

A CP approach of the variability testing of software product lines

Arnaud Gotlieb
Certus Centre

Simula Research Laboratory
Norway



The Certus Centre

Software Validation and Verification

Hosted by SIMULA

Established and awarded SFI in Oct. 2011

duration: 8 years

www.certus-sfi.no



Cisco Systems Norway



Norwegian Custom and excise



ABB Robotics Stavanger

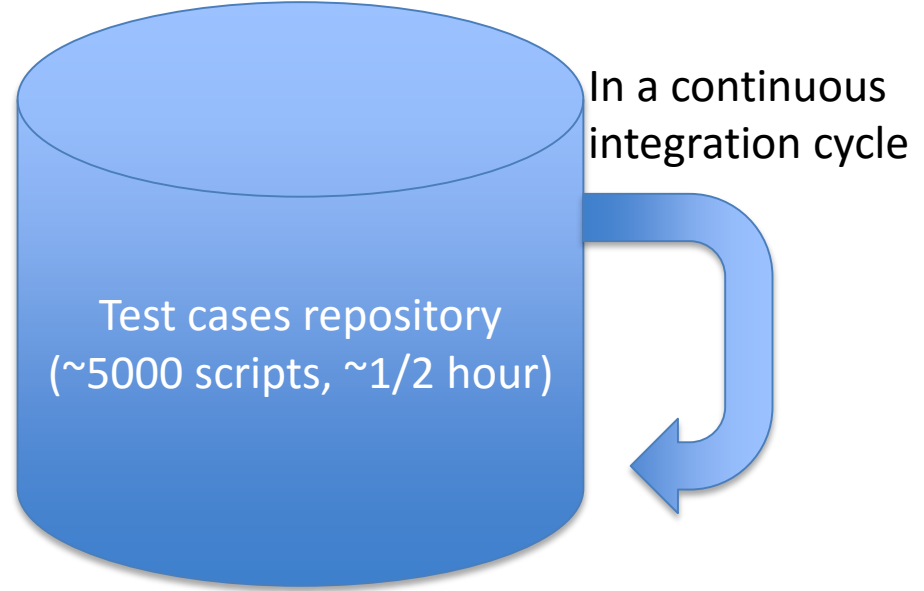


Kongsberg Maritime

[**simula** . research laboratory]
- by thinking constantly about it



The Validation of CISCO's Video Conferencing Product Line



Multisite features

- 4-way 1080p30 High Definition SIP/H.323 MultiSite
- Full individual audio and video transcoding
- Individual layouts in MultiSite CP (takes out self view)
- H.323/SIP/VoIP in the same conference
- Support for Presentation (H.239/BFCP) from any party
- Best Impression (Automatic CP Layouts)
- H.264, Encryption, Dual Stream from any site
- IP Downspeeding
- Dial in/Dial out
- Additional telephone call (no license required)
- Conference rates up to 10 Mbps

Audio features

- CD-Quality 20KHz Mono and Stereo
- Eight separate acoustic echo cancellers
- 8-port Audio mixer
- Automatic Gain Control (AGC)
- Automatic Noise Reduction
- Active lip synchronization

Audio standards

- G.711, G.722, G.722.1, 64 kbps & 128 kbps MPEG4 AAC-LD, AAC-LD Stereo

IP network features

- DNS lookup for service configuration
- Differentiated Services (DoS)
- IP adaptive bandwidth management (in)
- Auto gatekeeper discovery
- Dynamic playout and lip-sync buffering
- H.245 DTMF tones in H.323
- Date and Time support via NTP
- Packet Loss based Downspeeding
- URI Dialing
- TCP/IP
- DHCP
- 802.1x Network authentication
- ClearPath

Live video res.

- 176 x 144@30 fps (QCIF)
- 352 x 288@30 fps (CIF)
- 512 x 288@30 fps (w288p)
- 576 x 448@30 fps (448p)
- 768 x 448@30 fps (w448p)
- 704 x 576@30 fps (w576p)
- 1024 x 576@30 fps (w576p)
- 1280 x 720@30 fps (720p30)
- 1920 x 1080@30 fps (1080p30)
- 640 x 480@30 fps (VGA)
- 800 x 600@30 fps (SVGA)
- 1024 x 768@30 fps (XGA)
- 1280 x 1024@30 fps (SXGA)
- 1280 x 768@30 fps (WXGA)
- 1440 x 900@30 fps (WXGA+)
- 1680 x 1050@30 fps (WUXGA+)
- 1600 x 1200@30 fps (LUXGA)
- 1920 x 1200@25 fps (WUXGA)
- 512 x 288@60 fps (w288p60)
- 768 x 448@60 fps (w448p60)
- 1024 x 576@60 fps (w576p60)
- 1280 x 720@60 fps (720p60)
- 720p30 from 768kpbs
- 720p60 from 1152kpbs
- 1080p30 from 1472kpbs

1. Test case selection
2. Test suite optimization
3. Test execution scheduling

Video features

- Native 16:9 Widescreen
- Advanced Screen Layouts
- Intelligent Video Management
- Local Auto Layout
- 9 embedded individual video compositors

Security features

- Management via HTTPS and SSH
- IP Administration Password
- Menu Administration Password
- Disable IP services
- Network Settings protection

