IV&V International Workshop 2013

Evaluating the t-way Combinatorial Technique for Determining the Thoroughness of a Test Suite

Charley Price
Charles.Price@tasc.com

Rick Kuhn
d.kuhn@nist.gov

Ricky Forquer
Ricky.A.Forquer@nasa.gov

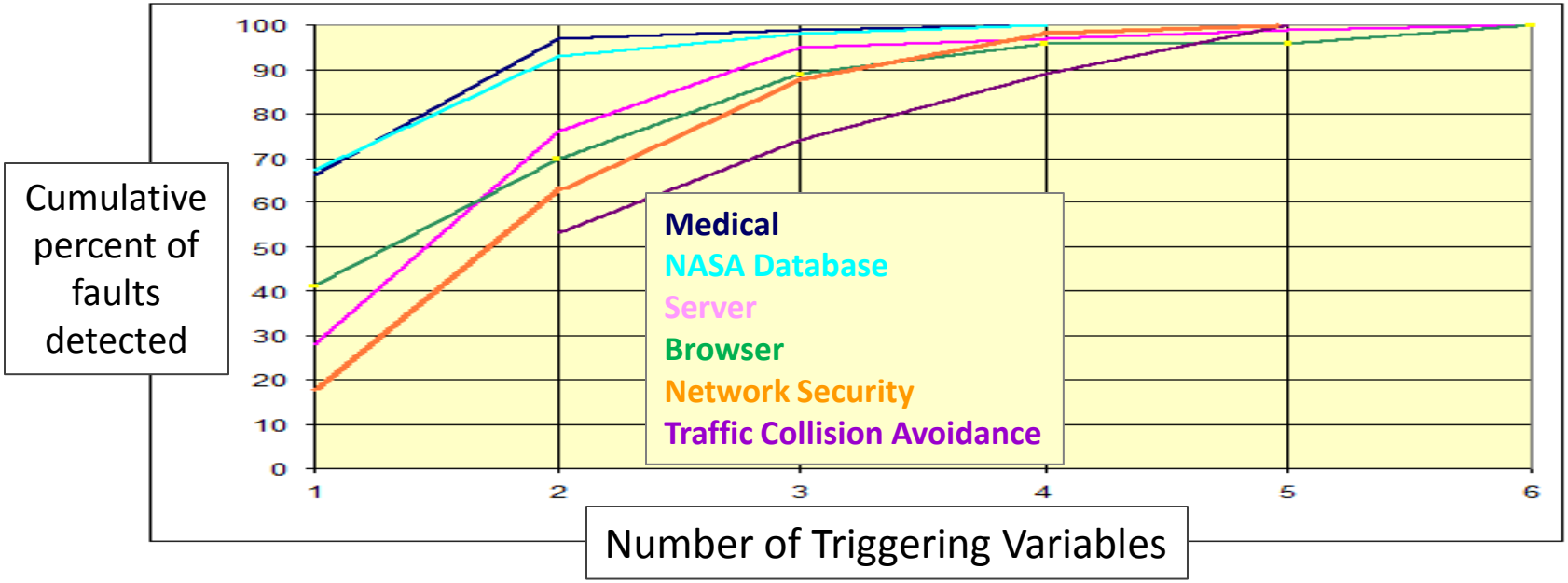
Adel Lagoy
Ricky.A.Forquer@nasa.gov

Raghu Kacker
d.kacker@nist.gov

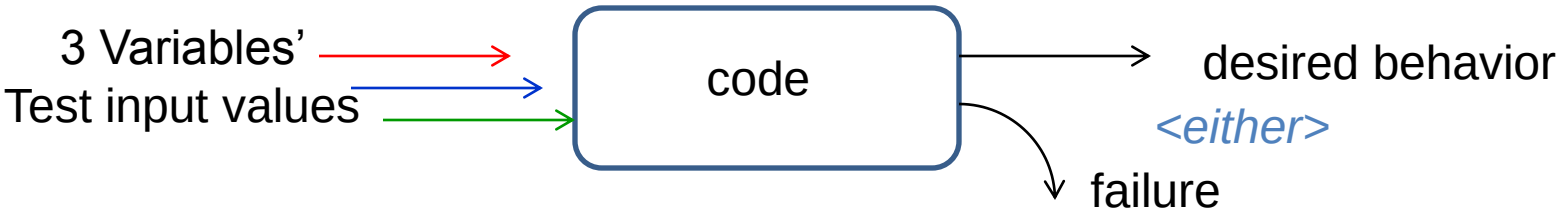
NIST Software Testing Work

Goal: Reduce testing cost

- **Key finding:**
  Most failures found are triggered by one or two variables, and progressively fewer by three, four, or more variables, and the maximum interaction degree is small.



