## Simulation

*Simulation* is performed for the purpose of tune parameters in the two versions of test generation algorithm of *MART* (with *static* and *dynamic* *d*), then evaluating their performance compared to the *Simple Random Sampling Without Replacement*. To do this it is necessary to introduce *simulation scenarios* and *evaluation criteria*.

### 3.1 Simulation Scenarios

To obtain the simulation scenarios it is necessary to consider three main factors about the population and the problem size:

* *Type of partitioning*: that represents how Test Cases are partitioned in Test Frame. This information is codified as the Test Frame’s failure probability, necessary to determine the weight associated with a network’s link. This information can be encoded in a *Correct/Failing* value, for which three settings are considered: 0/1, 0.25/0.75 and 0.1/0.9. Test Frame associated to a failing value are called *Failing Test Frames*, this term indicates the tendency to failure. The case 0/1 is ideal, in fact this means that in each Test Frame there are only *not failing points* or only f*ailing points*, this refers to a *Perfect Partitioning*, if the failure Test Frames are organized in clusters, the population distribution is *Perfect Clustered*; the couple 0.25/0.75 and 0.1/0.9 means that the partitioning is not very good, because failure are distributed in all Test Frames, with these failure probabilities there is the encoding of distance from ideal Perfect Partitioning, these probabilities generate the *Close to Uniform* (for 0.25/0.75) and *Close to Perfect Partitioning* (0.1/0.9), if failure Test Frames are organized in cluster, the population distribution is *Clustered*.
* *Failure distribution*: this value indicates the proportion of *Failing Test Frames* on total. Two values are simulated, 0.1 and 0.2. Greater values are not of interest, because is a realistic assumption assume that maximum the 20% of Test Frame partitions are Failing.
* *Total number of Test Frames (N):* two order of magnitudes are tried: N = 100, N = 1000. Assessment is made at 9 checkpoints: n1 = 0.1N, n2 = 0.2N, ... 0.9N.

The combination of first two parameters generates 12 different populations, shown in Table 4, beside these a *completely uniform population distribution* is added: the failure probability of each Test Frame is obtained as a random value between 0 and 1, this case refers to a *Random Population*, in which failures are uniformly distributed between Test Frames, this represent an ideal case.

Table 1: Populations

|  |  |  |
| --- | --- | --- |
|  | Population distribution | Failure distribution |
| 1 | Uniform (Random) | 0.5 |
| 2 | Close to uniform (0.25/0.75) | 0.1 |
| 3 | Close to perfect partitioning (0.1/0.9) | 0.1 |
| 4 | Clustered (0.25/0.75) | 0.1 |
| 5 | Clustered (0.1/0.9) | 0.1 |
| 6 | Perfect | 0.1 |
| 7 | Perfect Clustered (1/0) | 0.1 |
| 8 | Close to uniform (0.25/0.75) | 0.2 |
| 9 | Close to perfect partitioning (0.1/0.9) | 0.2 |
| 10 | Clustered (0.25/0.75) | 0.2 |
| 11 | Clustered (0.1/0.9) | 0.2 |
| 12 | Perfect | 0.2 |
| 13 | Perfect Clustered | 0.2 |

26 different scenarios are obtained adding to the combinations the last factor.

Described scenarios are generated with uniform operational profile and random failure probability, according to all features, before the execution of respective simulation’s blocks.

The populations between 1 and 7 are defined in ascended order of compatibility with testing strategy (from worst to best case), while between 8 and 13 are the same from 2 and 7, but with different value of failure probability.

#### 3.1.1 Population generators

Different population distributions are generated using following functions:

* *generatePopulationAndMatrix*: to generate a random set of test Frames with random Failure Probability, random Distance Factor (between 0 and *maxdistance*); occurrence probabilities can be random or equiprobables.
* *generatePopulationAndMatrixBinary*: to generate a random set of test Frames with Failure Probability choose random between two selected value ((0/1), (0.1/0.9), (0.25/0.75)) respecting a Failure proportion (proportion of values with the high probability of failure on the total), random Distance Factor (between 0 and *maxdistance*) and the probability of Occurrence can be random or equiprobables;
* *generatePopulationAndMatrixCluster*: to generate a set of test Frames with a policy defined below, random Distance Factor (between 0 and *maxdistance*) and the probability of Occurrence can be random or equiprobables.

The steps used to determine the Failure Probability of *N* Test Frames with *clustered distribution* are the following:

A *t%* of failing Test Frames is considered in this set, the *t%-th* part of *N* is called *X*:

1. At first at all points is assigned the lowest failure probability;
2. Each Cluster of failure points is made by a certain percentage (*clusterDimPerc* in the code, in our case 10% and 20%) of *X*’s cardinality (call it *T*);
3. points are chosen randomly as centroids;
4. Finally, *T* points are chosen as the ones at minimum distance to the centroid and assign them the maximum failure probability.