Bandwidth

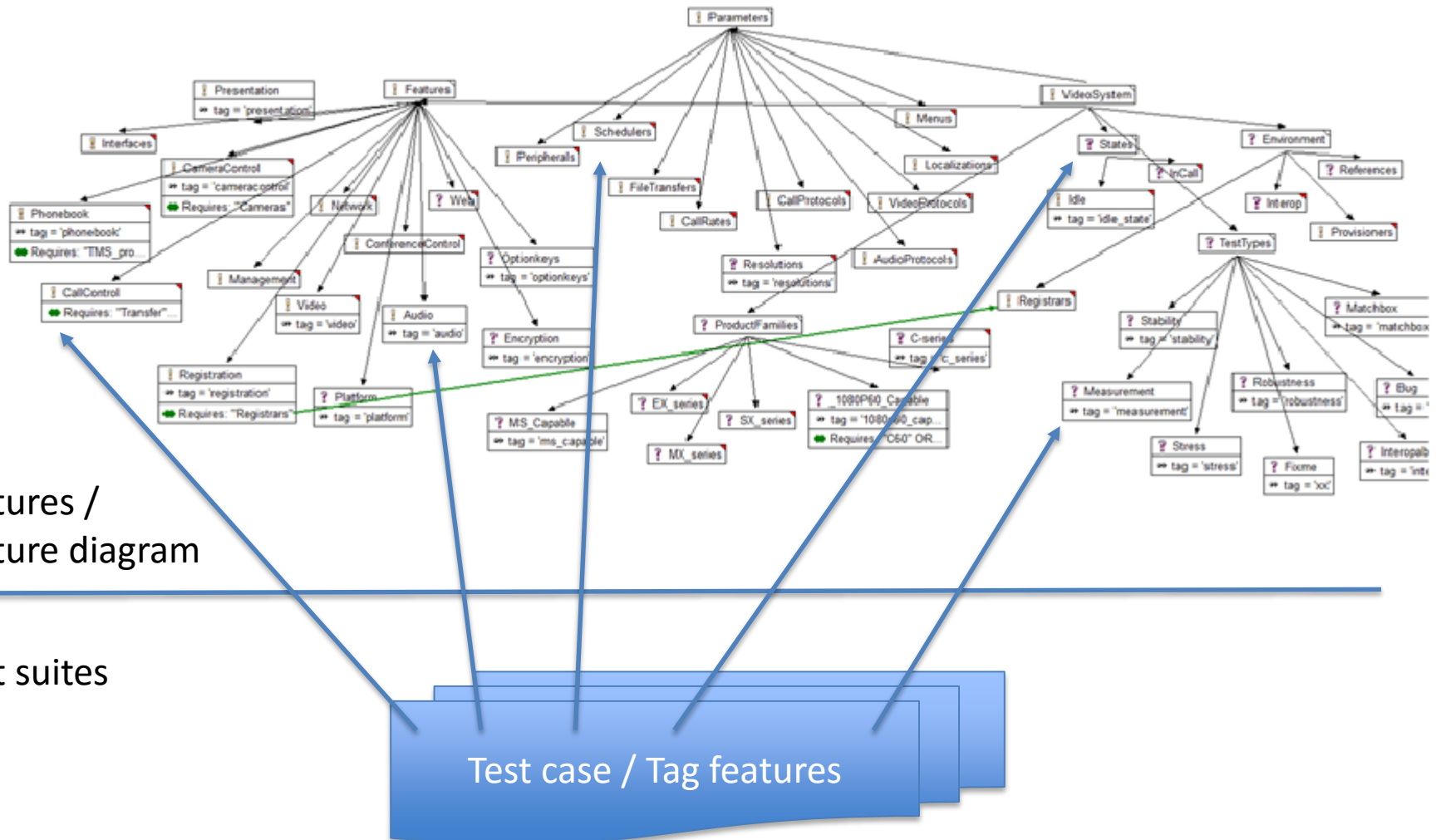
- H.323/SIP up to 6 Mbps point-to-point
- Up to 10 Mbps total MultiSite bandwidth

Protocols

- H.323
- SIP



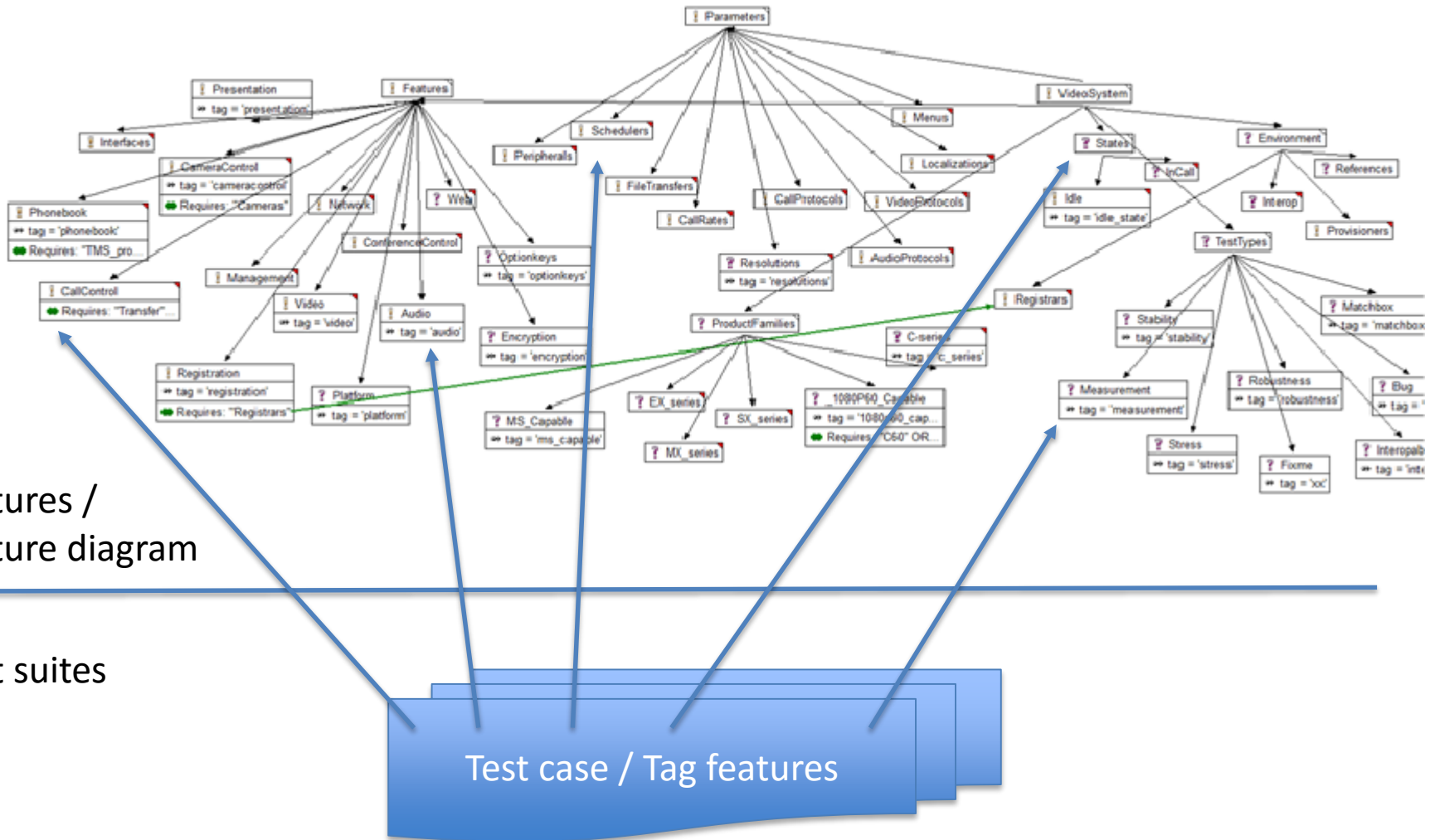
Test case selection based on feature modelling



S. Wang, S. Ali, A. Gotlieb, and M. Liaen. A systematic test case selection methodology for product lines: results and insights from an industrial case study. *Empirical Software Engineering*, pages 1–37, 2014.

