	TO1	16
2	TC1, TC3	MC1, MC2	FC1, FC2	TP1	MP1, MP2	L2, L3	TO2	32
3	TC1, TC3	MC1, MC2	FC1, FC2	TP2	MP1, MP2	L1	TO1	16
4	TC1, TC3	MC1, MC2	FC1, FC2	TP2	MP3, MP4	L2, L3	TO2	32
5	TC1, TC3	MC3	FC4	TP1	MP1, MP2	L1	TO1	4
6	TC1, TC3	MC3	FC4	TP1	MP1, MP2	L2, L3	TO2	8
7	TC1, TC3	MC3	FC4	TP2	MP1, MP2	L1	TO1	4
8	TC1, TC3	MC3	FC4	TP2	MP3, MP4	L2, L3	TO2	8
9	TC1, TC3	MC3	FC3, FC5, FC6	TP3, TP5	MP1, MP2	L1	TO1	24
10	TC1, TC3	MC3	FC3, FC5, FC6	TP3, TP5	MP1, MP2	L2, L3	TO2	48
11	TC1, TC3	MC3	FC3, FC5, FC6	TP4	MP3, MP4	L1	TO1	12
12	TC1, TC3	MC3	FC3, FC5, FC6	TP4	MP3, MP4	L2, L3	TO2	24
13	TC5, TC6	MC4, MC5	FC4	TP1	MP1, MP2	L5	TO2	8
14	TC5, TC6	MC4, MC5	FC4	TP2	MP3, MP4	L5	TO2	8
15	**TC5**, TC6	MC4, **MC5**	FC3, **FC5**, FC6	TP3, TP5	**MP1**, MP2	**L5**	**TO2**	24
16	TC5, TC6	MC4, MC5	FC3, FC5, FC6	TP4	MP3, MP4	L5	TO2	24
17	TC2	MC1, MC2	FC1, FC2	TP1	MP1, MP2	L4	TO3	8
18	TC2	MC1, MC2	FC1, FC2	TP2	MP3, MP4	L4	TO3	8
19	TC2	MC3	FC4	TP1	MP1, MP2	L4	TO3	2
20	TC2	MC3	FC4	TP2	MP3, MP4	L4	TO3	2
21	TC2	MC3	FC3, FC5, FC6	TP3, TP5	MP1, MP2	L4	TO3	12
22	TC2	MC3	FC3, FC5, FC6	TP4	MP3, MP4	L4	TO3	6
23	**TC4**	**MC4**, MC5	**FC4**	**TP1**	MP1, **MP2**	**L4**	**TO3**	4
24	TC4	MC4, MC5	FC4	TP2	MP3, MP4	L4	TO3	4
25	TC4	MC4, MC5	FC3, FC5, FC6	TP3, TP5	MP1, MP2	L4	TO3	24
26	TC4	MC4, MC5	FC3, FC5, FC6	TP4	MP3, MP4	L4	TO3	12
Total								374

According to the proposed model, at step 2 we should assign diversity and cost value for each pair of elements on all levels of diversity (TC, MC, FC, etc.). Each level of diversity must be characterized by weighting coefficients (Fig. 1) Diversity values and weighting coefficients are normalized between 0 and 1. Weighting coefficients determine the importance of diversity type and their sum is 1. The cost value is given in absolute units.

We have all feasible combinations of diversity types (Table 2) that can be presented as a set of decisions. We then need to make simple calculations for each pair of decisions to get an optimal design decision according to our model for a two-version system.

As an example, Tables 3 and 4 show the input data and calculations for two pairs of decisions (L_1, L_2) and (L_1, L_3), where L_1 is (TC1, MC2, FC2, TP1, MP1, L1, TO1) from path 1 (Table 2), L_2 is (TC3, MC3, FC3, TP4, MP3, L2, TO2) from path 12, and L_3 is (TC3, MC3, FC4, TP2, MP4, L3, TO2) from path 8.

