

Configuration-Aware Regression Testing

An Empirical Study of Sampling and Prioritization

Xiao Qu

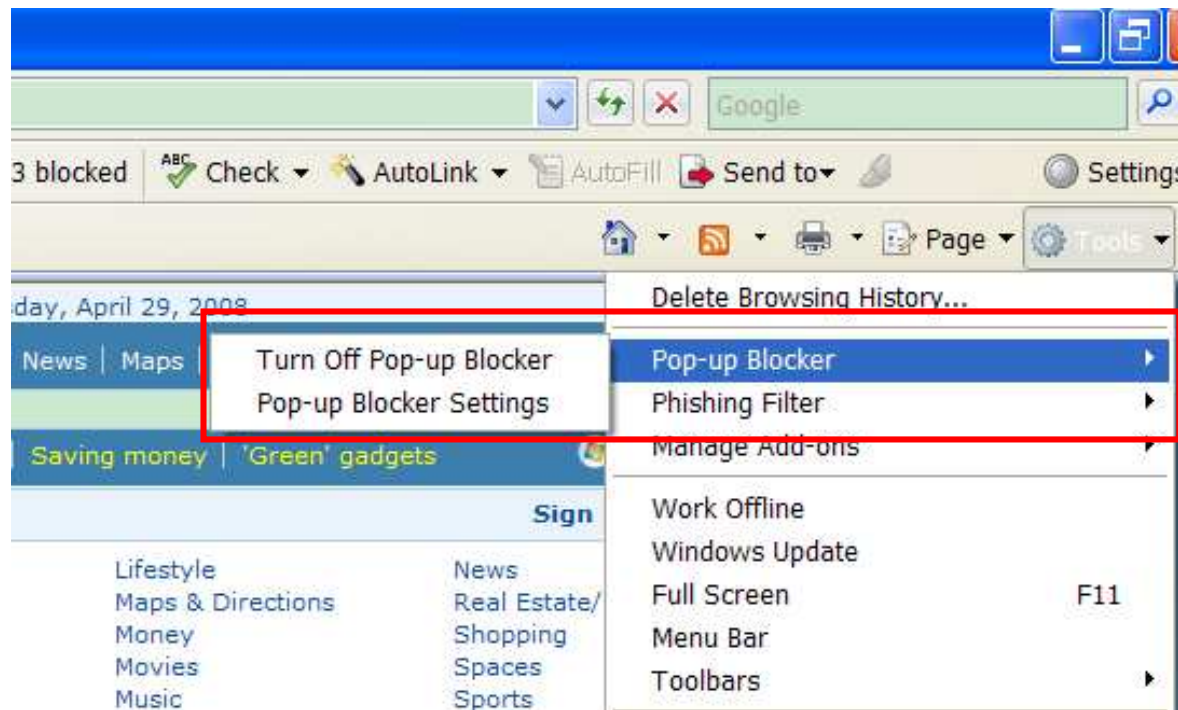
Myra B. Cohen

Gregg Rothermel



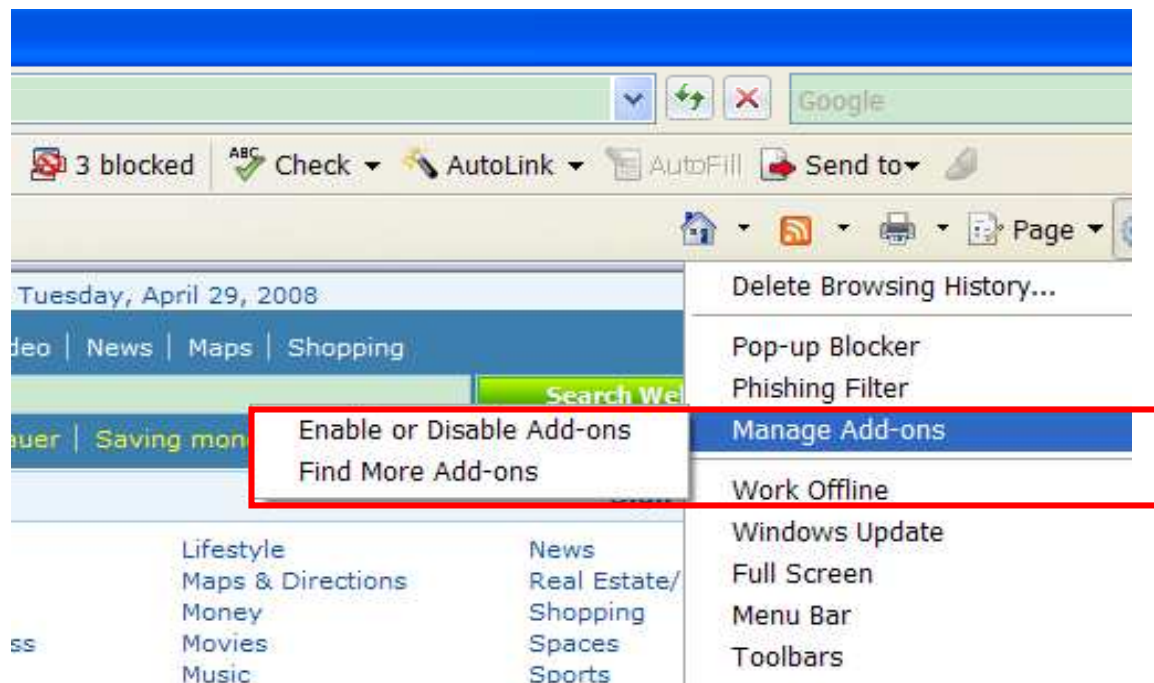
Configurable Software

- Software that can be customized through a set of options
- Examples:
 - Internet Explorer



Configurable Software

- Software that can be customized through a set of options
- Examples:
 - Internet Explorer



Lavasoft Support Forums

File Edit View Favorites Tools Help

IE7 + Google Toolbar - ... x Scott Hanselman's Compu...

LAVASOFT


IPB Portal Terms of Service Calendar Members Search Help

Welcome Guest ([Log In](#) | [Register](#))

► [Lavasoft Support Forums](#) > [Security News](#) > [Windows Updates](#)

[ADDREPLY](#) [NEWTOPIC](#)

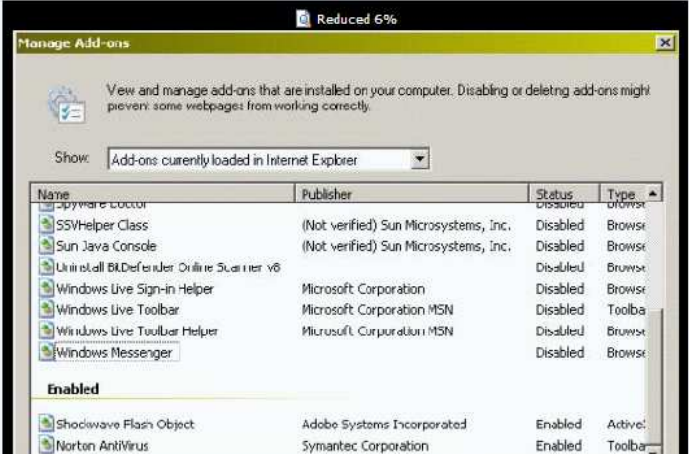
IE7 + Google Toolbar - Potential conflict Options ▼

 **spike-nz** Nov 13 2006, 02:40 AM Post #1

I have discovered from personal experience, that using the newly released Internet Explorer 7 with the current version of Google Toolbar (v4.0.1020.5070) can cause the right-click menu to lose the "Open in New Tab" option.

Until now, to open a new Tab in the same window, I had to hold down "Ctrl" whilst clicking on the Link.

For those of you who have encountered this problem, open IE7, click "Tools > Manage Add-ons > Enable or Disable Add-ons"



Name	Publisher	Status	Type
Spyware Doctor		Disabled	Browser
SSVHelper Class	(Not verified) Sun Microsystems, Inc.	Disabled	Browser
Sun Java Console	(Not verified) Sun Microsystems, Inc.	Disabled	Browser
Live! install & Defender Online Scanner v6		Disabled	Browser
Windows Live Sign-in Helper	Microsoft Corporation	Disabled	Browser
Windows Live Toolbar	Microsoft Corporation MSN	Disabled	Toolba
Windows Live Toolbar Helper	Microsoft Corporation MSN	Disabled	Browser
Windows Live Toolbar Helper	Microsoft Corporation MSN	Disabled	Browser
Windows Messenger		Disabled	Browser
Enabled			
Shockwave Flash Object	Adobe Systems Incorporated	Enabled	Active!
Norton AntiVirus	Symantec Corporation	Enabled	Toolba

Lavasoft Support Forums

File Edit View Favorites Tools Help

IE7 + Google Toolbar - ... x Scott Hanselman's Compu...