S. Wang, A. Gotlieb, S. Ali, and M. Liaen. Automated test case selection using feature model: An industrial case study. In *ACM/IEEE 16th Int. Conf. on Model Driven Eng. Languages and Systems (MODELS'13)*, Awarded best application paper, Miami, FL, USA, Sep. 2013.

Test suite optimization



How to select a test set which cover all the features in an acceptable amount of time (i.e., cost-effective optimization) ?

Agenda

I. Introduction



II. Optimal Test Suite Reduction

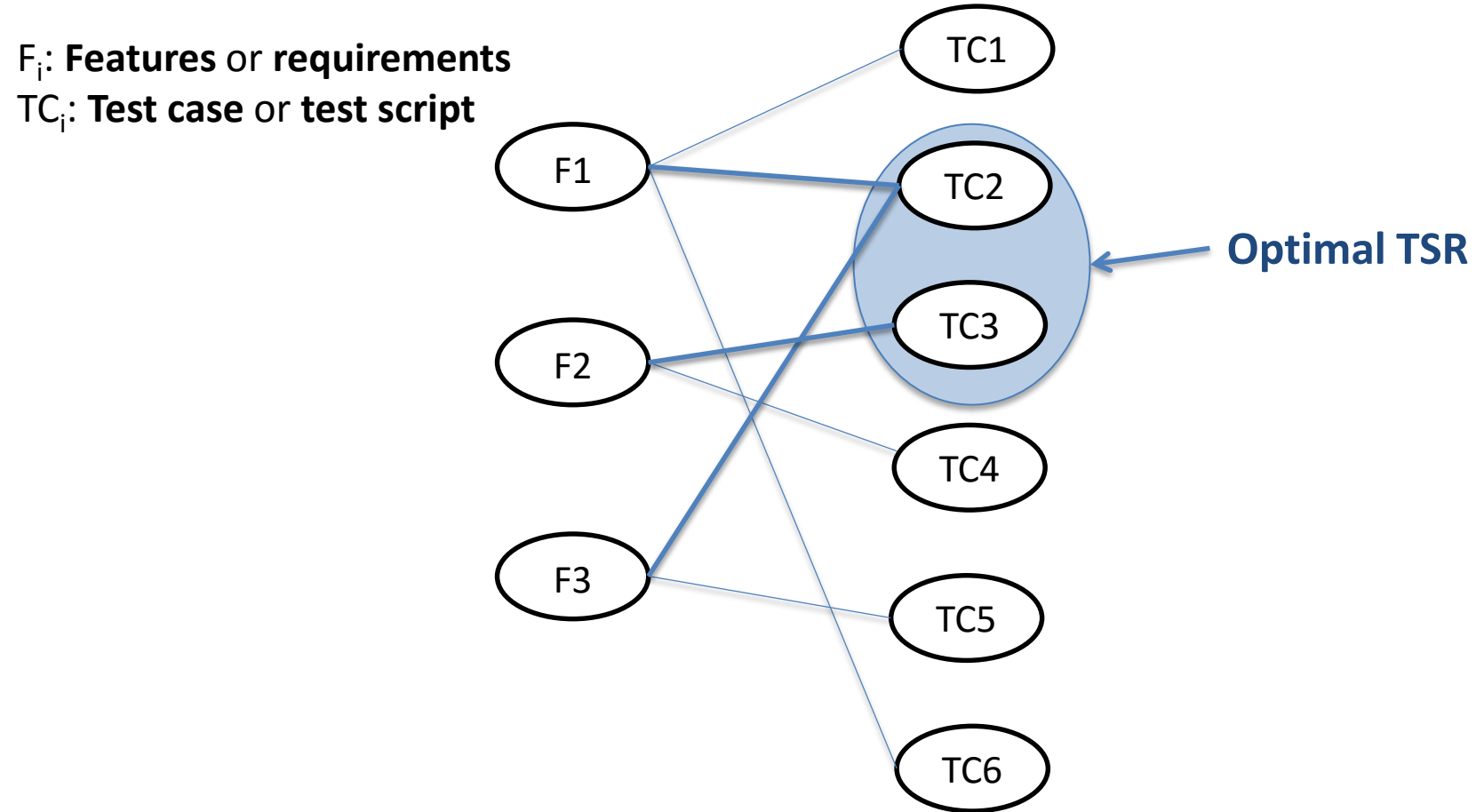
III. Multi-objectives Test Suite Reduction

IV. Industrial Application

V. Conclusions and Perspectives

Optimal Test Suite Reduction

Optimal TSR: the core problem



Optimal TSR: find a minimal subset of TC such that each F is covered at least once (Practical importance but NP-hard problem!) – An instance of *Minimum Set Cover*

Constraint Programming

Declarative programming paradigm where relations are modeled with variables, finite and continuous domains, and constraints

- e.g., arithmetical, $X \text{ in } 13..59, Y \text{ in } 4..9, Y > 6*Y - X$
- e.g., symbolic (terms, string..) $t(X, r(3, Y)) = t(p(4), Z)$
- e.g., numerical $X = \sin(Y), X + Y = 0.567898$

Global constraint: A constraint which captures a relation over a non-fixed number of variables and implements a dedicated filtering algorithm

The nvalue global constraint

$$nvalue(n, v)$$

Where:

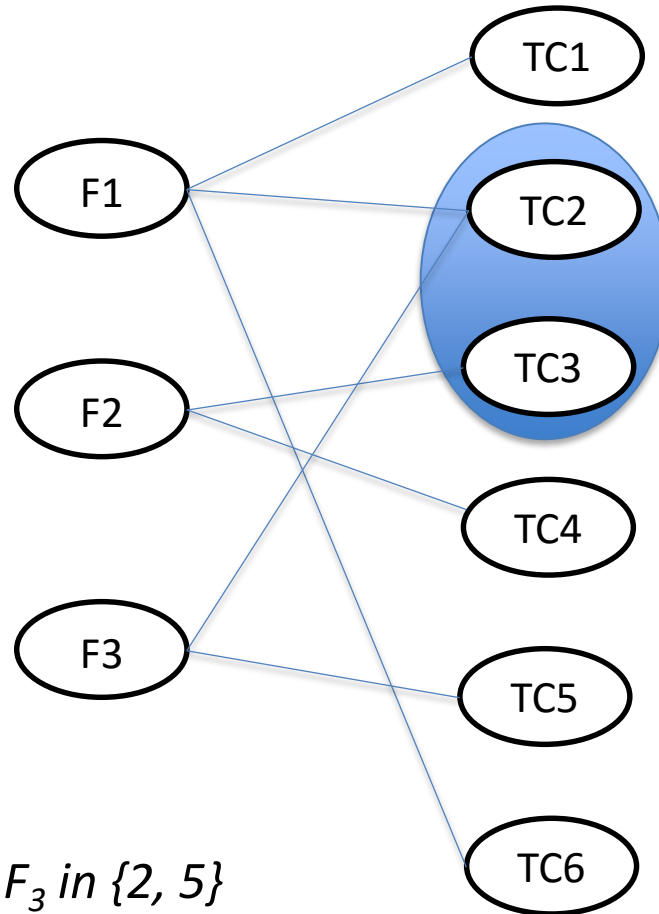
n is an FD_variable

$v = (v_1, \dots, v_k)$ is a vector of FD_variables

$$nvalue(n, v) \text{ holds iff } n = card(\{v_i\}_{i \text{ in } 1..k})$$

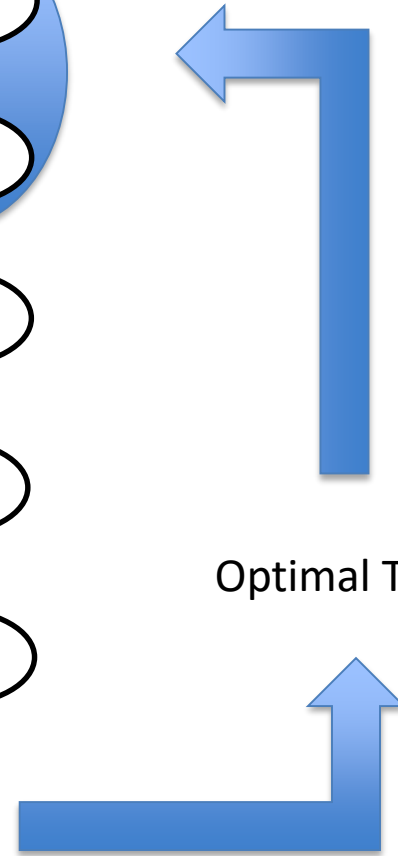
Introduced in [Pachet and Roy'99], first filtering algorithm in [Beldiceanu'01]
Solution existence for nvalue is NP-hard [Bessiere et al. '04]

Optimal TSR: CP model with nvalue (1)



F_1 in {1, 2, 6}, F_2 in {3, 4}, F_3 in {2, 5}
nvalue(MaxNvalue, (F₁, F₂, F₃)),
label(minimize(MaxNvalue))

Optimal TSR



The global_cardinality constraint

$$gcc(t, d, v)$$

Where

$t = (t_1, \dots, t_N)$ is a vector of N variables, each t_j in $Min_j..Max_j$

$d = (d_1, \dots, d_k)$ is a vector of k values

$v = (v_1, \dots, v_k)$ is a vector of k variables, each v_i in $Min_i..Max_i$

$$gcc(t, d, v) \text{ holds iff } \forall i \text{ in } 1..k, \\ v_i = \text{card}(\{t_j = d_i\}_{j \text{ in } 1..N})$$

Filtering algorithms for gcc are based on max flow computations in a network flow [Regin AAAI'96]

Example

$\text{gcc}((F_1, F_2, F_3), (1, 2, 3, 4, 5, 6), (V_1, V_2, V_3, V_4, V_5, V_6))$

means that:

In a solution of TSR

TC_1 covers exactly V_1 requirements in (F_1, F_2, F_3)

TC_2 " V_2 "

TC_3 " V_3 "

...

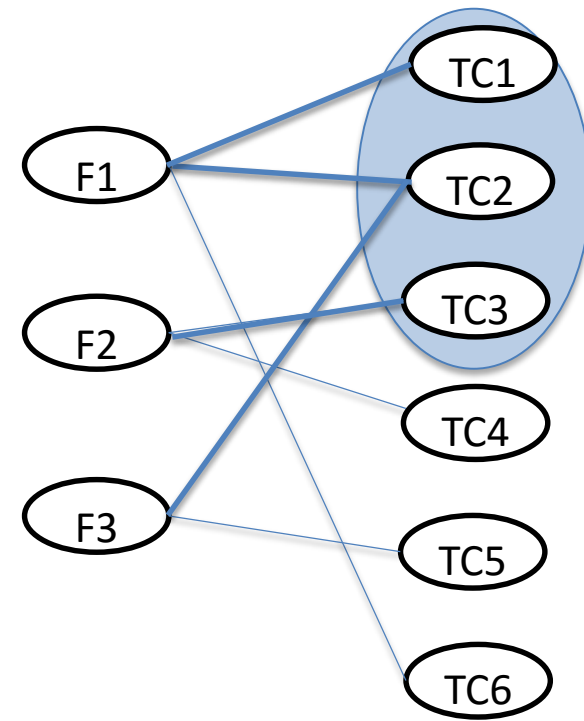
Where $F_1, F_2, F_3, V_1, V_2, V_3, \dots$ denote finite-domain variables

F_1 in $\{1, 2, 6\}$, F_2 in $\{3, 4\}$, F_3 in $\{2, 5\}$

V_1 in $\{0, 1\}$, V_2 in $\{0, 2\}$, V_3 in $\{0, 1\}$, V_4 in $\{0, 1\}$, V_5 in $\{0, 1\}$, V_6 in $\{0, 1\}$

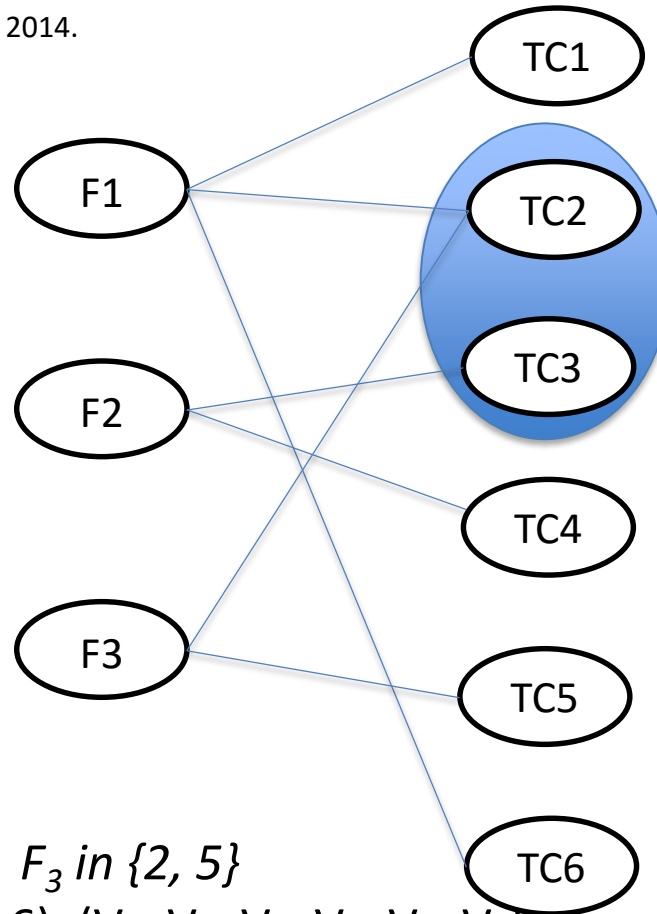
Here, for example, $V_1 = 1, V_2 = 2, V_3 = 1, V_4 = 0, V_5 = 0, V_6 = 0$ is a feasible solution

But, not an optimal one!