Examples of 't-way' Variable Value Interaction-Driven Failures



Variables >	Pressure	Volume	Velocity	Result	't-way' Interaction
Three possible failure scenarios	< 10			Failure	1 - Way
	< 10	> 300		Failure	2 - Way
	< 10	> 300	> 5	Failure	3 - Way

- The NIST Combinatorial Coverage Tool measures the capacity of a test suite for detecting faults due to t-way interactions.
- This is different from code coverage measures such as statement or branch coverage.

NIST Tool Mechanics

In excel, define Input Test Variables and their Values for test cases:

	V1	V2	V3
1	ENABLE	SIDE A	AVERAGE
2	ENABLE	SIDE A	MINIMUM
3	ENABLE	SIDE B	MINIMUM
4	DISABLE	SIDE A	AVERAGE
5	DISABLE	SIDE B	MINIMUM

Save Values in .csv file format

ENABLE, ENABLE, ENABLE, DISABLE, DISABLE,
Side A, Side A, Side B, Side A, Side B,
AVERAGE, MINIMUM, MINIMUM, AVERAGE, MINIMUM

Input .csv file into NIST Tool which compares the Test Values against all possible values:

2-way (12)

ENABLE	SIDE A
ENABLE	SIDE B
ENABLE	AVERAGE
ENABLE	MINIMUM
DISABLE	SIDE A
DISABLE	SIDE B
DISABLE	AVERAGE
DISABLE	MINIMUM
SIDE A	AVERAGE
SIDE A	MINIMUM
SIDE B	AVERAGE
SIDE B	MINIMUM

(11)

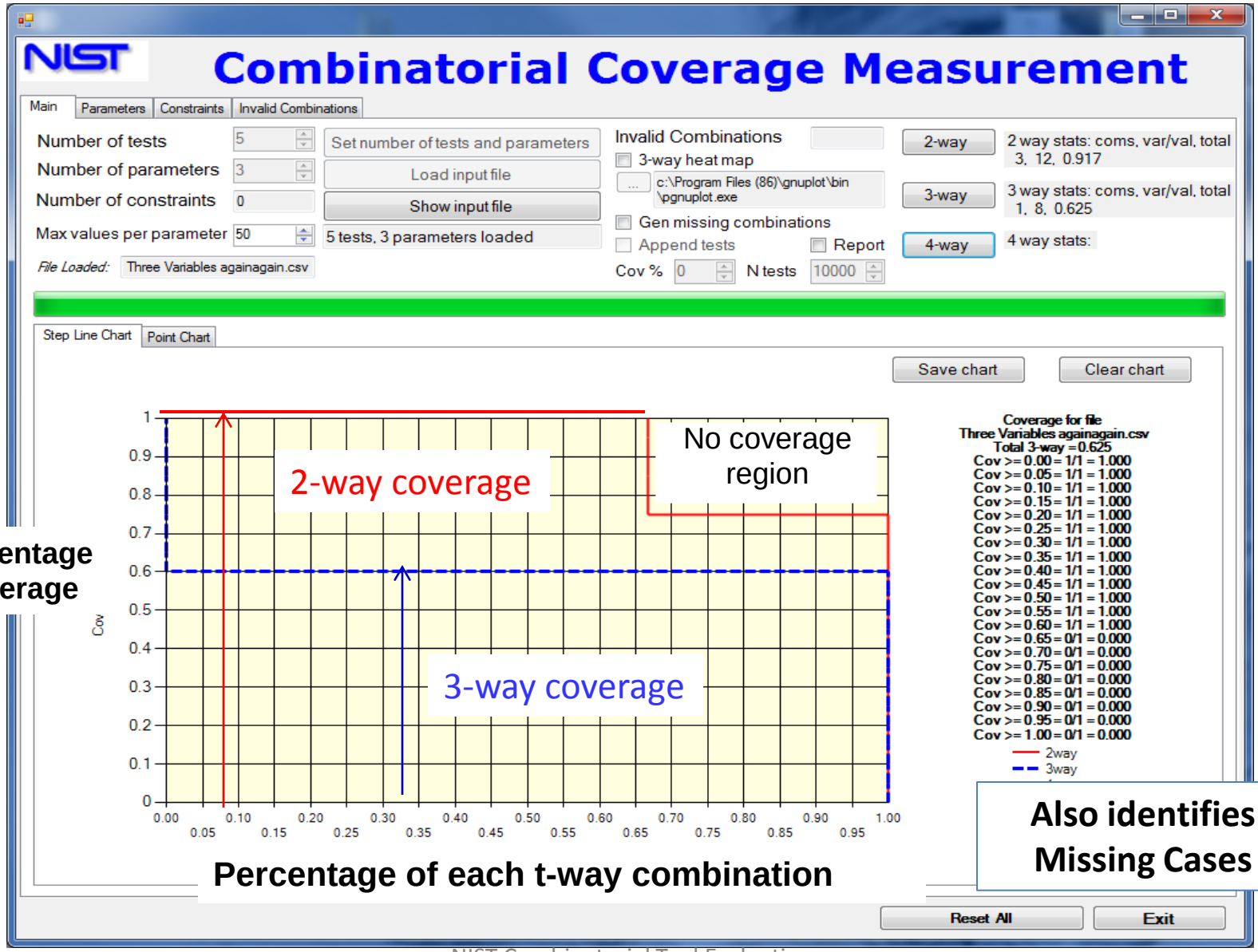
3-way (8)