Table 3: Diversity value for (L_1, L_2) pair of versions

(L_1, L_2)	**TC1**	**TC3**	**MC2**	**MC3**	**FC2**	**FC3**	**TP1**	**TP4**	**MP1**	**MP3**	**L1**	**L2**	**TO1**	**TO2**
Cost value	44	80	68	54	62	66	70	13	30	79	23	68	12	13
$j_i = \{0,1\}$	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Diversity value	0.11		0.5		0.15		0.73		0.13		0.86		0.8	
DPL_1	$DPL_1 = 0.11 \cdot 0.3 + 0.5 \cdot 0.15 + 0.15 \cdot 0.13 + 0.73 \cdot 0.1 + 0.13 \cdot 0.07 + 0.86 \cdot 0.2 + 0.8 \cdot 0.05 = 0.49$													
CPL_1	$CPL_1 = 44 \cdot 1 + 80 \cdot 1 + 68 \cdot 1 + 54 \cdot 1 + 62 \cdot 0 + 66 \cdot 1 + 70 \cdot 1 + 13 \cdot 1 + 30 \cdot 1 + 79 \cdot 1 + 23 \cdot 1 + 68 \cdot 1 + 12 \cdot 1 + 13 \cdot 1 = 620$													

Table 4: Diversity value for (L_1, L_3) pair of versions

(L_1, L_3)	**TC1**	**TC3**	**MC2**	**MC3**	**FC2**	**FC4**	**TP1**	**TP2**	**MP1**	**MP4**	**L1**	**L3**	**TO1**	**TO2**
Cost value	44	80	68	54	62	66	70	55	30	76	23	59	12	13
$j_i = \{0,1\}$	1	1	1	1	0	1	1	1	1	1	1	1	1	1
Diversity value	0.11		0.5		0.15		0.61		0.7		0.49		0.8	
DPL_2	$DPL_2 = 0.11 \cdot 0.3 + 0.5 \cdot 0.15 + 0.15 \cdot 0.13 + 0.61 \cdot 0.1 + 0.7 \cdot 0.07 + 0.49 \cdot 0.2 + 0.8 \cdot 0.05 = 0.73$													
CPL_2	$CPL_2 = 44 \cdot 1 + 80 \cdot 1 + 68 \cdot 1 + 54 \cdot 1 + 62 \cdot 0 + 66 \cdot 1 + 70 \cdot 1 + 55 \cdot 1 + 30 \cdot 1 + 76 \cdot 1 + 23 \cdot 1 + 59 \cdot 1 + 12 \cdot 1 + 13 \cdot 1 = 650$													

To make these calculations, we developed the Diversity Analyzer tool. To get the optimal design, we need to do following steps:

- The initial data of the model should be defined by an expert (Fig. 3).
- Create a full group of elements for each level of diversity (Table 1).
- Set the level of diversity for each pair of elements.
- Set the cost for each element and price relationship between elements (Fig. 3).
- Set a weighting coefficient for each level of diversity (Fig. 3).
- After all initial data are entered, the user can start working with the system (Fig. 4).

The result can be presented in text form (Fig. 4) and in the form of a graph. This software takes into consideration the dependencies among diversity types, diversity metrics, and the cost and presents a decision in the form of two decisions for a two-version system.

According to the model, it is necessary to choose two pair of decisions for the optimal configuration of a two-version system based on (10) or (11) criteria. According to (10) criterion, we need to assign value for required degree of diversity DPL_{req}, and according to (11) criterion, we need to assign a value for cost CPL_{assum}. The optimal design for our example according to (10) is (TC5, MC5, FC5, TP5, MP1, L5, TO2) from path 15 Table 2 and (TC4, MC4, FC4, TP1, MP2, L4, TO3) from path 23 Table 2. These optimal versions are marked bold in Table 2 and presented in the top of Fig. 4.