LAVASOFT

IPB Portal Terms of Service Calendar Members Search Help

Welcome Guest ([Log In](#) | [Register](#))

► [Lavasoft Support Forums](#) > [Security News](#) > [Windows Updates](#)

ADDREPLY NEWTOPIC

IE7 + Google Toolbar - Potential conflict Options

[spike-nz](#)

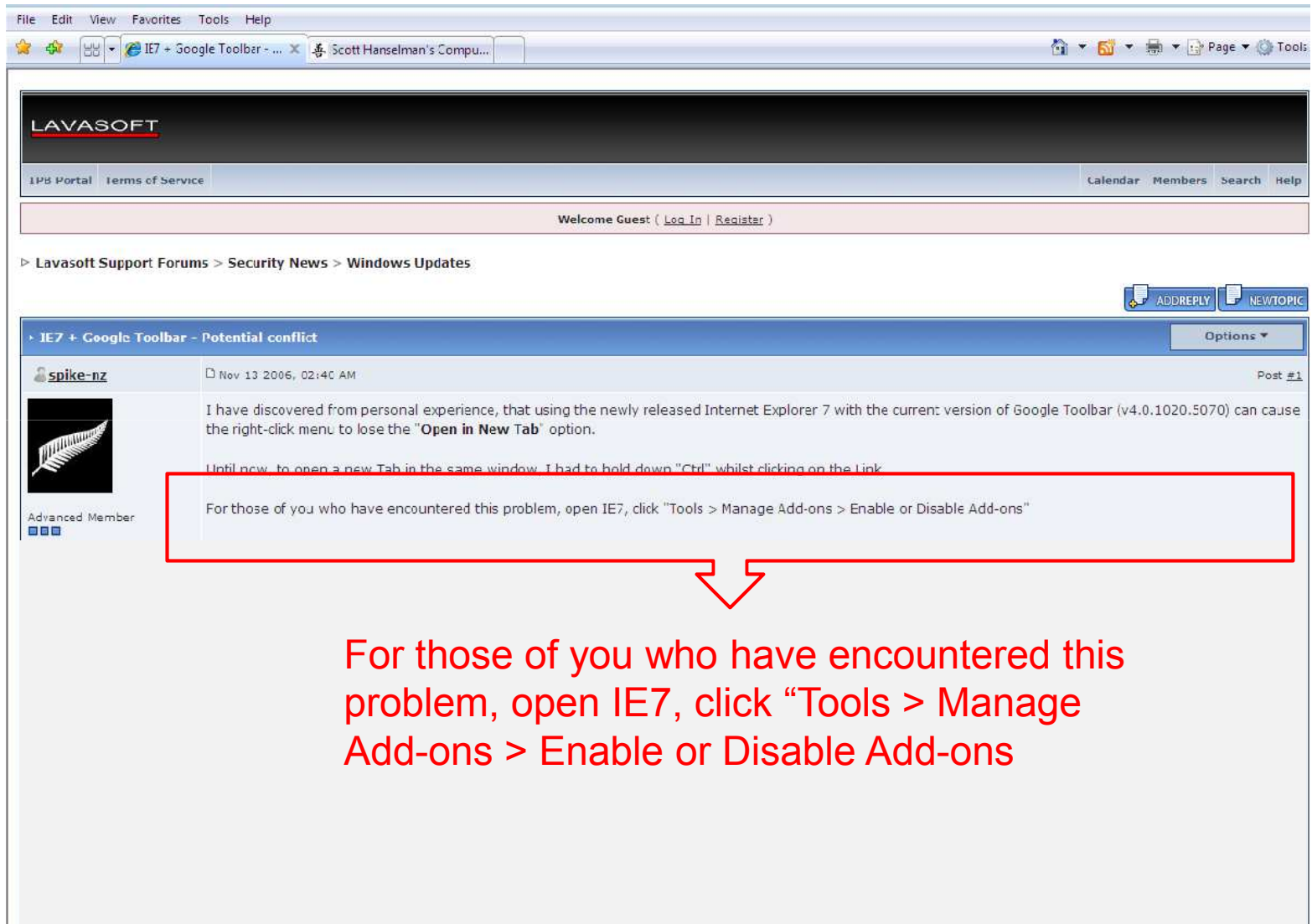
Nov 13 2006, 02:40 AM Post #1

I have discovered from personal experience, that using the newly released Internet Explorer 7 with the current version of Google Toolbar (v4.0.1020.5070) can cause the right-click menu to lose the "Open in New Tab" option.

Until now, to open a new Tab in the same window, I had to hold down "Ctrl" whilst clicking on the Link.

I have discovered that using the newly released IE 7 with the current version of Google Toolbar can cause the right-click menu to lose the "Open In New Tab" option

Lavasoft Support Forums



File Edit View Favorites Tools Help

IE7 + Google Toolbar - ... x Scott Hanselman's Compu...

LAVASOFT


IPB Portal Terms of Service Calendar Members Search Help

Welcome Guest ([Log In](#) | [Register](#))

► [Lavasoft Support Forums](#) > [Security News](#) > [Windows Updates](#)

ADDREPLY NEWTOPIC

IE7 + Google Toolbar - Potential conflict Options

 spike-nz Nov 13 2006, 02:40 AM Post #1

I have discovered from personal experience, that using the newly released Internet Explorer 7 with the current version of Google Toolbar (v4.0.1020.5070) can cause the right-click menu to lose the "Open in New Tab" option.

Until now, to open a new Tab in the same window, I had to hold down "Ctrl" whilst clicking on the Link.

For those of you who have encountered this problem, open IE7, click "Tools > Manage Add-ons > Enable or Disable Add-ons"

Advanced Member

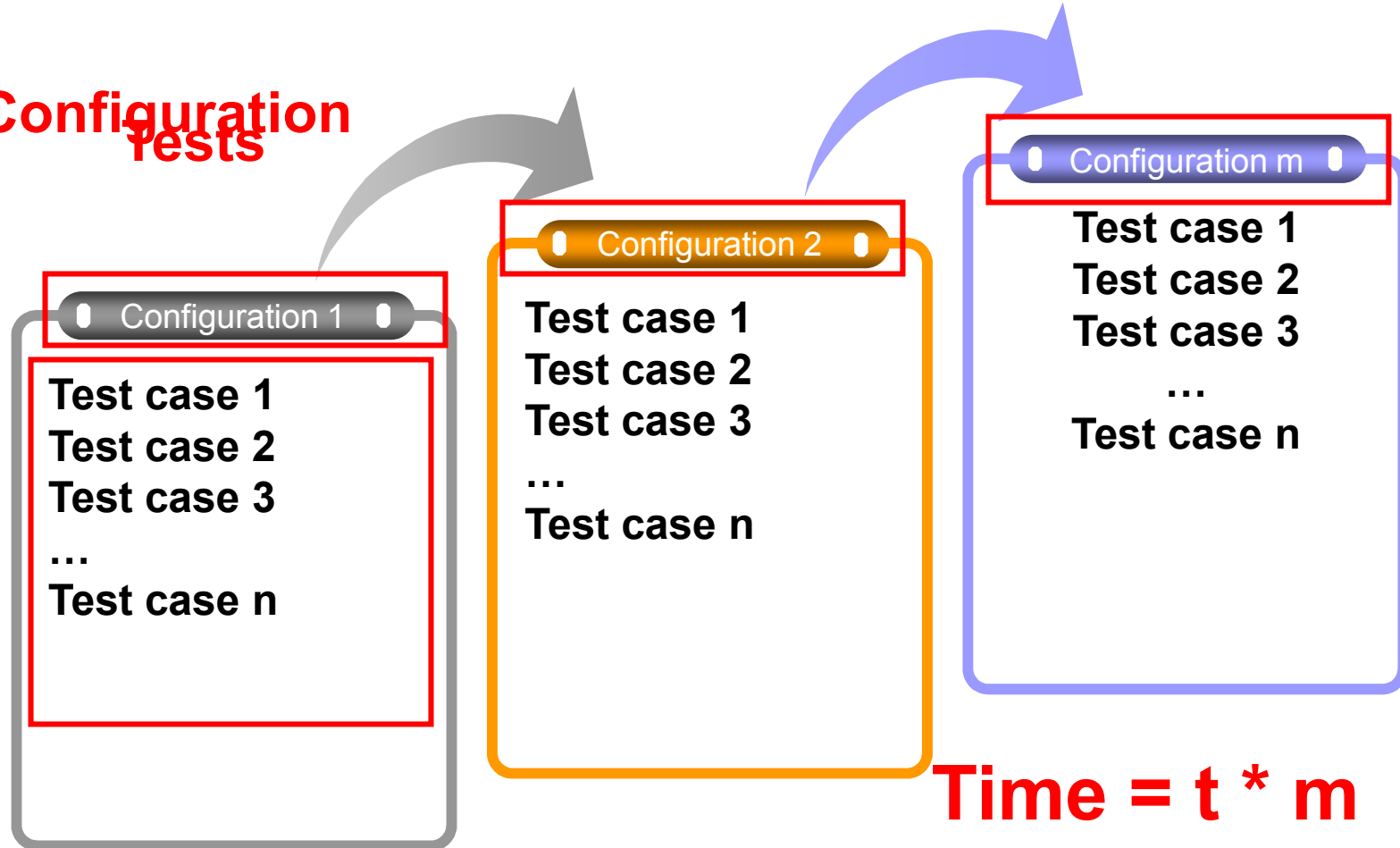
For those of you who have encountered this problem, open IE7, click "Tools > Manage Add-ons > Enable or Disable Add-ons"