ENABLE	SIDE A	AVERAGE
ENABLE	SIDE A	MINIMUM
ENABLE	SIDE B	AVERAGE
ENABLE	SIDE B	MINIMUM
DISABLE	SIDE A	AVERAGE
DISABLE	SIDE A	MINIMUM
DISABLE	SIDE B	AVERAGE
DISABLE	SIDE B	MINIMUM

(5)

NIST Tool Output Chart

NIST Tool Output Chart



Percentage coverage

Percentage of each t-way combination

Also identifies Missing Cases

IV&V use of NIST Tool Conclusions:

"The NIST Tool provides a structure for recording and reporting test coverage."

1. As an inline IV&V analysis tool (including peer reviews):
   1) Example: analyze coverage/lack of coverage of developer tests
   2) Expect moderate overhead for worthwhile value added.
2. As an inline Verification tool in JSTAR/ITS:
   1) Example: identify the coverage of tests as they are defined
   2) Expect low overhead for high value added for test planning.
3. As an IV&V audit tool:
   1) Example: auditing completed IV&V analysis of developer tests
   2) Significant overhead for value added.
4. As a macroscopic IV&V tool:
   1) Example: analyze project or multi-project test plans
   2) Expect a moderate overhead for additional high level insight.

IV&V use of NIST Tool Recommendations:

"The value of the NIST Tool must be proven in IV&V field trials."

1. Add the NIST Tool and Instructions for Use into the Catalog of Methods.

2. Use the Tool during active IV&V projects (including peer reviews)
   1) For evaluation of test coverage of developer's test plans.
   2) During analysis of developer test case scripts.

3. Implement Tool in JSTAR/ITC as verification test planning aid to measure coverage of tests as they are planned and before they are run
   1) During inhouse testing of developer flight software
   2) During inhouse development of test tools and other support software
   3) During development of simulators

4. Consider
   1) Use of Tool in comparing developer tests plans among projects.
   2) Collaboration with USAF and JHU/APL regarding Tool experiences
   3) Introducing Tool to software developers

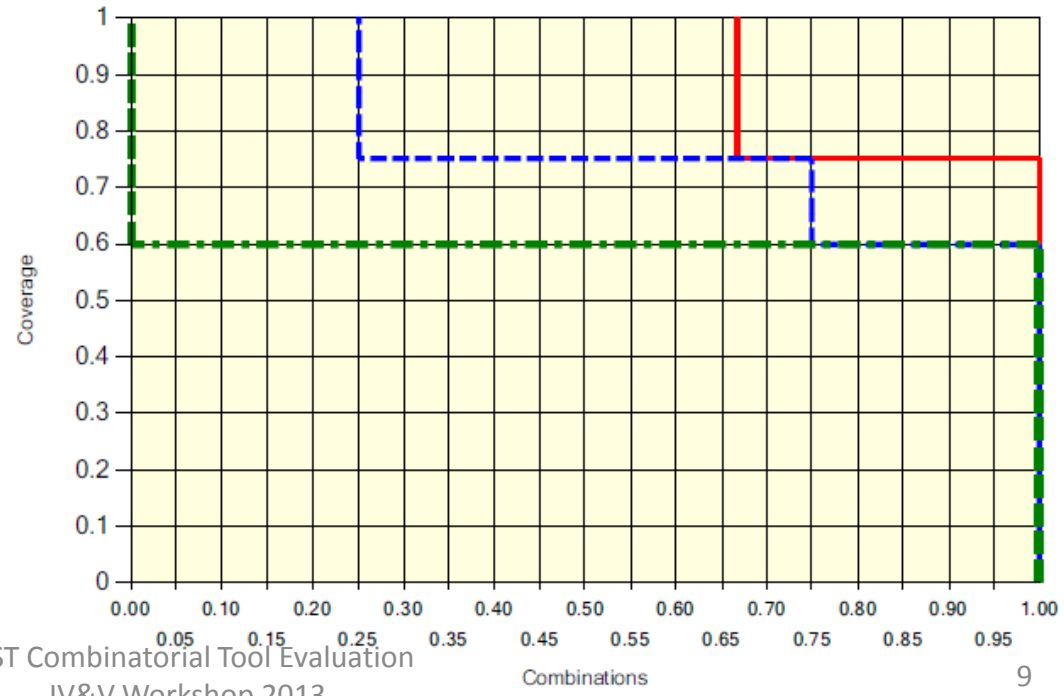
Combinatorial Coverage Measurement Example

