Combinatorial testing

SWE261P

Combinatorial Testing

- Partitions for multiple categories presents the opportunity for each partition to be tested
- How to ***combine*** them becomes the next choice
- 1-way testing, 2-way testing, 3-way testing, 4-way testing, ... Up to the number of categories.
- "2-way testing" is more commonly known as "***pairwise testing***"

Why Combinatorial?

- Testing all combinations is often prohibitively expensive

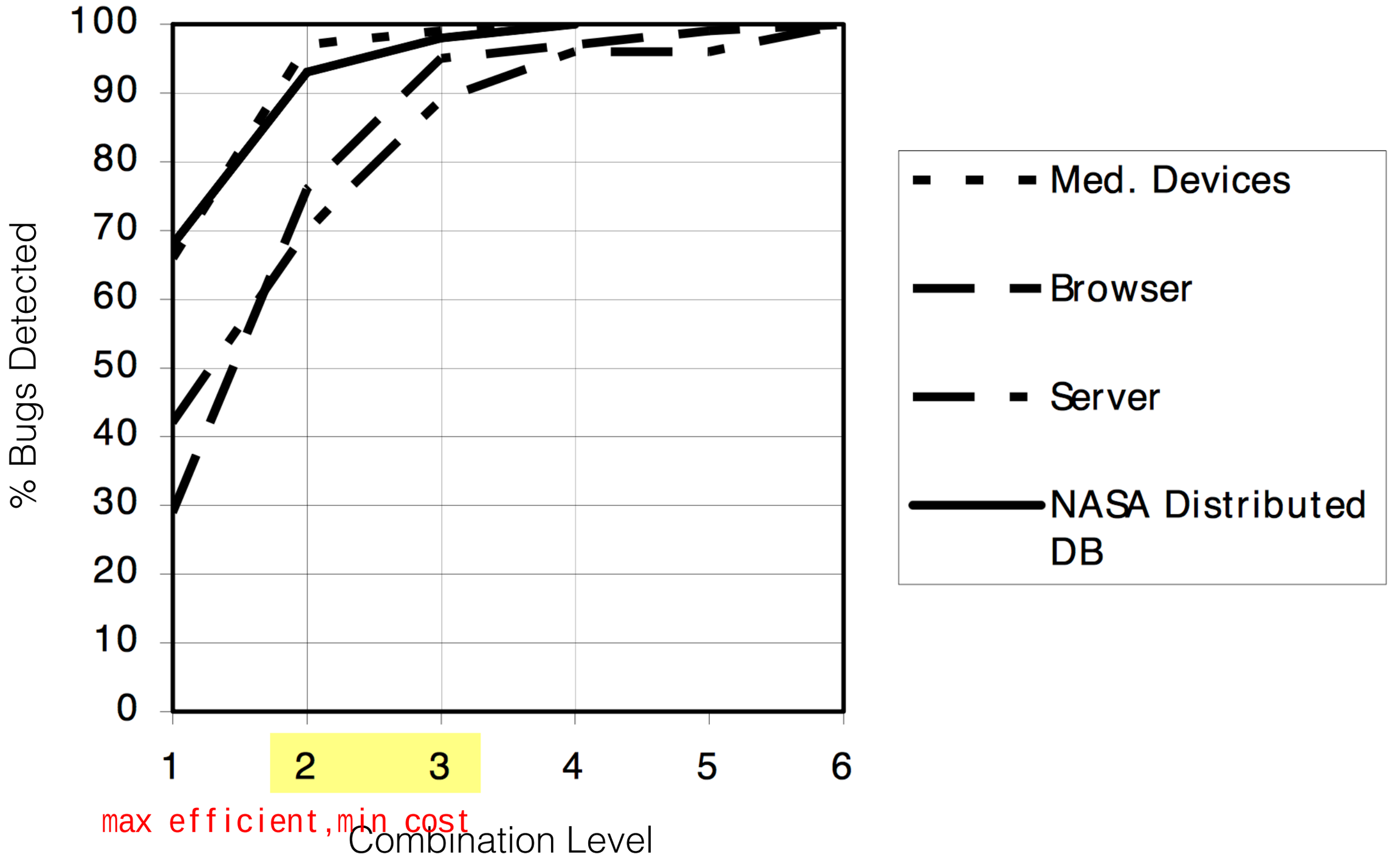
- With real constraints on the costs of testing (tests to write and run) — we can be smart about *which* test cases to run

Report by NIST

(National Institute of Standards and Technology, "Practical Combinatorial Testing")

- "... Some of these rare combinations trigger failures that have escaped previous testing and extensive use. Such failures are known as interaction failures, because they are only exposed when two or more input values interact to cause the program to reach an incorrect result."

- "Failures appear to be caused by interactions of only a few variables, so tests that cover all such few-variable interactions can be very effective."



[adapted from National Institute of Standards and Technology, "Practical Combinatorial Testing”]

Forms of Combinatorial Testing

- Configuration Testing: Often hardware & software configurations

- Input Parameter Testing: Input values, including boundary values

- Sequence Testing: Input event sequences

OS	Browser	Protocol	CPU	DBMS
Win	IE	IPv4	Intel	MySQL
MacOS	Firefox	IPv6	AMD	Sybase
Linux				Oracle

# of exhaustive configurations =
$3 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 72$

t	# Tests	% of Exhaustive
2	9	12.5
3	18	25
4	36	50
5	72	100

Strategies

- Rearrange the categories from largest to smallest (or, equivalently, *address* the categories from largest to smallest)

- Work on creating all pairs for the two largest categories, and then repeatedly introduce one more (next largest) category.