### 3.2 Evaluation Criteria

*Accuracy* and *efficiency* are considered as evaluation criteria estimated as follows. A simulation scenario j is repeated 100 times; denote with r one of such repetitions. At the end of each repetition, the reliability estimates is computed by the technique under assessment as well as the *true* reliability . For simulation, it is known in advance which input t is a failure point (hence, , where is 1 if the input is a failure point and 0 otherwise).

For each scenario j, the sample mean (denoted as *M*), sample variance (*var*) and mean squared error (*MSE*) are computed:

* ;
* ;
* ;

Comparison of estimation accuracy is done by looking at the *MSE*. Comparison of efficiency is done by the *sample variance*.

### 3.3 Empirical correction of Estimator

During simulation a problem is emerged: for little values of *n*, *z*i at each pass is strongly influenced by the order and by the eventuality to take or not the major part of failures. This latter is a consequence of the choices made in formulation, in fact there is an important approximation in case of “failure probability” between [0,1]: Unreliability is calculated as where is a binary value, because only a Test Case from each Test Frame is taken. A consequence is the possibility that a Test Case taken from a Test Frame with “high” failure probability (ex.: 0.9) may not fail and a Test Case taken from a Test Frame with “low” failure probability (ex.: 0.1) may fail, causing underestimation or overestimation respectively. This phenomenon is more evident when are considered little values of n.

For the discussed problem it is more dangerous have a Reliability overestimation, thus the idea is to adjust this value for avoid this condition.

The estimation used so far is based on the mean value of the estimations calculated with *zi*. This quantity is a representation of estimation, influenced by the presence of "outliers”, that is very heavy when is considered a little value of n, but the mean is still the most representative value of the set.

By simulation it is possible observe that testing strategy takes Test Frames with the highest failure probability in the initial part of the sampling, the estimation is very prone to outliers, with consequential overestimation of Reliability.

At this point the idea is calculate the mean of estimated value in a single algorithm execution and use it to “filter” values that respect the following rules:

* values that differing not later than 90% by mean in overestimation;
* values that differing not later than 10% by mean in underestimation.

In other words, a *window with two limit values* is considered: the upper bound, as mean plus 90% of mean, and lower bound, as mean minus 10% of mean.

This idea doesn’t touch the unbiasedness of the estimators, in fact the concept is to choose the mean only of values that better representing the examined population, cutting liars value that contaminate estimation.

This adjustment is not functional with big values of *n*, in fact there is a constant increment of MSE (respect the estimator described in the article [2]).

The solution is based on a combination of two explained estimators, in base of *n* value. To do this is fundamental define what means little or big values of *n*.

After the execution of the first part of simulations is been observed that the estimator corrected have good performance when n is included between 0 and 30/40%. For this the idea is to divide all possible values of n in three parts as follow:

  
In the first part the estimator defined in this paragraph is used; in the third part the estimator defined in the article [2] (implemented as in Static Formulation) is used; in the second part a linear combination of both estimators is used. This linear combination is realized using a “*coefficient of correction*” defined as: and the estimation is calculated as:

Figure 13: Different Estimator usage

.

In the Table 5 it is possible observe that curves’ trend is convergent toward the true value.

Table 2: Results of the correct estimator based on MSE

|  |
| --- |
|  |
| Configuration 1 |
|  |
| Configuration 2 |
|  |
| Configuration 10 |

### 3.4 Sensitivity Analysis

#### 3.4.1 Sensitivity Analysis of *d* in Static Approach

The choice of *d* is taken in base of comparing results obtained considering four values: 0.2, 0.4, 0.6, 0.8. In this four value is encoded the trust in the sampling weight-based, respect to the random.

The Sensitivity Analysis has been made on 5 choices populations: 1, 2, 8, 9, that represent the limit cases, and 12, as best case example.

The comparison is made on two representative values: *MSE* and *Sample* *Variance*.

Table 3: MSE comparison between d = 0.2, d = 0.4, d = 0.6 and d = 0.8, with reference to the most significant configurations

|  |
| --- |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf1/Schermata%202018-04-19%20alle%2011.57.19.png |
| Configuration 1 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf2/Schermata%202018-04-19%20alle%2012.01.17.png |
| Configuration 2 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf8/Schermata%202018-04-19%20alle%2012.03.15.png |
| Configuration 8 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf9/Schermata%202018-04-19%20alle%2012.05.14.png |
| Configuration 9 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf12/Schermata%202018-04-19%20alle%2012.07.51.png |
| Configuration 12 |

From results shows in Table 6, the best value of *d* about *MSE* is certain 0.8, in particular for little values of *n*.