Rick Kuhn

Coverage of flag combinations

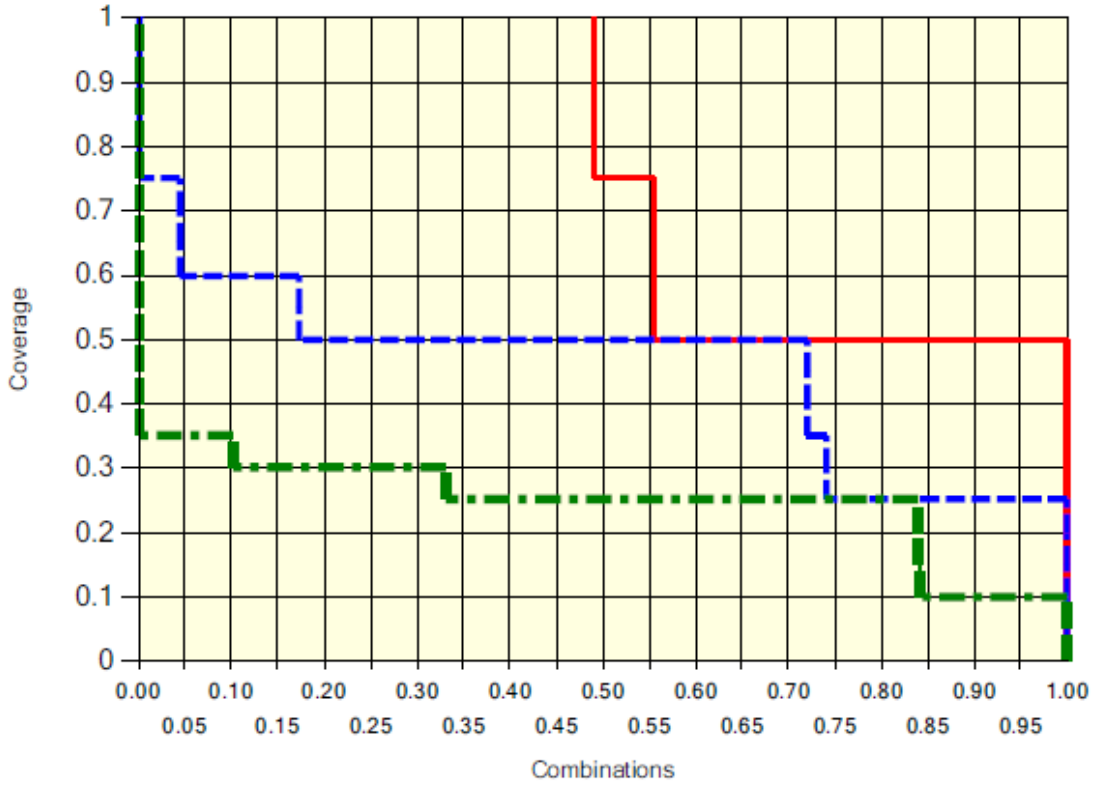
- Four flags: control enable/disable; control side; telemetry side; control flag (avg/min)
- Reasonably good : 88.9% (2-way), 75% (3-way), 62.5% (4-way)
- Review of test values shows only one test for telemetry side B, so coverage would be higher if supplemented with more side B tests

- If flags affect execution sequence of software, combinations may be significant;
- Suggests need for additional tests