Figure 3: Diversity Analyzer tool. Values of diversity metrics and cost

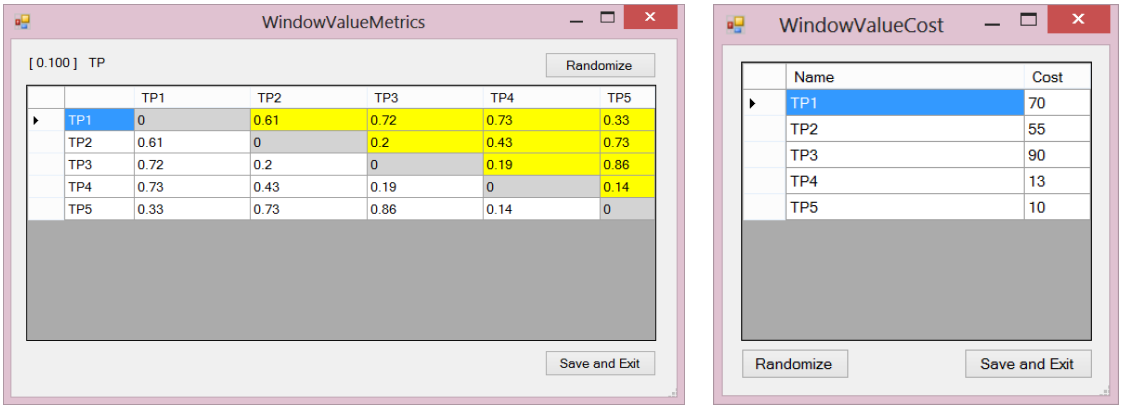
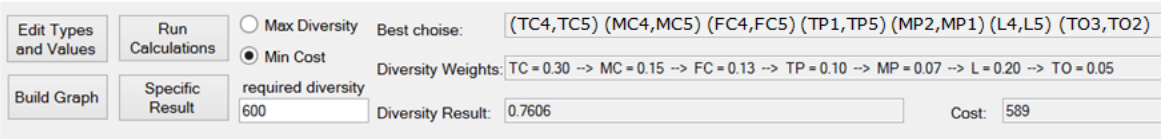


Figure 4: Diversity Analyzer tool. Criteria and results



6. CONCLUSIONS

The application of diversity decreases the probability of CCF. The complexity of diversity type choices is caused by a very large number of version pairs and dependencies between different diversity types (e.g., between different manufacturers of chips and technologies of chips, between technologies and families of chips, etc.). The suggested model can be used during safety-critical systems development or modernization for making an optimal design decision based on the criterion of safety-cost.

The proposed model takes into consideration the dependencies among diversity types, diversity metrics, and costs and presents a decision for each version of a two-version system. Future steps may be related to the development of general procedure for n-version systems (n more than 2) while taking into account reliability and other indicators and limitations.

References

[1] J. Knight. "*Diversity*," in Dependable and Historic Computing / C. Jones, J. Lloyd (editors), Lecture Notes in Computer Science, Volume 6875, Springer, 2011, pp. 298-312.

[2] K. Salako, L. Strigini. "*When does "diversity" in development reduce common failures? Insights from probabilistic modelling*," IEEE Transactions on Dependable and Secure Computing, 99, 2014, doi: 10.1109/TDSC.2013.32

[3] V. Kharchenko, A. Siora, V. Sklyar, A. Volkoviy. "*Multi-Diversity Versus Common Cause Failures: FPGA-Based Multi-Version NPP I&C Systems*," Proceedings of the 7th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology (NPIC-HMIT), Las-Vegas, Nevada, USA, 2010, pp. 85-96.

[4] NUREG/CR-7007, "*Diversity Strategies for Nuclear Power Plant Instrumentation and Control Systems*," ONL, Oak Ridge, USA, 2009.

[5] Standard IEC 60880 Ed. 2.0 b: 2006, "*Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions*."

[6] L. Pullum, "*Software Fault Tolerance Techniques and Implementation*," Artech House Computing Library, 2001.

[7] V. Kharchenko, A. Siora, E. Bakhmach. "*Diversity-scalable decisions for FPGA-based safety-critical I&Cs: from Theory to Implementation*," Proceedings of the 6th American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Controls, and Human Machine Interface Technology (NPIC-HMIT), Knoxville, TN, USA, 5-9 April 2009, pp. 78-84.

[8] S. Vilkomir, V. Kharchenko. "*A Diversity Model for Multi-Version Safety-Critical I&C Systems*," Proceedings of the 11th International Probabilistic Safety Assessment and Management Conference and the Annual European Safety and Reliability Conference (PSAM-ESREL), Helsinki, Finland, 25–29 June 2012, pp. 1791-1799.