Table 4: Sample Variance comparison between d = 0.2, d = 0.4, d = 0.6 and d = 0.8, with reference to the most significant configurations

|  |
| --- |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf1/Schermata%202018-04-19%20alle%2011.57.28.png |
| Configuration 1 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf2/Schermata%202018-04-19%20alle%2012.01.26.png |
| Configuration 2 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf8/Schermata%202018-04-19%20alle%2012.03.22.png |
| Configuration 8 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf9/Schermata%202018-04-19%20alle%2012.05.23.png |
| Configuration 9 |
| Simulazione/Simulazioni%20utilizzate/Analisi%20di%20sensitività%20su%20d%20statico/conf12/Schermata%202018-04-19%20alle%2012.07.59.png |
| Configuration 12 |

As shown in Table 7, the *Sample* *Variance* is few influenced by the *d* value in the different configurations, in fact except for few cases the variance values fall in the same order of magnitude. In the configurations 1 and 2 there are better values for values 0.2 and 0.4 of *d*, but in configuration 9 and 12 there are better values for 0.6 and 0.8.

With this consideration the selected value of *d* for the simulation evaluation is 0.8.

#### 3.4.2 Sensitivity Analysis in Dynamic variant

Considering the Dynamic approach, there are two different values for which consider a sensitivity analysis: *d0* (value between 0.5 and 0.9 with step 0.1) and the *shift register size* (3, 4 and 5).

Sensitivity Analysis on *d0*

How previously mentioned to tune the used value of *d0* in simulation five different values between 0.5 and 0.9 with step 0.1 are considered. *MSE*, *Sample* *Variance* and Number of Failing Point (*NFP*) are the used parameters to make this sensitivity analysis.

By observing all simulations, it is possible to observe 3 different trends:

* *MSE* has a growing trend with the increment of *d0*;
* *Sample Variance* has a downward trend with the increment of *d0*;
* *NFP* is almost constant with the increment of *d0*.

In this case up to 13 different populations could be considered, but to evaluate the difference only two subsets are considered: a subset made by Configurations 1, 2, 8 and 9 (the limit cases, with the Configuration 9 as the worst of best cases); a subset made by all other configurations (best cases), in which the performances are more or less the same. In general, is possible observe that for values of *n* bigger than 30% of population size the performances of the algorithm in different populations is the same. When little values of *n* are considered, in particular values smaller than 30%, the performances are better according to the previous description of trends.

More interesting is the consideration about the limit cases, the first subset, in fact there are substantial differences for different values of d0.

Table 5: Sensitivity Analysis on d0 respect to MSE, where Ad1 is the implementation with dynamic d and initial sample unitary, while Ad2 is the implementation with dynamic d and variable initial sample

|  |
| --- |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%201/Schermata%202018-02-02%20alle%2012.07.21.png |
| Configuration 1 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%202/Schermata%202018-02-02%20alle%2012.10.23.png |
| Configuration 2 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%208/Schermata%202018-02-03%20alle%2010.14.57.png |
| Configuration 8 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%209/Schermata%202018-02-03%20alle%2010.16.37.png |
| Configuration 9 |

How described previously the *MSE* have a growing trend with the increment of *d0*. As shown in Table 8 the best values of *d0* are 0.8 and 0.9.

Table 6: Sensitivity Analysis on d0 respect to Sample Variance, where Ad1 is the implementation with dynamic d and initial sample unitary, while Ad2 is the implementation with dynamic d and variable initial sample

|  |
| --- |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%201/Schermata%202018-02-02%20alle%2012.07.35.png |
| Configuration 1 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%202/Schermata%202018-02-02%20alle%2012.10.34.png |
| Configuration 2 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%208/Schermata%202018-02-03%20alle%2010.15.15.png |
| Configuration 8 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%209/Schermata%202018-02-03%20alle%2010.16.46.png |
| Configuration 9 |

In Table 9 it is shown that *Sample Variance* is the best for little values of d0, in particular 0.9 is evidently the worst case.

Table 7: Sensitivity Analysis on d0 respect to NFP, where Ad1 is the implementation with dynamic d and initial sample unitary, while Ad2 is the implementation with dynamic d and variable initial sample

|  |
| --- |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%201/Schermata%202018-02-02%20alle%2012.07.46.png |
| Configuration 1 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%202/Schermata%202018-02-02%20alle%2012.10.45.png |
| Configuration 2 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%208/Schermata%202018-02-03%20alle%2010.15.30.png |
| Configuration 8 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20diversi%20valori%20di%20d0/screen/Configurazione%209/Schermata%202018-02-03%20alle%2010.16.57.png |
| Configuration 9 |

In case of *NFP* performances, shown in Table 10, are more or less the same for each value of *d0*.