14 tests, valid/invalid value combination coverage

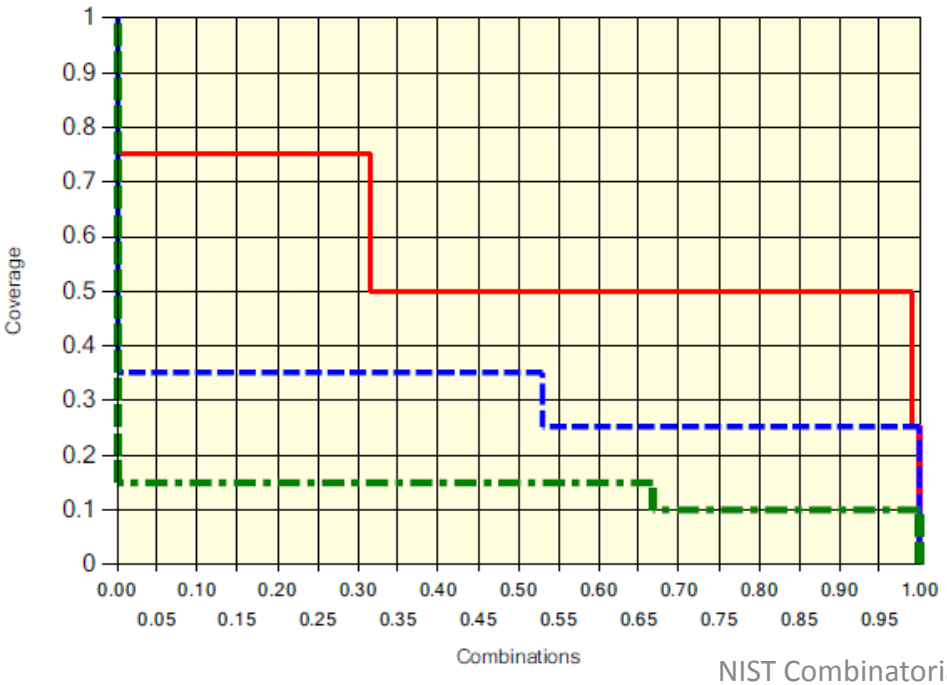
- 2, 3, 4-way coverage = 76.1%, 45.9%, 25.7%
- Combinations of values probably less significant than for flags that control s/w



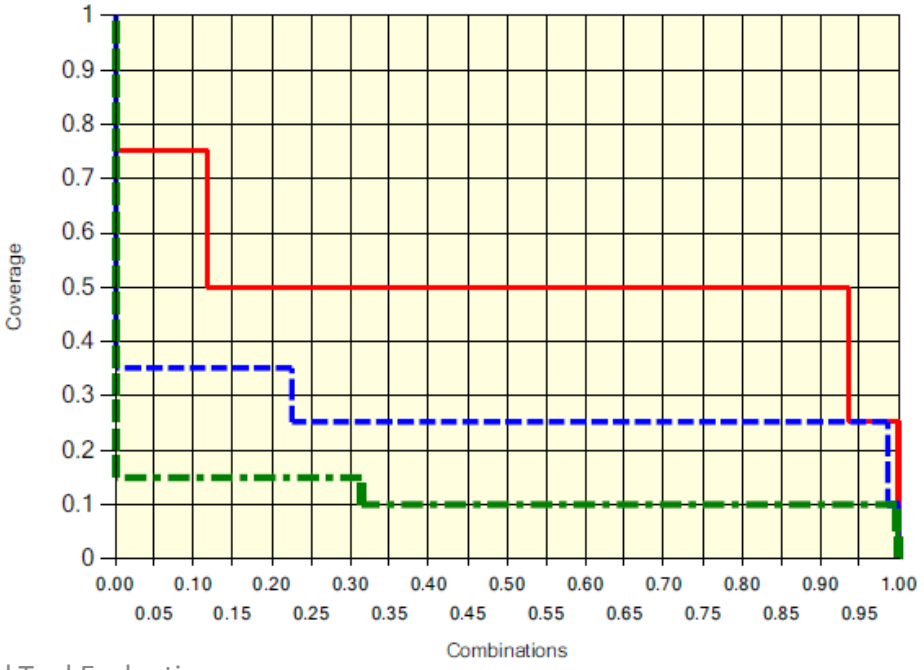
Comparison of coverage: tests for MINIMUM and AVERAGE

- Coverage similar although 4 tests with MINIMUM flag, 10 with AVERAGE flag
- Coverage significantly lower than for flags