Configurable Software

- From this example, we can see that
 - If we tested the software with “Google toolbar” **disabled**, we wouldn’t detect this incompatibility
 - If we tested it with “Google toolbar” **enabled**, we would detect it
- Some faults are detected **only** under certain configurations
- We may miss faults if we test **only** under one single configuration

Testing In Configurable Software

Configuration Tests



$$\text{Time} = t * m$$

t – execution time of running the whole test suite once
m – number of configurations

Example – IE7

Configuration Specification

Options	Filter Level	TIF	Pop-Up	Open Links
	High	VPage	IED	NWin
Values	Medium	VIE	NWin	NTab
	Low	Auto	NTab	CTab

Full Configuration Space

Options	Filter Level	TIF	Pop-Up	Open Links
Values	High	VPage	IED	NWin
	Medium	VIF	NWin	NTab
	Low	Auto	NTab	CTab

$3 * 3 * 3 * 3 = 81$

Sampling Configurations In Testing

- Configuration space is huge
 - 90 configurable options, each with 2 choices leads to 2^{90} configurations
- How do we sample configuration space?
 - Combinatorial Interaction Testing (CIT)

Outline



Motivation

Combinatorial Interaction Testing

Prioritization Techniques

Empirical Study Results

Conclusions and Future Work

CIT

- CIT is a technique to systematically sample all t-way combinations of configuration options
 - t – testing strength

2-way CIT Sample

	Filter Level	TIF	Pop-Up	Open Links
1	High	VPage	IED	NWin
2	Medium	VIE	NWin	NWin
3	Low	Auto	NTab	NWin
4	High	VIE	NTab	NTab
5	Medium	Auto	IED	NTab
6	Low	VPage	NWin	NTab
7	High	Auto	NWin	CTab
8	Medium	VPage	NTab	CTab
9	Low	VIE	IED	CTab

2-way CIT Sample

	Filter Level	TIF	Pop-Up	Open Links
1	High	VPage	IED	NWin
2	Medium	VIE	NWin	NWin
3	Low	Auto	NTab	NWin
4	High	VIE	NTab	NTab
5	Medium	Auto	IED	NTab
6	Low	VPage	NWin	NTab
7	High	Auto	NWin	CTab
8	Medium	VPage	NTab	CTab
9	Low	VIE	IED	CTab

9 < 81

2-way Combinations

Factors	Filter Level	TIF	Pop-Up	Open Links
Values	High	VPage	IED	NWin
	Medium	VIE	NWin	NTab
	Low	Auto	NTab	CTab

2-way CIT Sample

	Filter Level	TIF	Pop-Up	Open Links
1	High	VPage	IED	NWin
2	Medium	VIE	NWin	NWin
3	Low	Auto	NTab	NWin
4	High	VIE	NTab	NTab
5	Medium	Auto	IED	NTab
6	Low	VPage	NWin	NTab
7	High	Auto	NWin	CTab
8	Medium	VPage	NTab	CTab
9	Low	VIE	IED	CTab

2-way CIT Sample

	Filter Level	TIF	Pop-Up	Open Links
1	High	VPage	IED	NWin
2	Medium	VIE	NWin	NWin
3	Low	Auto	NTab	NWin
4	High	VIE	NTab	NTab
5	Medium	Auto	IED	NTab
6	Low	VPage	NWin	NTab
7	High	Auto	NWin	CTab
8	Medium	VPage	NTab	CTab
9	Low	VIE	IED	CTab

2-way CIT Sample

	Filter Level	TIF	Pop-Up	Open Links
1	High	VPage	IED	NWin
2	Medium	VIE	NWin	NWin
3	Low	Auto	NTab	NWin
4	High	VIE	NTab	NTab
5	Medium	Auto	IED	NTab
6	Low	VPage	NWin	NTab
7	High	Auto	NWin	CTab
8	Medium	VPage	NTab	CTab
9	Low	VIE	IED	CTab

Problem Even With CIT

- CIT reduces 2^{90} to **60** configurations
- Even with “only” 60 configurations, it’s still costly
 - Suppose each test suite is running for 7 hours
 - $t * m = 7 * 60 = 420$ hours, more than 2 weeks

Product Evolution

Internet Explorer



IE1.0 & 2.0



IE5



IE6



IE7

Problem In Regression Testing

- Even if we are able to test all 60 configurations in the very first version, in subsequent versions, we may not be able to afford this
 - **Budget** limitation → we can not retest all
 - Suppose we have 1 week, only 30 configurations can be tested → we want to select the most valuable ones

Prioritization

Configurations

Faults

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

Prioritization

Configurations

Faults

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

We only have **budget** to run two configurations

Prioritization

Configurations

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

We only have budget to run two configurations

Before ordering:

C1, C2 → F2, F7, F8

Prioritization

Configurations

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

We only have budget to run two configurations

Before ordering:
C1, C2 → F2, F7, F8

Detects 3 faults

Prioritization

Order C3 C5 C2 C4 C1

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

Prioritization

Order C3 C5 C2 C4 C1

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

After ordering:

C3, C5 → F2, F4, F5, F6, F7

Prioritization

Order C3 C5 C2 C4 C1

	C1	C2	C3	C4	C5
F1				X	
F2		X	X		
F3				X	
F4			X		X
F5					X
F6					X
F7	X		X		
F8	X				

After ordering:
C3, C5 → F2, F4, F5, F6, F7

Detects 5 faults

Our Contribution

- We developed prioritization techniques for regression testing CIT samples for configurable software
- We conducted a study to examine their effectiveness and efficiency