Considering all the observation the value selected of *d0* is 0.8, that offer a good tradeoff between *MSE* and *Sample Variance*.

Sensitivity Analysis on Shift Register

The analysis is made on four values of shift registers’ dimension: 3, 4 and 5.

The values associated to each cell are organized as follow:

|  |  |  |
| --- | --- | --- |
| 0.5 | 0.3 | 0.2 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0.4 | 0.3 | 0.2 | 0.1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0.4 | 0.3 | 0.15 | 0.1 | 0.05 |

Like in the previous case, simulation has been made for each configuration of examined population. About *MSE* and *Sample Variance*, the three configuration are less different.

In case of *MSE* the better value of Shift Register’s dimension is 4, because it gives better performances also in the limit cases.

Table 8: Sensitivity Analysis on Shift Register dimension respect to MSE

|  |
| --- |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%201/Schermata%202018-02-05%20alle%2010.18.28.png |
| Configuration 1 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%202/Schermata%202018-02-05%20alle%2010.20.33.png |
| Configuration 2 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%208/Schermata%202018-02-05%20alle%2010.21.39.png |
| Configuration 8 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%209/Schermata%202018-02-05%20alle%2010.23.06.png |
| Configuration 9 |

As shown in Table 11, in these cases performances are better in cases SR = 3 and 4, with a slight difference in favor of the value 4.

As in the case of *MSE*, *Sample Variance* is more or less the same in all cases, except that in the limit cases:

Table 9: Sensitivity Analysis on Shift Register dimension respect to MSE

|  |
| --- |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%201/Schermata%202018-02-05%20alle%2010.18.37.png |
| Configuration 1 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%202/Schermata%202018-02-05%20alle%2010.20.43.png |
| Configuration 2 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%208/Schermata%202018-02-05%20alle%2010.21.52.png |
| Configuration 8 |
| Simulazione/Analisi%20di%20sentività%20versione%20dinamica/Adaptive%20con%20varie%20dimensioni%20SR/screen/Configuration%209/Schermata%202018-02-05%20alle%2010.23.14.png |
| Configuration 9 |

Observing Table 12, the performances of SR=5 are better, while values 4 and 3 show the same performances. It is important note that in this case there is an uncertainty linked to the third decimal place.

The chosen value is SR=4, because is the best value in case of MSE, with acceptable values for the variance (2nd best value).

### 3.5 Results

Simulation results are based on six different configurations:

1. Static *d* with *n0=1* (1);
2. Static *d* with *n0≥1* (2);
3. Simple Random Sampling (SRS);
4. Dynamic *d* with *n0=1* (Ad1);
5. Dynamic *d* with *n0≥1* (Ad2).

#### 3.5.1 MSE

To evaluate the difference between the three different approaches is possible observe the following histograms. The evaluation of MSE is important to make consideration on how the obtained values deviate from the mean average. All results are shown in Table 13.

Table 10: MSE simulation results for each Configuration

|  |
| --- |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%201/Schermata%202018-01-31%20alle%2016.54.15.p |
| Configuration 1 (a) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%202/Schermata%202018-01-31%20alle%2017.00.31.p |
| Configuration 1 (b) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%203/Schermata%202018-01-31%20alle%2017.04.05.p |
| Configuration 3 (c) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%204/Schermata%202018-01-31%20alle%2017.06.32.p |
| Configuration 4 (d) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%205/Schermata%202018-01-31%20alle%2017.13.22.p |
| Configuration 5 (e) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%206/Schermata%202018-01-31%20alle%2017.15.50.p |
| Configuration 6 (f) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%207/Schermata%202018-01-31%20alle%2017.53.21.p |
| Configuration 7 (g) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%208/Schermata%202018-01-31%20alle%2017.56.08.p |
| Configuration 8 (h) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%209/Schermata%202018-01-31%20alle%2017.59.20.p |
| Configuration 9 (i) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2010/Schermata%202018-01-31%20alle%2018.01.41.p |
| Configuration 10 (j) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2011/Schermata%202018-01-31%20alle%2018.04.46.p |
| Configuration 11 (k) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2012/Schermata%202018-01-31%20alle%2018.06.41.p |
| Configuration 12 (l) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2013/Schermata%202018-01-31%20alle%2018.10.06.p |
| Configuration 13 (m) |

#### 3.5.2 Sample Variance

*Sample Variance* is important to evaluate the goodness of estimation; in particular it underline how much estimation is representative of population mean. All results are shown in Table 14.