MINIMUM: 57.6%, 31.6%, 16.7%



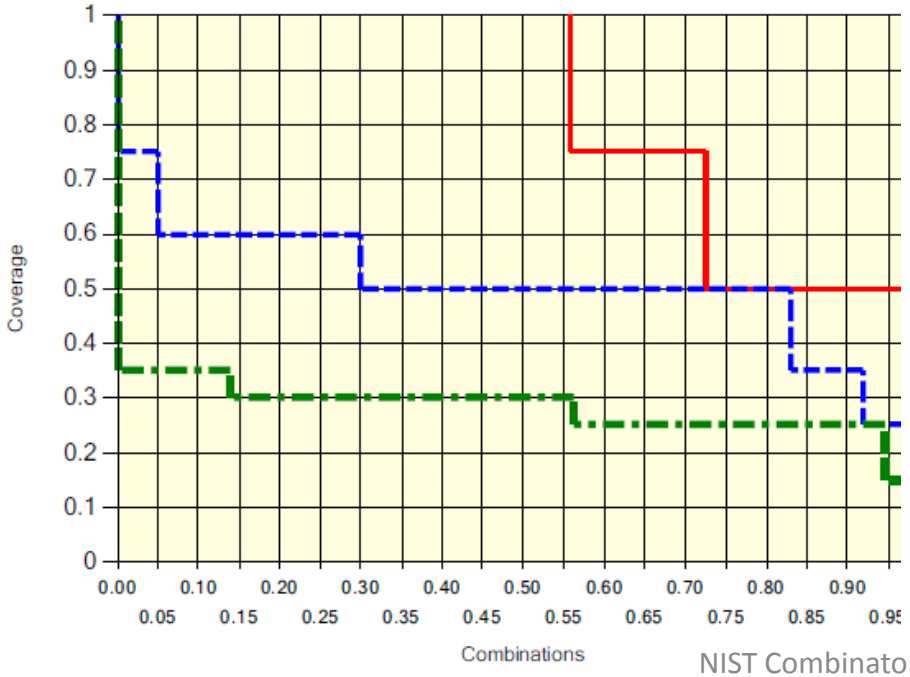
AVERAGE: 51.3%, 27.6%, 14.4%



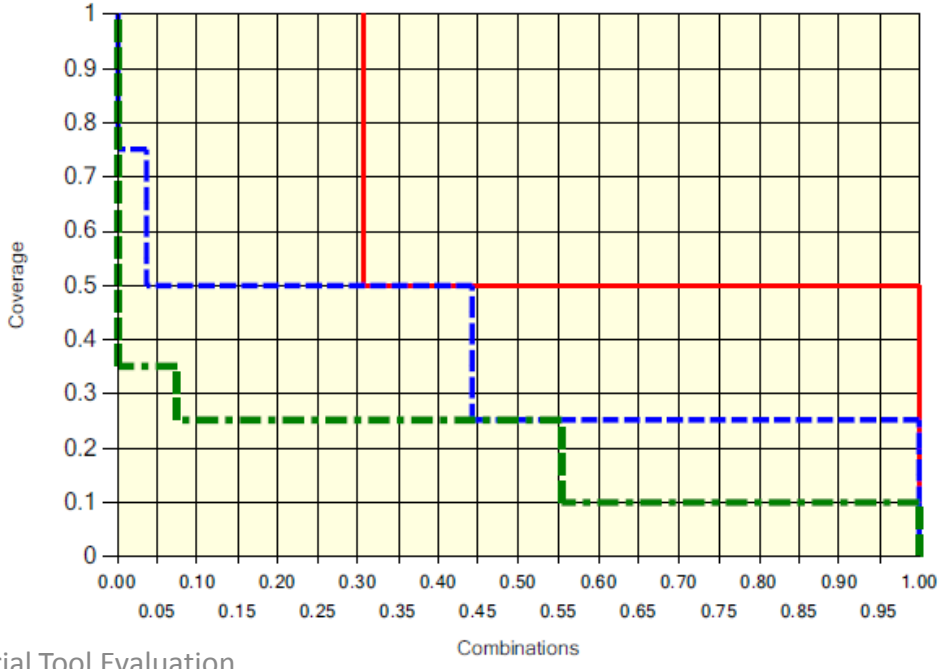
Comparison of coverage: tests for type A and B sensors

- Significant difference in coverage
- Suggests tests are more thorough for A type; higher risk of untested situations for B
- More tests for B may be helpful

A: 82.1%, 51.2%, 28.9%



B: 65.3%, 36.9%, 20.3%



Difference in coverage for type A and B not obvious

A:

1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0	1	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

B:

1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CCM Tool

Columns=parameters

Rows=tests

a,b,c,e,f
c,b,e,a,b
b,a,f,c,e

Load input file

Parallel processing (optional)