Outline



Motivation

Combinatorial Interaction Testing

Prioritization Techniques

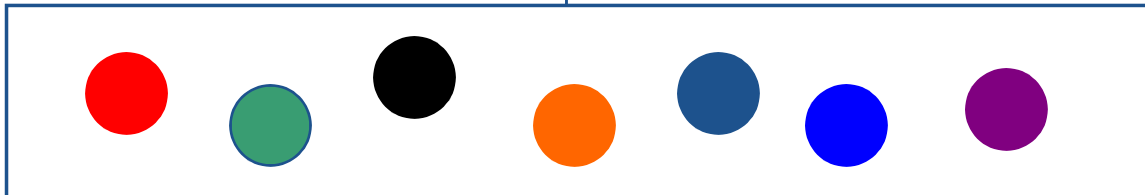
Empirical Study Results

Conclusions and Future Work

CIT sample configurations



CIT technique



Full configuration space

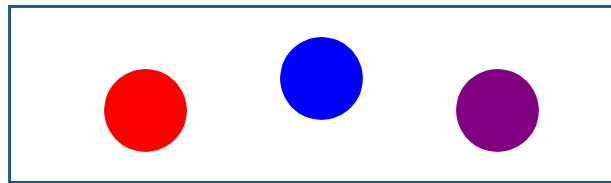
Technique 1



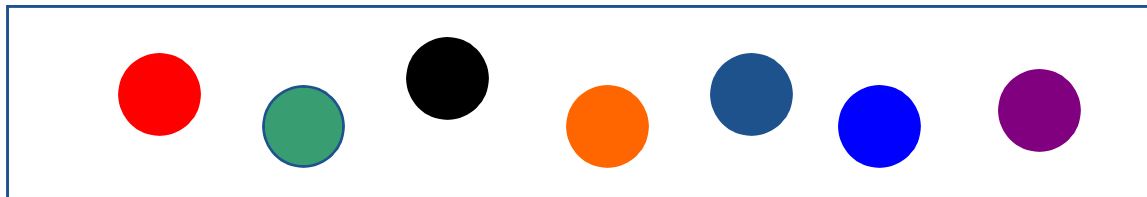
Reordered CIT sample



information

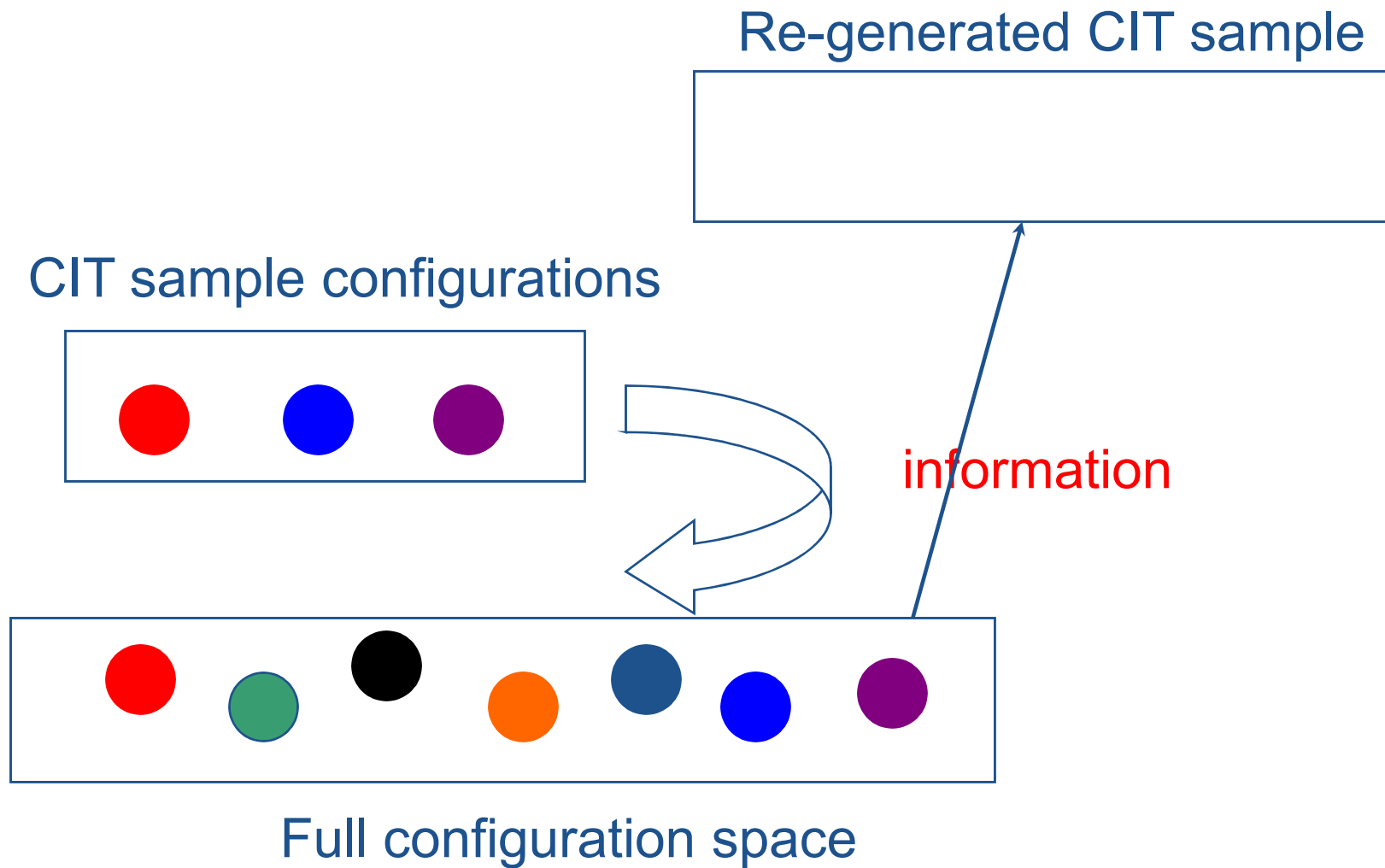


CIT sample configurations



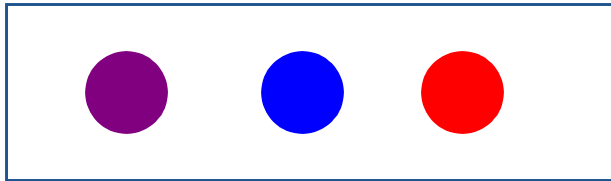
Full configuration space

Technique 2



Technique 1 vs. Technique 2

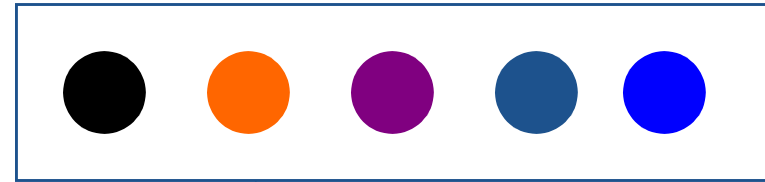
Reordered CIT sample



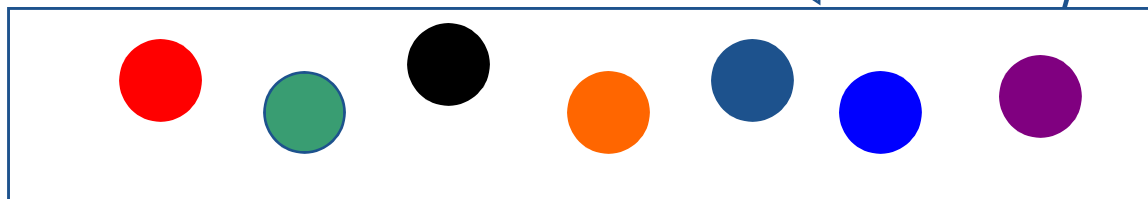
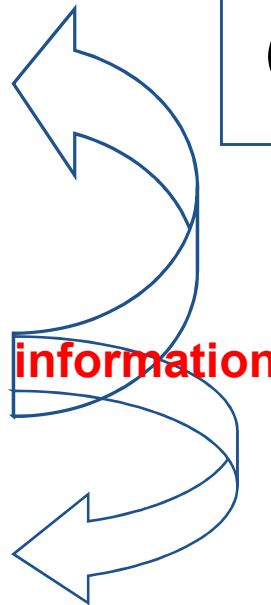
Technique 1:
"Pure" prioritization

CIT sample

Re-generated CIT sample



Technique 2:
Re-generation



Full configuration space

Interaction Benefit

Interaction Benefit

Two steps:

1. Assign **weights (information)** to each value of each option according to their importance
2. Calculate **interaction benefit** based on weights

Weighting Schemes

How do we set weights?

- Code coverage based
 - Ordered subset of original configuration (CIT) sample based on cumulative block coverage
- Fault detection based
 - Similar to code coverage based
 - Based on cumulative fault detection

Weighting Schemes

How do we set weights if we don't have information from the previous version?