Optimal TSR: CP model with two gcc (2)

A. Gotlieb and D. Marijan. Flower: Optimal test suite reduction as a network maximum flow. In *Proc. of Int. Symp. on Soft. Testing and Analysis (ISSTA'14)*, San José, CA, USA, Jul. 2014.



F_1 in $\{1, 2, 6\}$, F_2 in $\{3, 4\}$, F_3 in $\{2, 5\}$

`gcc((F1, F2, F3), (1,2,3,4,5,6), (V1, V2, V3, V4, V5, V6)),`

`gcc((V1, V2, V3, V4, V5, V6), (0-_), (Max0Req-_)),`

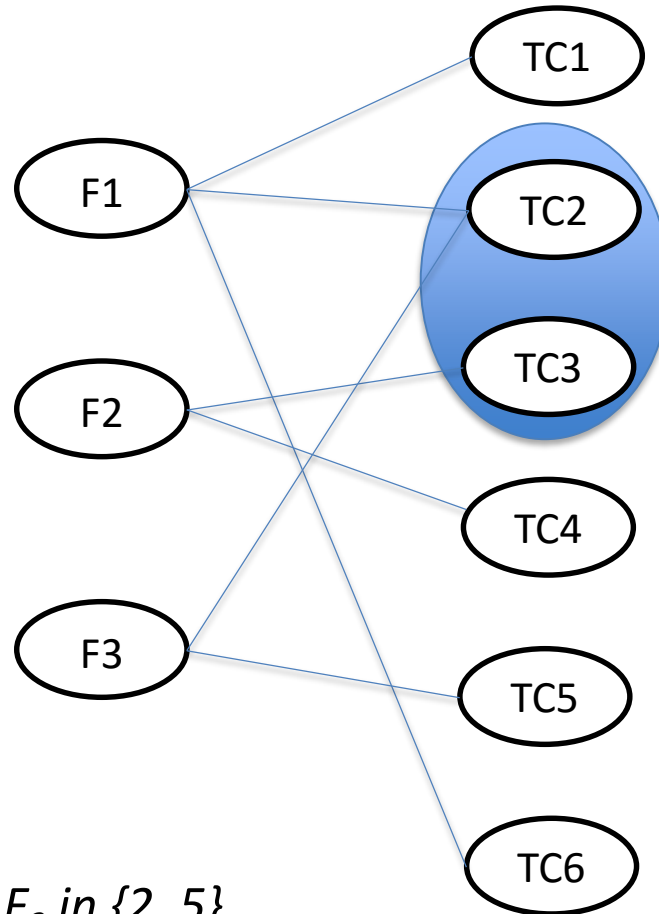
`label(maximize(Max0Req))`

Optimal TSR

/ search heuristics by enumerating the Vi first */*

3. Optimal TSR: CP model Mixt (3)

A. Gotlieb M. Carlsson M. Liaen D. Marijan A. Petillon. *Automated Regression Testing using CP*. Under submission 2015



F_1 in {1, 2, 6}, F_2 in {3, 4}, F_3 in {2, 5}

```
gcc( (F1, F2, F3), (1,2,3,4,5,6), (V1, V2, V3, V4, V5, V6) ),
```

```
nvalue(MaxNvalue, (F1, F2, F3),
```