Table 11: Sample Variance simulation results for each Configuration

|  |
| --- |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%201/Schermata%202018-01-31%20alle%2016.54.49.p |
| Configuration 1 (a) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%202/Schermata%202018-01-31%20alle%2017.01.17.p |
| Configuration 2 (b) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%203/Schermata%202018-01-31%20alle%2017.04.42.p |
| Configuration 3 (c) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%204/Schermata%202018-01-31%20alle%2017.11.22.p |
| Configuration 4 (d) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%205/Schermata%202018-01-31%20alle%2017.13.44.p |
| Configuration 5 (e) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%206/Schermata%202018-01-31%20alle%2017.16.20.p |
| Configuration 6 (f) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%207/Schermata%202018-01-31%20alle%2017.53.52.p |
| Configuration 7 (g) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%208/Schermata%202018-01-31%20alle%2017.56.32.p |
| Configuration 8 (h) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%209/Schermata%202018-01-31%20alle%2017.59.53.p |
| Configuration 9 (i) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2010/Schermata%202018-01-31%20alle%2018.02.06.p |
| Configuration 10 (j) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2011/Schermata%202018-01-31%20alle%2018.05.14.p |
| Configuration 11 (k) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2012/Schermata%202018-01-31%20alle%2018.07.02.p |
| Configuration 12 (l) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2013/Schermata%202018-01-31%20alle%2018.10.39.p |
| Configuration 13 (m) |

#### 3.3.3 Failing Point Number

This quantity explains the trend of different techniques to expose failures, all results are shown in Table 15.

Table 12: NFP simulation results for each Configuration

|  |
| --- |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%201/Schermata%202018-01-31%20alle%2016.54.38.p |
| Configuration 1 (a) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%202/Schermata%202018-01-31%20alle%2017.00.43.p |
| Configuration 2 (b) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%203/Schermata%202018-01-31%20alle%2017.04.17.p |
| Configuration 3 (c) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%204/Schermata%202018-01-31%20alle%2017.11.09.p |
| Configuration 4 (d) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%205/Schermata%202018-01-31%20alle%2017.13.32.p |
| Configuration 5 (e) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%206/Schermata%202018-01-31%20alle%2017.16.03.p |
| Configuration 6 (f) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%207/Schermata%202018-01-31%20alle%2017.53.35.p |
| Configuration 7 (g) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%208/Schermata%202018-01-31%20alle%2017.56.20.p |
| Configuration 8 (h) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%209/Schermata%202018-01-31%20alle%2017.59.42.p |
| Configuration 9 (i) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2010/Schermata%202018-01-31%20alle%2018.01.54.p |
| Configuration 10 (j) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2011/Schermata%202018-01-31%20alle%2018.04.58.p |
| Configuration 11 (k) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2012/Schermata%202018-01-31%20alle%2018.06.50.p |
| Configuration 12 (l) |
| ../../Desktop/UNINA/TESI/screen%20Simulazione/Configurazione%2013/Schermata%202018-01-31%20alle%2018.10.26.p |
| Configuration 13 (m) |

#### 3.3.4 Considerations

At first time it is considered how *SRS* works respect the four different test generation algorithm of *MART* versions, to do this limit configurations 1, 2 and 8, represented in graphs (a), (b) and (h), are considered. In this Configurations *MSE* is better in *SRS*, this result depends on the uniform distribution of failures (or close to uniform distribution in case 0.25/0.75), that is an ideal case away from the real world. On the other hand, *SRS* is worse than test generation algorithm of *MART* versions respect *Sample Variance*, where the difference is very evident.

For all other Configurations there is a better behavior of our techniques respect *SRS*, except for few isolated point regard *MSE*, but with no discussions regard *Sample Variance*.

At last the *NFP* values are considered, where the four implementations of our techniques are globally better.

Verified that out techniques are globally better than *SRS*, it is time to verify what is the better between the four described. At first time the different “initial sample dimensions” are compared: the two techniques with *unitary initial sample* are globally better, both for *MSE* than for Variance.

This consideration reduces the comparison between technique with *static d* and technique with *dynamic d*, knowing that the differences are very slightly. For *MSE*, values are more or less the same, with configuration like 5 in which the dynamic approach is better, while there are other configurations, like 6 in which the static approach is better. For Variance the static d is meanly better.

Finally, the test generation algorithm of *MART with static d* is considered better, not only for a better *Sample Variance*, but also for the simpler formulation.