- Specification based

Example Of Specification

Properties:

- pop-up blocker:
 - turn on
 - turn off
- privacy:
 - accept all cookies
 - block all cookies
 - low
 - high
 - Medium
 - medium high
-

Example Of Specification

Properties:

- pop-up blocker:
 - turn on
 - turn off
- privacy:
 - accept all cookies
 - block all cookies
 - low
 - high
 - Medium
 - medium high
-

Weighting Schemes

- Specification based
 - Binary choices
 - Set **higher weight** to **ON** option, **lower weight** to **OFF** option
 - **Intuition**: **ON** will cause more code to be executed

Example Of Specification Weighting

Properties:

- pop-up blocker:
 - turn on 0.9 – weight for “on” option
 - turn off 0.1 – weight for “off” option
- privacy:
 - accept all cookies 0.05
 - block all cookies 0.10
 - low 0.20
 - high 0.15
 - Medium 0.25
 - medium high 0.25
-

Weighting Schemes

- **Specification based**
 - Multiple choices
 - More features or higher complexity of the choices as a proxy for higher code coverage, hence higher weight
 - **Intuition**: More features or higher complexity choices will cause more code to be executed

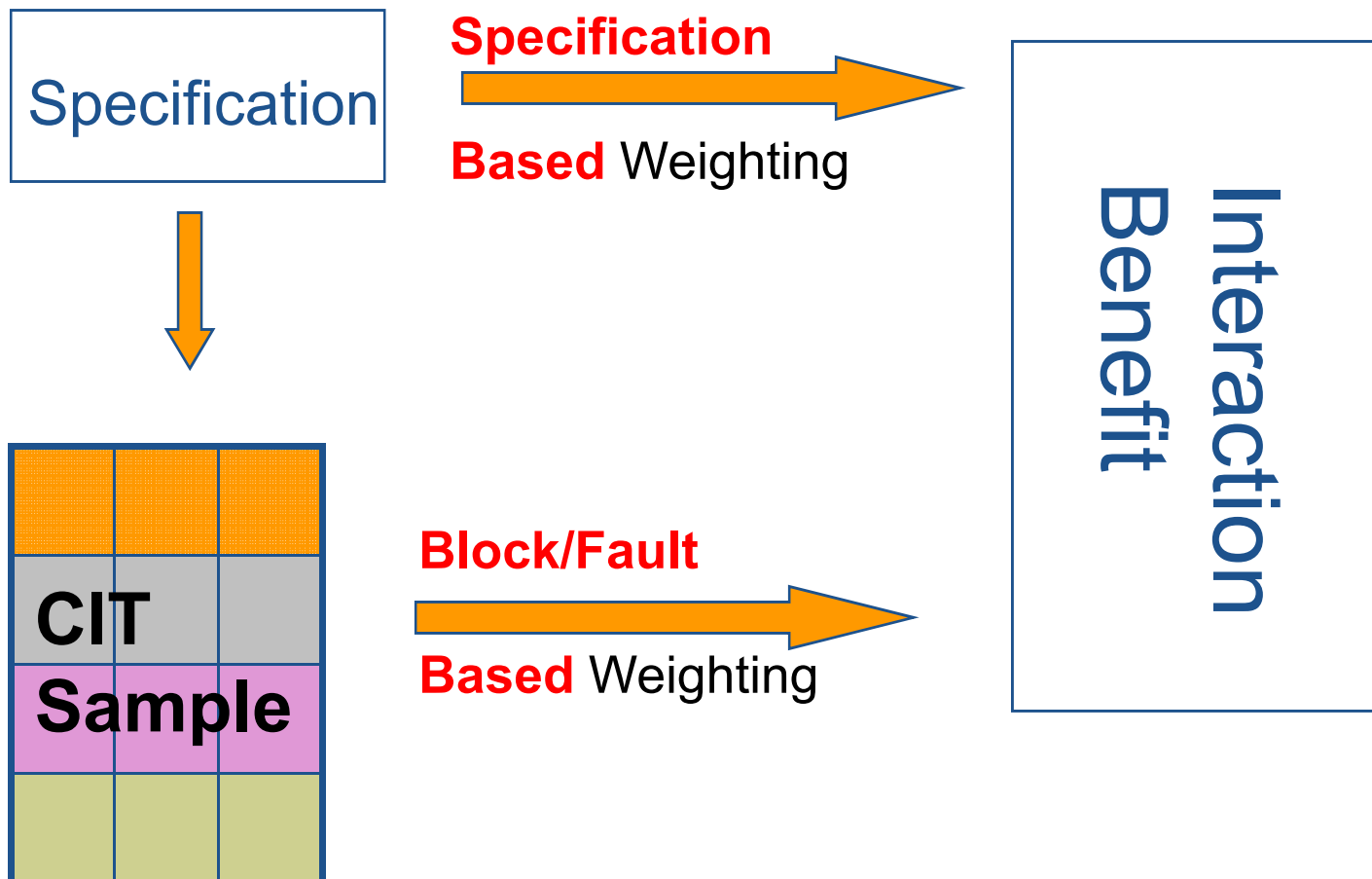
Example of Specification Weighting

Properties:

- pop-up blocker:
 - turn on 0.9
 - turn off 0.1
- privacy:
 - accept all cookies 0.05
 - block all cookies 0.10
 - low 0.20
 - high 0.15
 - medium 0.25
 - medium high 0.25
-

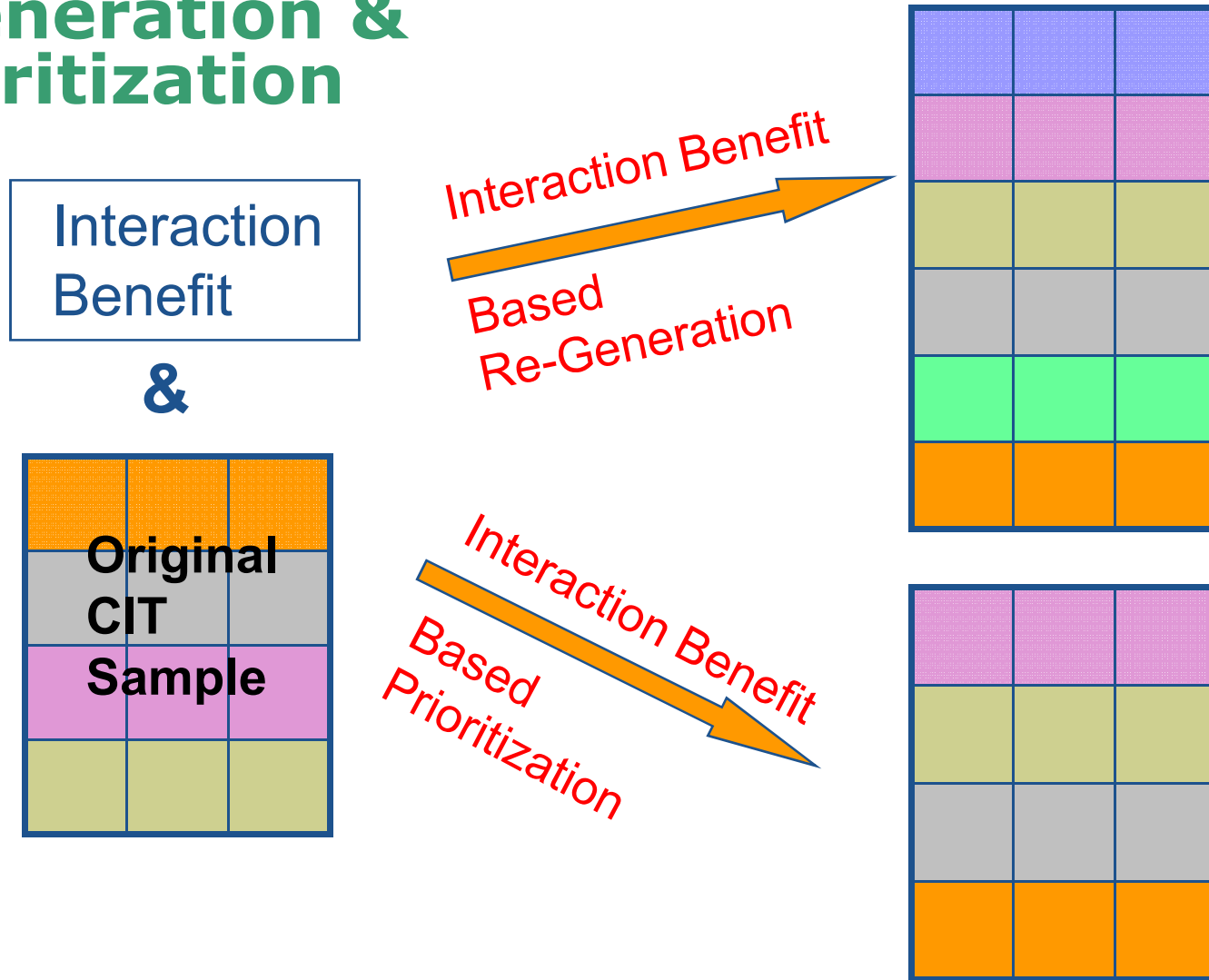
Summary of Techniques

Interaction Benefit



Summary of Techniques

Re-generation & Prioritization



Outline



Motivation

Combinatorial Interaction Testing

Prioritization Techniques

Empirical Study and Results

Conclusions and Future Work

Research Questions

1

Do configurations impact testing?

2

Is CIT effective in sampling configurations in testing?

3

Can we prioritize configurations to improve testing efficiency?