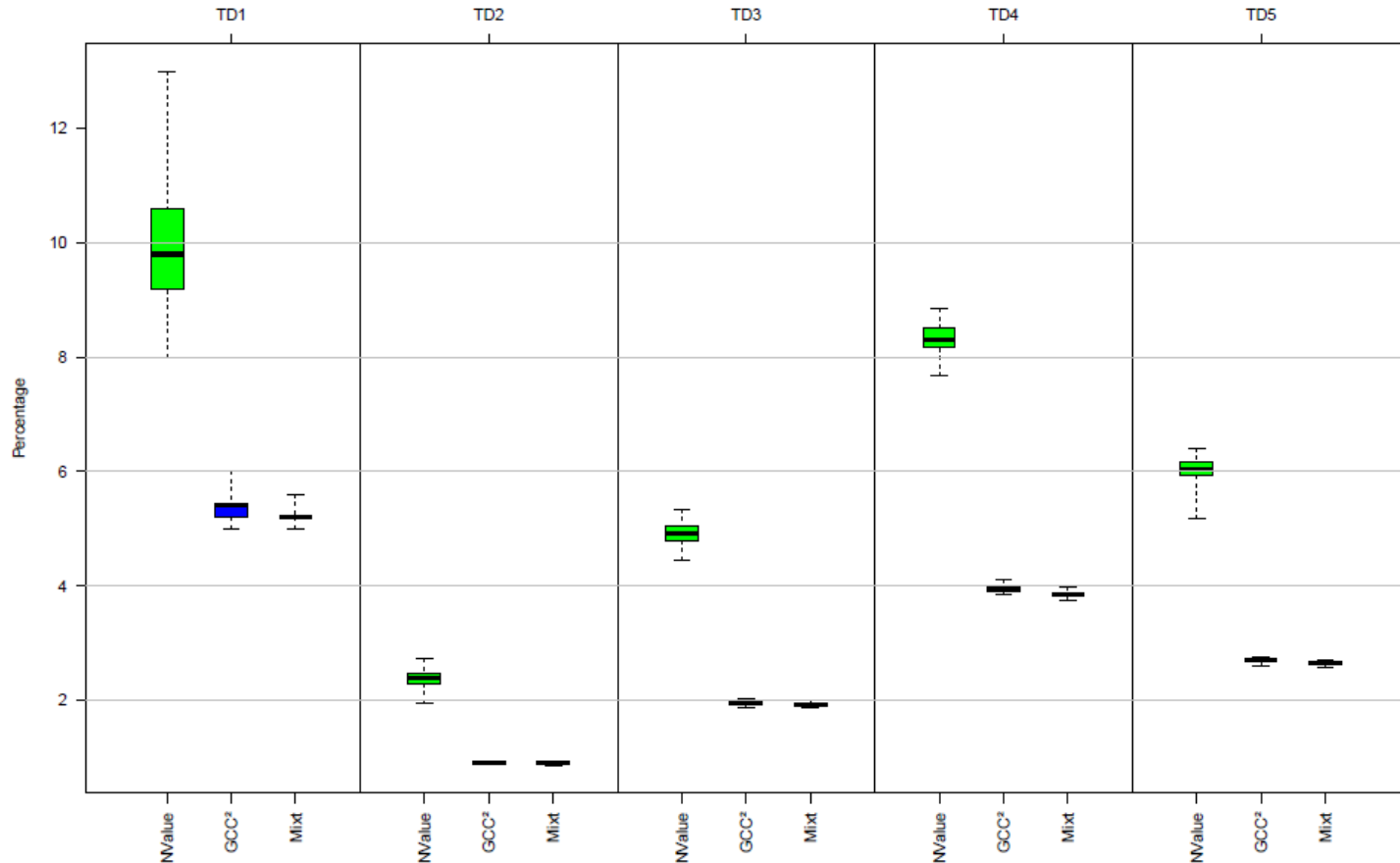
```
label(minimize(MaxNvalue))
```

/* + presolve + labelling heuristics based on max */

Optimal TSR

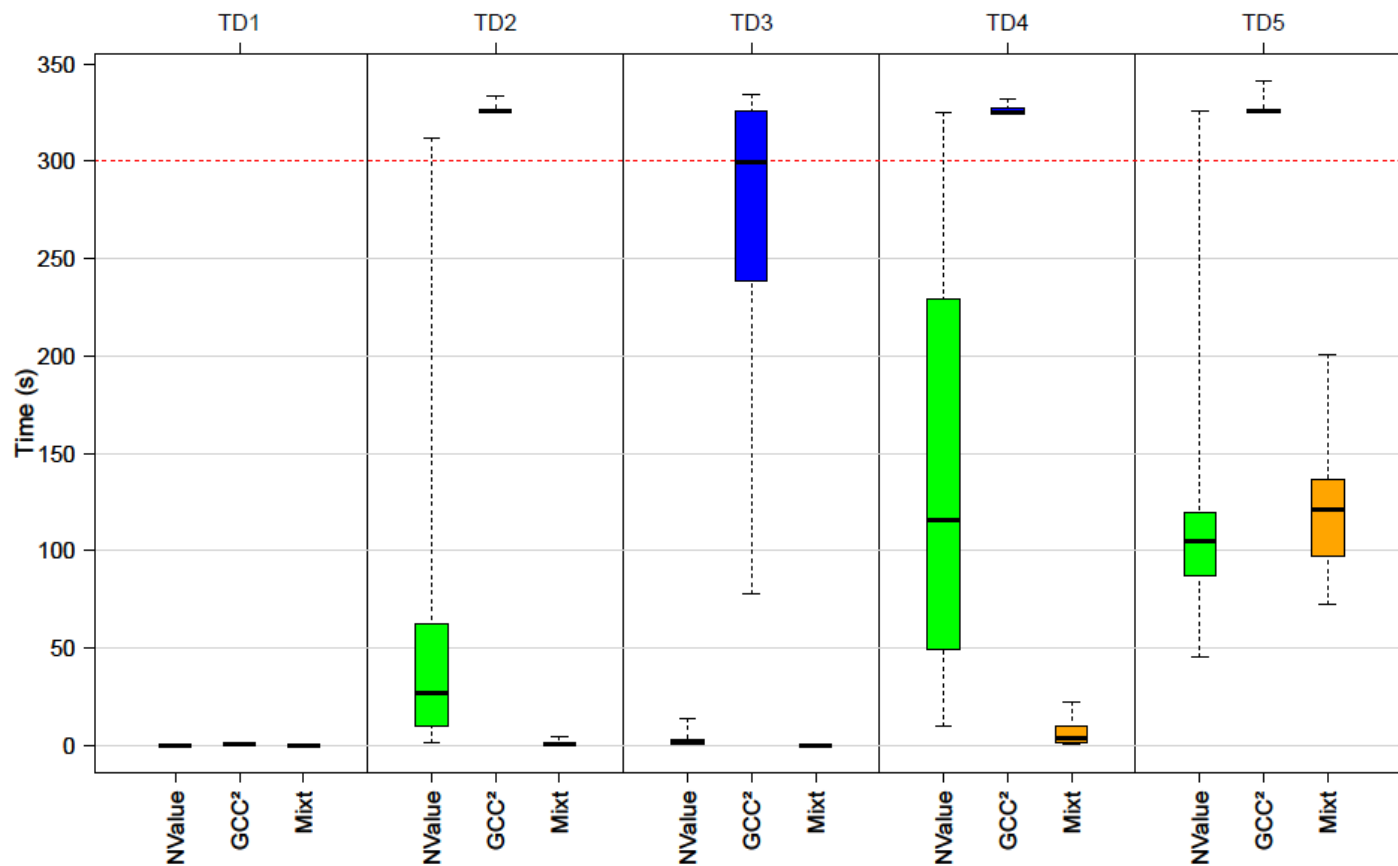
Model comparison on random instances

(Reduced Test Suite percentage in 30sec of search)



	TD1	TD2	TD3	TD4	TD5
Requirements	250	500	1000	1000	1000
Test cases	500	5000	5000	5000	7000
Density	20	20	20	8	8

Model comparison on random instances (CPU time to find a global optimum)



	TD1	TD2	TD3	TD4	TD5
Requirements	20	90	60	60	30
Test cases	70	100	100	200	500
Density	8	20	20	20	8

Optimal TSR: existing approaches

- Exact method: ILP formulation [Hsu Orso ICSE 2009] –
MINTS/CPLEX, MINTS/MiniSAT

Minimize $\sum_{i=1..6} x_i$ (minimize the number of test cases)

subject to $\left\{ \begin{array}{l} x_1 + x_2 + x_6 \geq 1 \\ x_3 + x_4 \geq 1 \\ x_2 + x_5 \geq 1 \end{array} \right\}$ (cover every req. at least once)