t-way measurement

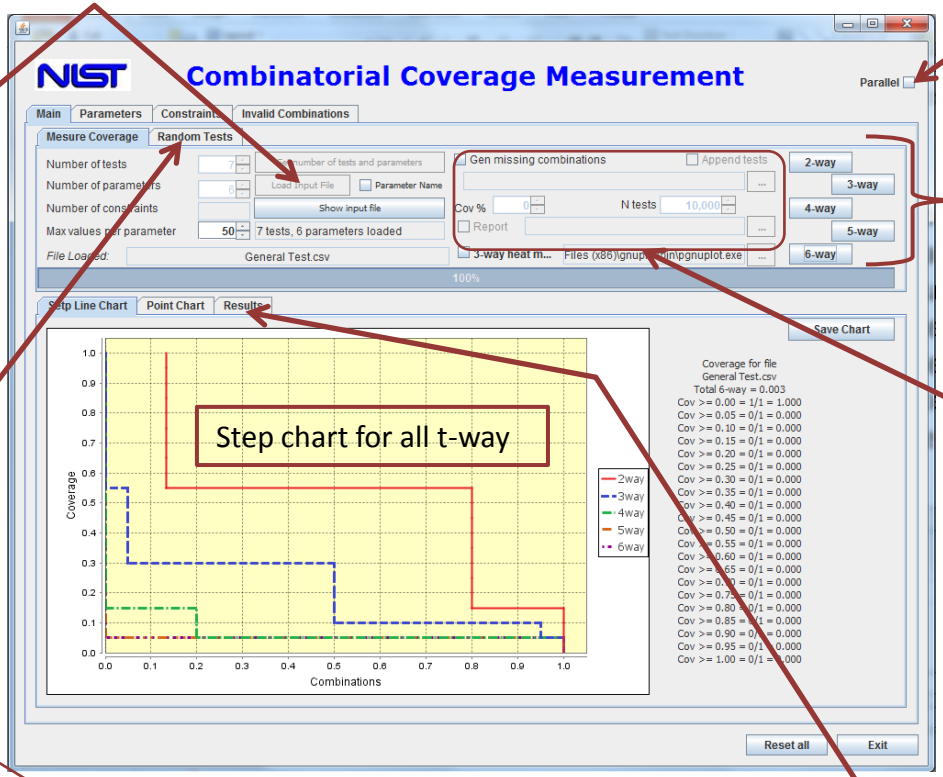
Main Screen

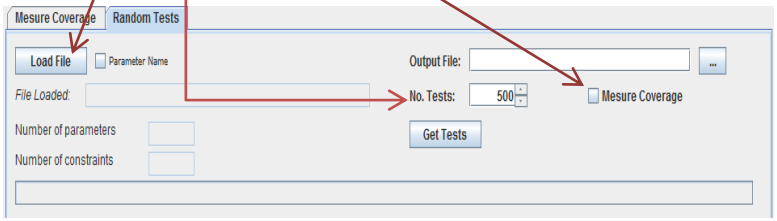
Generate missing combinations

Random Tests

This feature gets an **input file** with all the parameters and their values and generates a random test file with the **number of tests specified**. It can be specified to **measure the coverage** from the test set obtained.