Subject - vim

SIR: Software Infrastructure Repository [1]

- Test suite: size – 975, execution time – 7 hours
- Default configuration

Versions	# blocks	LoC	# Changed Func	Seeded Faults
V2	26,081	87,422	664	13
V3	30,217	105,225	830	13
V4	30,426	105,690	323	14
V5	30,553	106,391	317	14
V6	30,744	107,992	324	12
V7	30,764	105,944	311	13

[1] Do, H., Elbaum, S., and Rothermel, G. Supporting Controlled Experimentation with Testing Techniques: An Infrastructure and its Potential Impact. *Empirical Softw. Eng.* 10, 4 (Oct. 2005), 405-435.

Subject Configurations

Versions	Full Configuration Space	2-way CIT Sample
V2	$2^{77} \times 3^7 \times 4^2 \times 6^3 \times 10^1$	60
V3	$2^{80} \times 3^7 \times 4^2 \times 6^3 \times 10^1$	60
V4	$2^{80} \times 3^7 \times 4^2 \times 6^3 \times 10^1$	60
V5	$2^{65} \times 3^6 \times 4^1 \times 6^3 \times 10^1$	60
V6	$2^{80} \times 3^7 \times 4^2 \times 6^3 \times 10^1$	60
V7	$2^{80} \times 3^7 \times 4^2 \times 6^3 \times 10^1$	60

Subject Configurations

- All configurable options are defined in a `.vimrc` file
- Each time before running test suite, we read the `.vimrc` file to set each option

Subject Configurations

```
set coptions=aABceFs
  set notbs
  set taglength=1
  set notr
  set nosft
  set verbose=14
  set title
  set titlelen=85
set titlestring=VimTitle
  set selectmode=cmd
  set nobk
  set wb
  set patchmode=.org
  set nohid
set maxmemtot=2048
  set maxmem=256
  set warn
  set mod
  set nocf
  set sn
  set ro
  set wa
```

```
  set eb
  set notimeout
  set ttimeout
  set ttimeoutlen=100
  set nsecure
  set eventignore=
set maxfuncdepth=100
  set nolisp
  set nosm
  set wrapmargin=1
  set nocin
  set noai
  set si
  set aw
  set noswf
set swapsync=fsync
set updatecount=0
set updatetime=0
  set eol
  set writedelay=1
  set shiftwidth=8
  set softtabstop=0
```

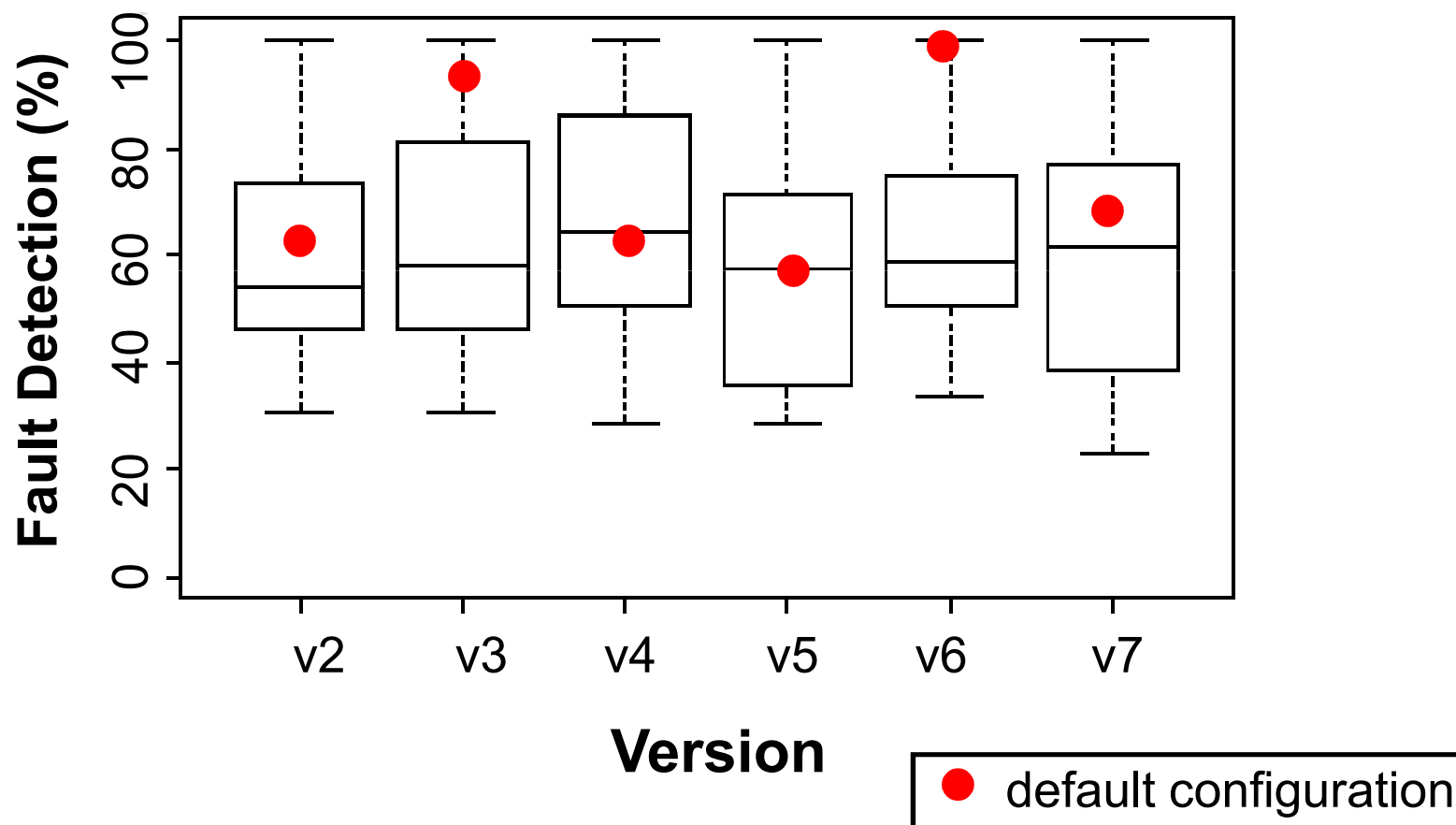
```
  set tabstop=8
  set textwidth=78
  set nolist
  set noet
  set sta
set background=dark
  set helpheight=20
  set more
  set winheight=5
set winminheight=1
  set nonu
  set nowrap
  set sidescroll=1
  set scrolloff=1
  set noea
  set ws
set laststatus=2
  set nolz
  set nolbr
  set ru
  set scroll=0
  set scrolljump=0
```

```
  set whichwrap=
set undolevels=1
  set nojs
  set nosol
  set report=2
  set dictionary=
  set nomagic
  set nosc
  set gd
  set shelltype=0
set cmdheight=2
  set history=0
  set noml
set modelines=5
  set ic
  set noscs
  set is
  set inf
  set top
set maxmapdepth=1000
  set noremap
  ... ..
```

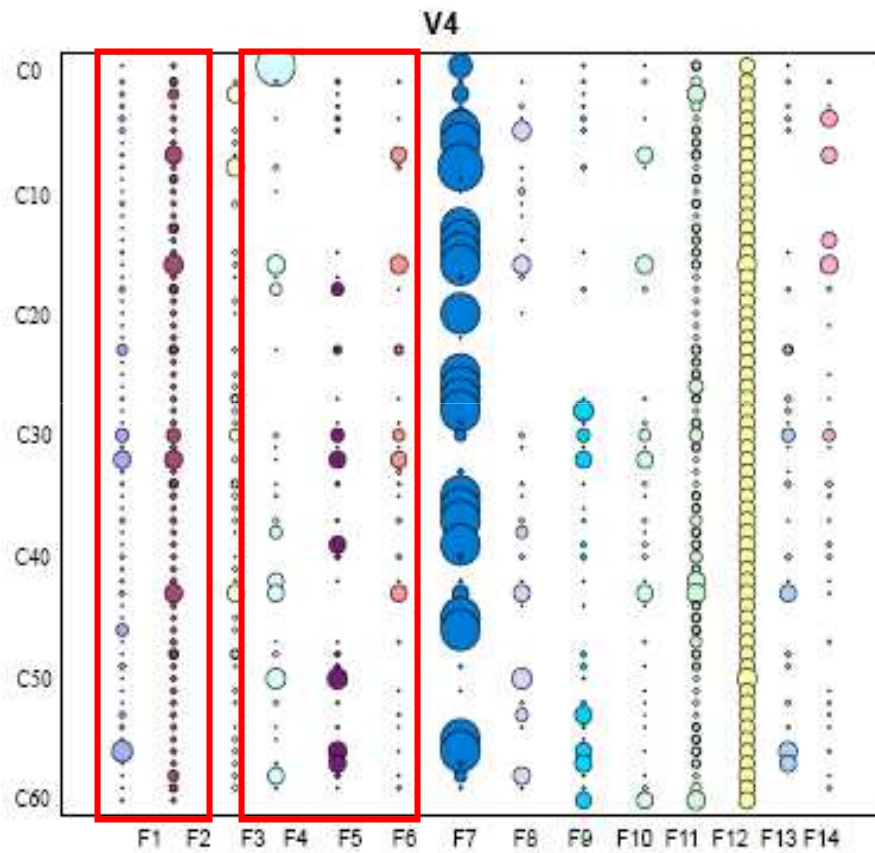
Design of Empirical Study – RQ1 & RQ2

- Independent variables
 - **CIT**: Individual configurations in our CIT samples
 - **Rand**: Random samples generated from same model and with same size
- Dependent variables
 - **BC**: the percentage of **Blocks Covered** by a given test suite
 - **FD**: the percentage of **Faults Detected** by a given test suite