- Approximation algorithms (greedy) –

R = Set of reqs, Current = \emptyset

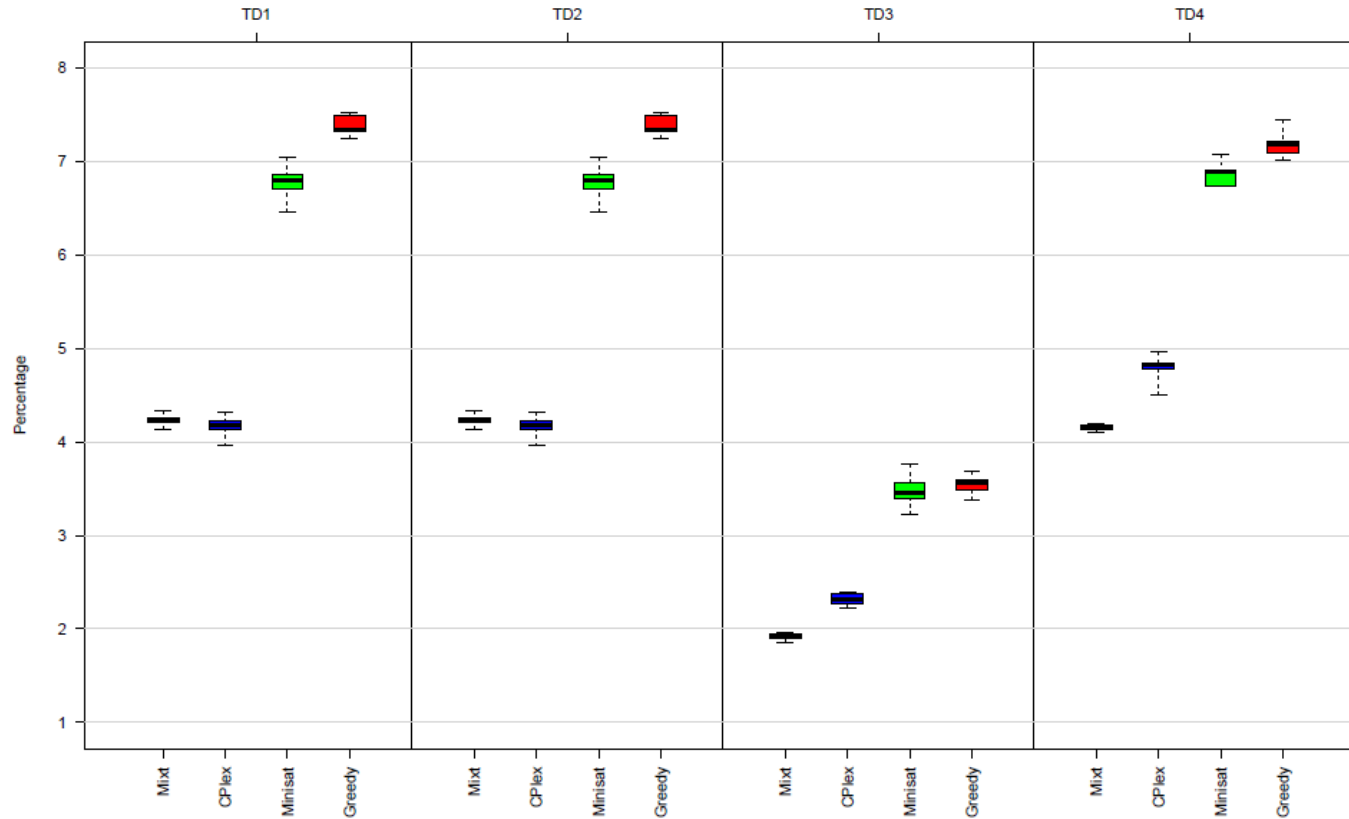
while(Current \neq R)

 Select a test case that covers the most uncovered reqs ;

 Add covered reqs to Current ;

return Current

Comparison with other approaches (Reduced Test Suite percentage in 60 sec)



	TD1	TD2	TD3	TD4
Requirements	1000	1000	1000	2000
Test cases	5000	5000	5000	5000
Density	7	7	20	20

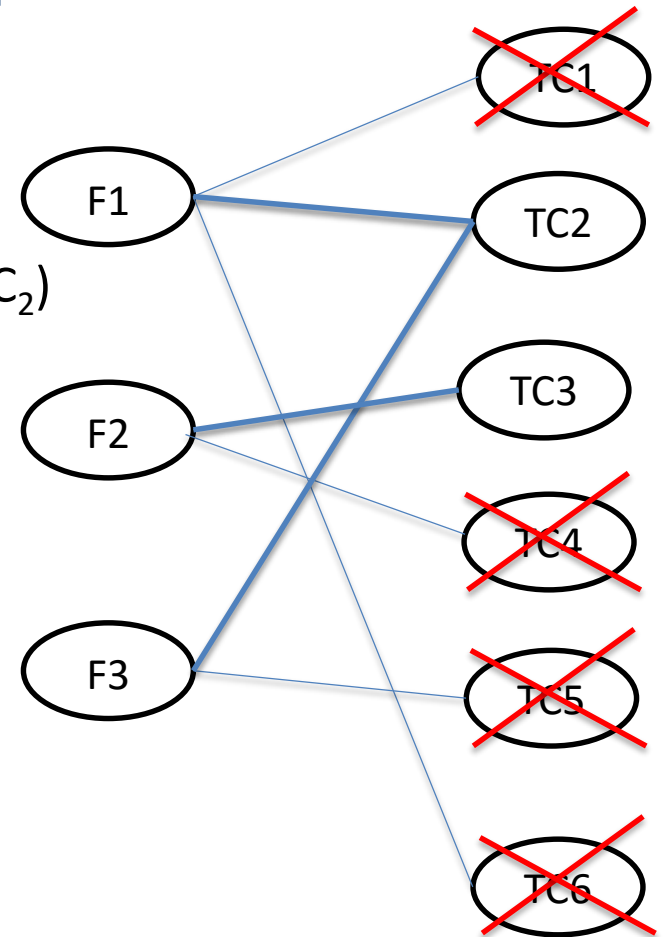
Introducing model presolve

F_1 in $\{1, 2, 6\} \rightarrow F_1 = 2$ as $\text{cov}(TC_1) = \text{cov}(TC_6) \subset \text{cov}(TC_2)$
withdraw TC_1 and TC_6