Results

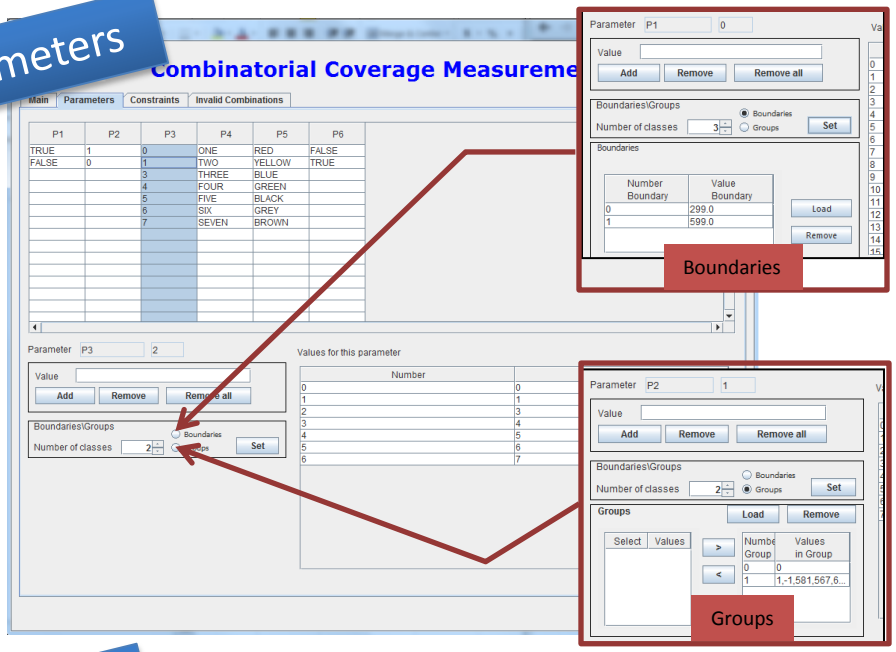




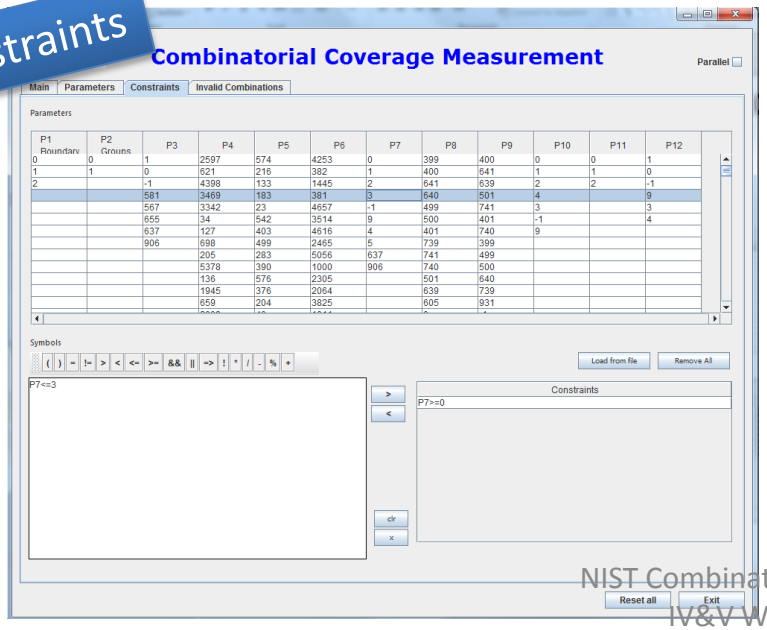
t-way	Coms.	Var/Val	Var/Val Cov.	Invalid Tests	Total	Time Constraints	Time t-way	Total Time
2-way	15	285	94	0	32.9824561%	00:00:00.000000	00:00:00.000017	00:00:00.000017
3-way	20	1485	137	0	9.2255892%	00:00:00.000000	00:00:00.000001	00:00:00.000001
4-way	15	3990	105	0	2.6315789%	00:00:00.000000	00:00:00.000001	00:00:00.000001
5-way	6	5292	42	0	0.7936508%	00:00:00.000000	00:00:00.000002	00:00:00.000002
6-way	1	2744	7	0	0.255102%	00:00:00.000000	00:00:00.000002	00:00:00.000002

CCM Tool

Parameters



Constraints



EXAMPLE, TCAS [Siemens suite], universe file

TCAS configuration:
- Boundaries
- Groups
- Constraints

	Parameter	Type	Boundaries/Groups/ Constraints
P1	Cur_Vertical_Sep Cur_Vertical_Sep > MAXALTDIFF [600] Cur_Vertical_sep > = MINSEP [300]	Int	Value / Bound 0 / 299 / x<=299 1 / 599 / 299<x<=599 2 / x>599
P2	High_confidence	Bool	X=0 -> FALSE , otherwise -> TRUE No. Group / Value 0 / X=0 1 / x<0 , x>0
P3	Two_of_three_reports_valid	Bool	X=0 -> FALSE , otherwise -> TRUE No. Group / Value 0 / X=0 1 / x<0 , x>0
P4	Own_Tracked_Alt Own_Tracked_Alt < Other_Tracked_Alt [own_below_threat] [bool] Other_Tracked_Alt < Own_Tracked_Alt [own_above_threat][bool]	Int	Value / Bound 0 / 0 / x<=0 1 / x>0
P5	Own_tracked_alt_rate Own_tracked_alt_rate <= OLEV [600]	Int	Value / Bound 0 / 600 / x<=600 1 / x>600
P6	Other_tracked_alt Own_Tracked_Alt < Other_Tracked_Alt [own_below_threat][bool] Other_Tracked_Alt < Own_Tracked_Alt [own_above_threat][bool]	Int	Value / Bound 0 / 0 / x<=0 1 / x>0
P7	Alt_layer_value	Int	[0,1,2,3] Constraints => P7>=0 P7<=3
P8	Up_separation Up_Separation >= ALIM() [Positive_RA_Alt_Thresh[Alt_Layer_Value]] – 400,500,640,740 Up_Separation + NOZCROSS [100]	Int	Value / Bound 0 / 399 / X<399 1 / 499 / 400<x<=499 2 / 639 / 500<x<=639 3 / 739 / 640<x<=739 4 / 839 / 739<x<=839 5 / 840 / x>840
P9	Down_separation Inhibit_Biased_Climb() > Down_Separation Down_Separation >= ALIM() [Positive_RA_Alt_Thresh[Alt_Layer_Value]] – 400,500,640,740	Int	Value / Bound 0 / 399 / X<399 1 / 499 / 400<x<=499 2 / 639 / 500<x<=639 3 / 739 / 640<x<=739 4 / 839 / 739<x<=839 5 / 840 / x>840
P10	Other_RAC Other_RAC == NO_INTENT [0]	Int	No. Group / Value 0 / X=0 1 / x<0 , x>0
P11	Other_capability Other_Capability == TCAS_TA [1]	Int	No. Group / Value 0 / X=1 1 / x<1 , x>1
P12	Climb_inhibit	Int [declared as int but used as bool]	X=0 -> FALSE , otherwise -> TRUE No. Group / Value 0 / X=0

t-way	Coms.	Var/Val	Var/Val Cov.	Invalid Tests	Total
2-way	66	552	531	35	96.1956522%
3-way	220	4755	4359	35	91.6719243%
4-way	495	26462	21668	35	81.8834555%
5-way	792	105050	69101	35	65.7791528%
6-way	924	312451	145844	35	46.677399%

TCAS Results

Coverage

Step Chart