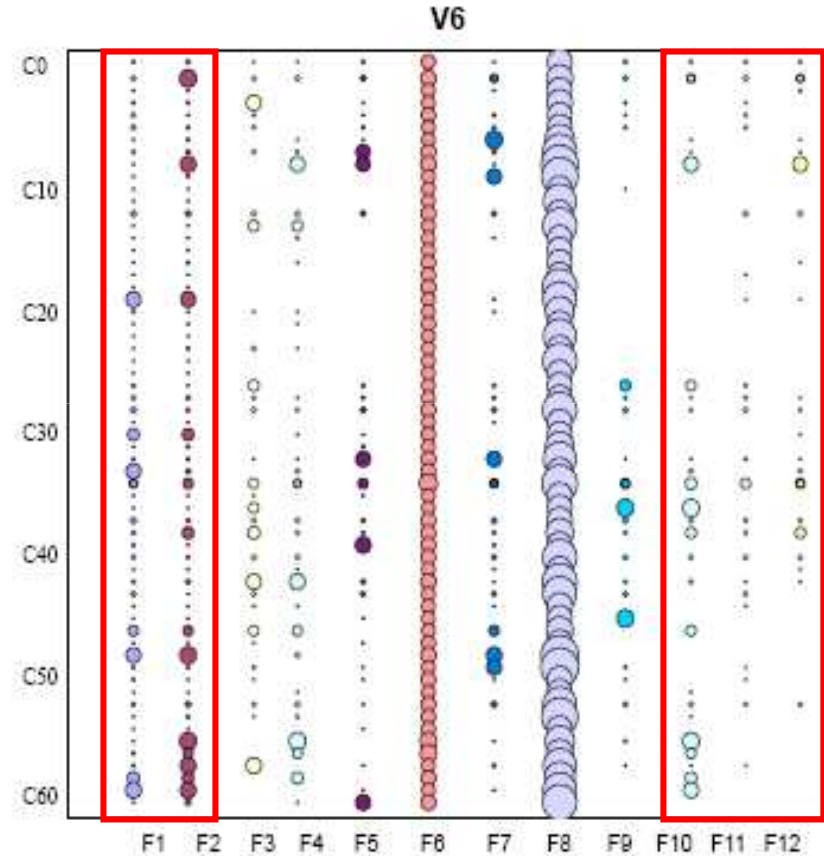
CIT vs. Default Configuration



Fault Detection and Density

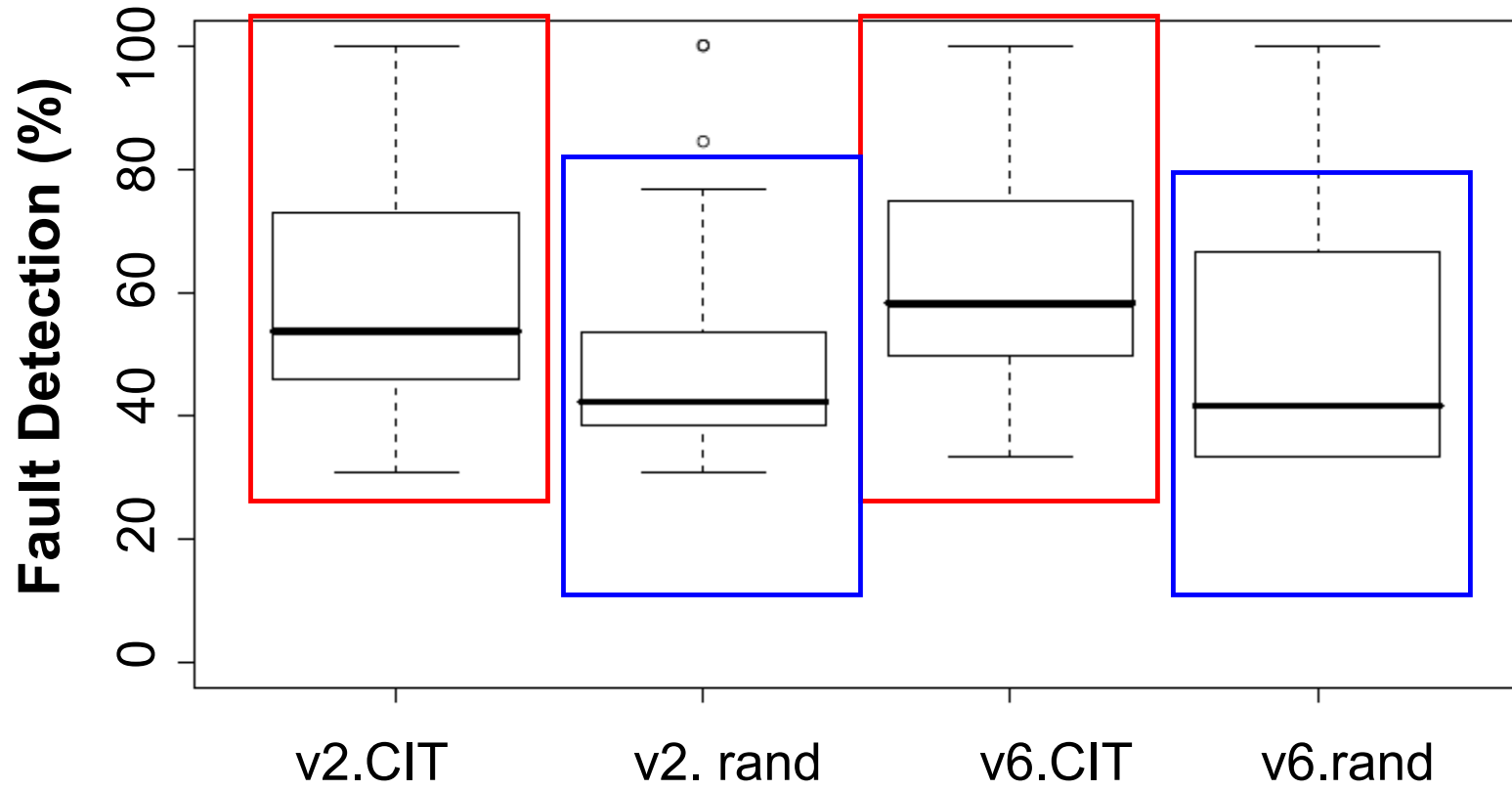


**Configuration –
dependent faults**



**Configuration –
dependent faults**

CIT vs. Rand



Research Questions

1

Do configurations impact testing?

2

Is CIT effective in sampling configurations in testing across multiple versions?

3

Can we prioritize configurations to improve testing efficiency?