- Always try to “cover” pairs that have not yet been covered. When there are no options that do not produce a repeat, choose a value that repeats the least (or one of them).

OS (Win, MacOS, Linux)	Browser (IE, Firefox)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	DBMS (MySQL, Sybase, Oracle)

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)
Win	My	4	In	IE
Win	Sy	6	AMD	F
Win	Or	4	AMD	IE
Mac	My	6	In	F
Mac	Sy	4	In	F
Mac	Or	6	AMD	IE
Lin	My	4	AMD	F
Lin	Sy	6	In	IE
Lin	Or	4	In	F

OS (Win, MacOS, Linux)	Browser (IE, Firefox)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	DBMS (MySQL, Sybase, Oracle)

Rearrange the above categories from largest to smallest

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)

Starting from the left, create all pairs of just the first 2 columns, then 3 columns, ...

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)

Starting from the left, create all pairs of just the first 2 columns, then 3 columns, ...

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)

…then 4 columns

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)

OS (Win, MacOS, Linux)	DBMS (MySQL, Sybase, Oracle)	Protocol (IPv4, IPv6)	CPU (Intel, AMD)	Browser (IE, Firefox)
Win	M	4	I	IE
Win	S	6	A	F
Win	O	4	A	IE
Mac	M	6	I	F
Mac	S	4	I	F
Mac	O	6	A	IE
Lin	M	4	A	F
Lin	S	6	I	IE
Lin	O	4	I	F

Parameter Name	Values	# Values
HARDKEYBOARDHIDDEN	NO, UNDEFINED, YES	3
KEYBOARDHIDDEN	NO, UNDEFINED, YES	3
KEYBOARD	12KEY, NOKEYS, QWERTY, UNDEFINED	4
NAVIGATIONHIDDEN	NO, UNDEFINED, YES	3
NAVIGATION	DPAD, NONAV, TRACKBALL, UNDEFINED, WHEEL	5
ORIENTATION	LANDSCAPE, PORTRAIT, SQUARE, UNDEFINED	4
SCREENLAYOUT_LONG	MASK, NO, UNDEFINED, YES	4
SCREENLAYOUT_SIZE	LARGE, MASK, NORMAL, SMALL, UNDEFINED	5
TOUCHSCREEN	FINGER, NOTOUCH, STYLUS, UNDEFINED	4

t	# Tests	% of Exhaustive
2	29	0.02
3	137	0.08
4	625	0.4
5	2532	1.5
6	9168	5.3

# of exhaustive configurations =
$3 \cdot 3 \cdot 4 \cdot 3 \cdot 5 \cdot 4 \cdot 4 \cdot 5 \cdot 4 = 172{,}800$

Adding constraints

- Simple constraints

  example: IE is not compatible with MacOS and Linux

  can be handled by considering the case in separate tables or eliminating violating combinations from a unified table

Example: IE only compatible with Win

OS	**Browser**	**Protocol**	**CPU**	**DBMS**
		IPv4	Intel	MySQL
MacOS	Firefox	IPv6	AMD	Sybase
Linux				Oracle

OS	Browser	Protocol	CPU	DBMS
Win	IE	IPv4	Intel	MySQL
	Firefox	IPv6	AMD	Sybase
				Oracle

The problem can be broken down into separate combinatorial problems. Solve each to ensure that constraints are not violated.

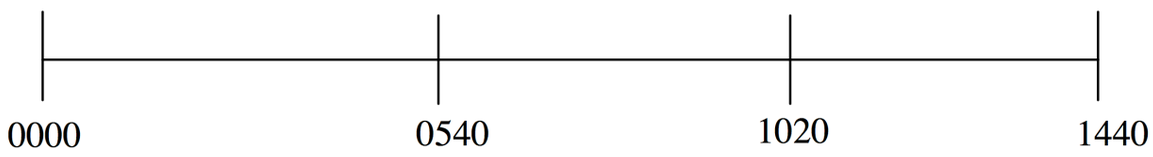
Input Parameter Combinatorial Testing

Access is allowed if and only if:

- the subject is an employee
  AND current time is between 9 am and 5 pm
  AND it is not a weekend
- OR subject is an employee with a special authorization code
- OR subject is an auditor
  AND the time is between 9 am and 5 pm
  (not constrained to weekdays).

```
emp:  boolean;
time:  0..1440;  // time in minutes
day:   {m,tu,w,th,f,sa,su};
auth: boolean;
aud:  boolean;
```

Parameter	Values
emp	0,1
time	??
day	`m,tu,w,th,f,sa,su`
auth	0, 1
aud	0, 1



Tools

- A large list of combinatorial testing tools are shown, and some compared at: http://www.pairwise.org/tools.html

iPhone Size (Std, Max, SE)	OS version (16, 15, 14)	Connected to network (yes, no)	GPS available (yes, no)