Independent Variables

- CIT(unordered CIT sample)
- Pure-BC
- Pure-FD
- Pure-SPEC

Independent Variables

- CIT(unordered CIT sample)
- Pure-BC
- Pure-FD
- Pure-SPEC
- Regen-BC
- Regen-FD
- Regen-SPEC

Independent Variables

- CIT(unordered CIT sample)
- Pure-BC (pure prioritization by block coverage)
- Pure-FD
- Pure-SPEC
- Regen-BC
- Regen-FD
- Regen-SPEC

Independent Variables

- CIT(unordered CIT sample)
- Pure-BC (pure prioritization by block coverage)
- Pure-FD (pure prioritization by fault detection)
- Pure-SPEC
- Regen-BC
- Regen-FD
- Regen-SPEC

Independent Variables

- CIT(unordered CIT sample)
- Pure-BC (pure prioritization by block coverage)
- Pure-FD (pure prioritization by fault detection)
- Pure-SPEC (pure prioritization by specification)
- Regen-BC
- Regen-FD
- Regen-SPEC

Independent Variables

- CIT(unordered CIT sample)
- **Pure-BC** (pure prioritization by block coverage)
- **Pure-FD** (pure prioritization by fault detection)
- **Pure-SPEC** (pure prioritization by specification)
- **Regen-BC** (regeneration by block coverage)
- **Regen-FD** (regeneration by fault detection)
- **Regen-SPEC** (regeneration by specification)

Dependent Variable

- NAPFD: Normalized Average Percentage of Faults Detected
 - Measures how fast the faults are detected
 - Value is between 0 and 1
 - The higher, the better

Study Methodology – RQ3

- For all re-generation and prioritization, we use version v_i to prioritize v_{i+1}
- For unordered CIT sample, we use 50 random orders and use the average
- Examine various testing **budgets**: number of configurations we are able to run due to time/resource limitation:

5, 10, ..., 60

Results

version	unordered	pure-FD	p-BC	p-SPEC	regen-FD	r-BC	r-SPEC
NAPFD value (budget: 5/60)							
v3	0.76	0.78	0.78	0.81	0.90	0.90	0.90
v4	0.76	0.90	0.90	0.90	0.90	0.90	0.90
v5	0.74	0.71	0.69	0.68	0.90	0.90	0.90
v6	0.73	0.90	0.90	0.90	0.90	0.90	0.90
v7	0.76	0.90	0.73	0.90	0.90	0.90	0.90
NAPFD value (budget: 10/60)							
... ..							
NAPFD value (budget: 15/60)							
v3	0.92	0.93	0.93	0.94	0.97	0.97	0.97
v4	0.92	0.97	0.97	0.97	0.97	0.97	0.97
v5	0.91	0.87	0.86	0.86	0.97	0.97	0.97
v6	0.91	0.97	0.97	0.97	0.97	0.97	0.97
v7	0.92	0.97	0.91	0.97	0.97	0.97	0.97
NAPFD value (10^{-2}) (budget: 60/60)							
... ..							

Results

version	unordered	Pure			Re-generation		
		FD	BC	SPEC	FD	BC	SPEC
NAPFD value (budget: 5/60)							
v3	0.76	0.78	0.78	0.81	0.90	0.90	0.90

Results

version	unordered	pure-FD	p-BC	p-SPEC	regen-FD	r-BC	r-SPEC
NAPFD value (budget: 5/60)							
v3	0.76	0.78	0.78	0.81	0.90	0.90	0.90
v4	0.76	0.90	0.90	0.90	0.90	0.90	0.90
v5	0.74	0.71	0.69	0.68	0.90	0.90	0.90
v6	0.73	0.90	0.90	0.90	0.90	0.90	0.90
v7	0.76	0.90	0.73	0.90	0.90	0.90	0.90
NAPFD value (budget: 10/60)							
... ..							
NAPFD value (budget: 15/60)							
v3	0.92	0.93	0.93	0.94	0.97	0.97	0.97
v4	0.92	0.97	0.97	0.97	0.97	0.97	0.97
v5	0.91	0.87	0.86	0.86	0.97	0.97	0.97
v6	0.91	0.97	0.97	0.97	0.97	0.97	0.97
v7	0.92	0.97	0.91	0.97	0.97	0.97	0.97
NAPFD value (10^{-2}) (budget: 60/60)							
... ..							

Results

NAPFD value (budget: 15/60)							
version	unordered	Pure			Re-generation		
		FD	BC	SPEC	FD	BC	SPEC
v3	0.92	0.93	0.93	0.94	0.97	0.97	0.97
v4	0.92	0.97	0.97	0.97	0.97	0.97	0.97
v5	0.91	0.87	0.86	0.86	0.97	0.97	0.97
v6	0.91	0.97	0.97	0.97	0.97	0.97	0.97
v7	0.92	0.97	0.91	0.97	0.97	0.97	0.97
NAPFD value (10^{-2}) (budget: 60/60)							
... ..							

Results

		Pure			Re-generation		
version	unordered	FD	BC	SPEC	FD	BC	SPEC
NAPFD value (budget: 5/60)							
v3	0.76	0.78	0.78	0.81	0.90	0.90	0.90
v4	0.76	0.90	0.90	0.90	0.90	0.90	0.90
v5	0.74	0.71	0.69	0.68	0.90	0.90	0.90
v6	0.73	0.90	0.90	0.90	0.90	0.90	0.90
v7	0.76	0.90	0.73	0.90	0.90	0.90	0.90

Outline

Motivation

Combinatorial Interaction Testing

Prioritization Techniques

Empirical Study and Results

Conclusions and Future Work

Conclusions

- We validated that configurations play an important role in fault detection in our subject
- We validated that CIT is effective for sampling the configuration space across multiple versions
- We provided various methods to prioritize configurations, including a specification-based method that does not require information from previous versions
- Our results show that we are able to prioritize CIT configurations to improve early fault detection

Conclusions

- We validated that configurations play an important role in fault detection in our subject
- We validated that CIT is effective for sampling the configuration space across multiple versions
- We provided various methods to prioritize configurations, including a specification-based method that does not require information from previous versions
- Our results show that we are able to prioritize CIT configurations to improve early fault detection

Conclusions

- We validated that configurations play an important role in fault detection in our subject
- We validated that CIT is effective for sampling the configuration space across multiple versions
- We provided various methods to prioritize configurations, including a specification-based method that does not require information from previous versions
- Our results show that we are able to prioritize CIT configurations to improve early fault detection

Conclusions

- We validated that configurations play an important role in fault detection in our subject
- We validated that CIT is effective for sampling the configuration space across multiple versions
- We provided various methods to prioritize configurations, including a specification-based method that does not require information from previous versions
- Our results show that we are able to prioritize CIT configurations to improve early fault detection

Future Work

- Other subjects
- More CIT samples (only one in our subject)
- Additional heuristics to be used for prioritization
- Consider CIT testing (prioritization) applied to both configurations and tests combined



Thank You !





Related Work

Regression Testing Prioritization

[1] Z. Li, M. Harman, and R. M. Hierons. Search algorithms for regression test case prioritization. IEEE SE 07

- [APFD Metric](#)

[2] S. Elbaum, A. G. Malishevsky, and G. Rothermel. Test case prioritization: A family of empirical studies. IEEE SE 02

[3] K. R. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos. Time-aware test suite prioritization. ISSTA 06

[4] S. Elbaum, A. Malishevsky, and G. Rothermel. Incorporating varying test costs and fault severities into test case prioritization. ICSE 01

- [NAPFD Metric](#)

[5] X. Qu, M. B. Cohen, and K. M. Woolf. Combinatorial interaction regression testing: A study of test case generation and prioritization, ICSM 08

CIT In Testing

- [1] D. M. Cohen, S. R. Dalal, M. L. Fredman, and G. C. Patton. The AETG system: an approach to testing based on combinatorial design. IEEE SE 97
- [2] M. B. Cohen, C. J. Colbourn, P. B. Gibbons, and W. B. Mugridge. Constructing test suites for interaction testing. ICSE 03
- [3] D. Kuhn and M. Reilly. An investigation of the applicability of design of experiments to software testing. NASA/IEEE SE Workshop 02
- [4] R. Brownlie, J. Prowse, and M. S. Phadke. Robust testing of AT&T PMX/StarMAIL using OATS. AT& T Technical Journal 92
- [CIT In Configuration Sampling](#)
- [5] C. Yilmaz, M. B. Cohen, and A. Porter. Covering arrays for efficient fault characterization in complex configuration spaces. IEEE SE 06
- [6] M. B. Cohen, J. Snyder, and G. Rothermel. Testing across configurations: implications for combinatorial testing. Intl. Workshop on advances in MBT 06

CIT Prioritization

- Re-generation

[1] R. Bryce and C. Colbourn. Prioritized interaction testing for pair-wise coverage with seeding and constraints. Journal of Information and Software Technology 06

- Coverage In Interaction

[2] A. Avritzer and E. R. Weyuker. The automatic generation of load test suites and the assessment of the resulting software. IEEE SE 95

- Pure Prioritization And Empirical Study

[3] X. Qu, M. B. Cohen, and K. M. Woolf. Combinatorial interaction regression testing: A study of test case generation and prioritization. ICSM 07