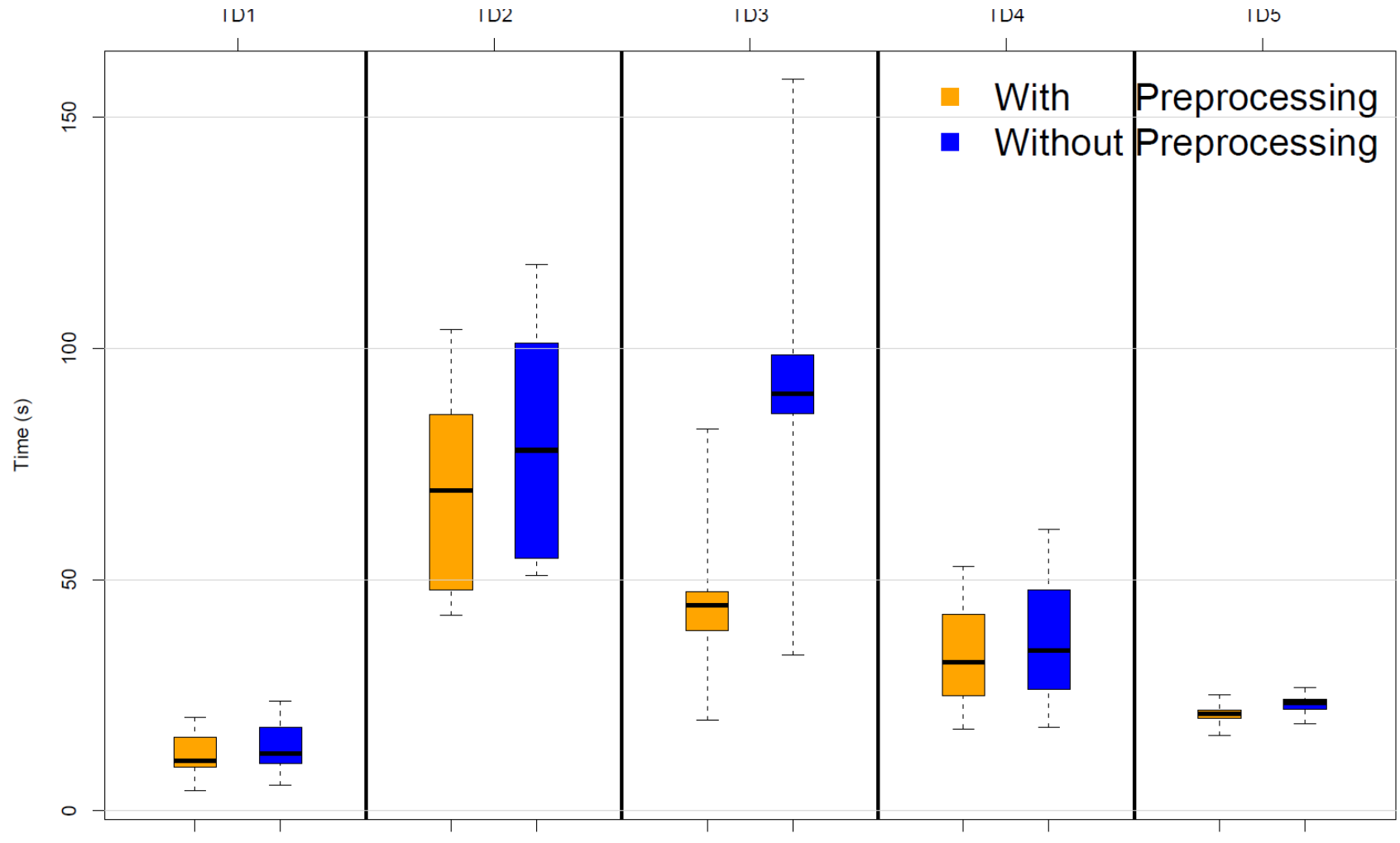
F_3 is covered \rightarrow withdraw TC_5

F_2 in $\{3,4\} \rightarrow$ e.g., $F_2 = 3$, withdraw TC_4

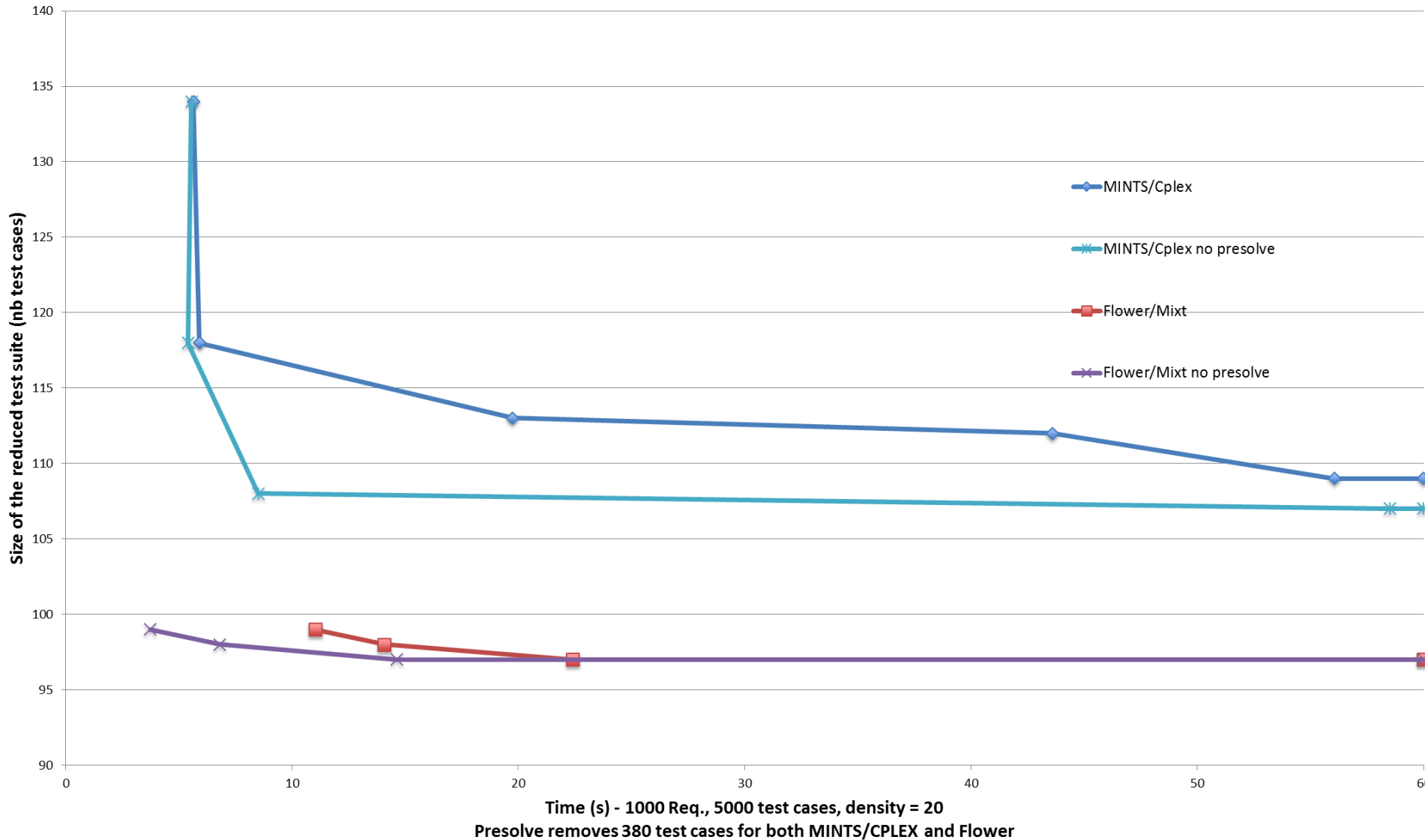
We proposed an iterative algorithm to apply these preprocessing rules to simplify the problem



Presolve: Experimental results (1)



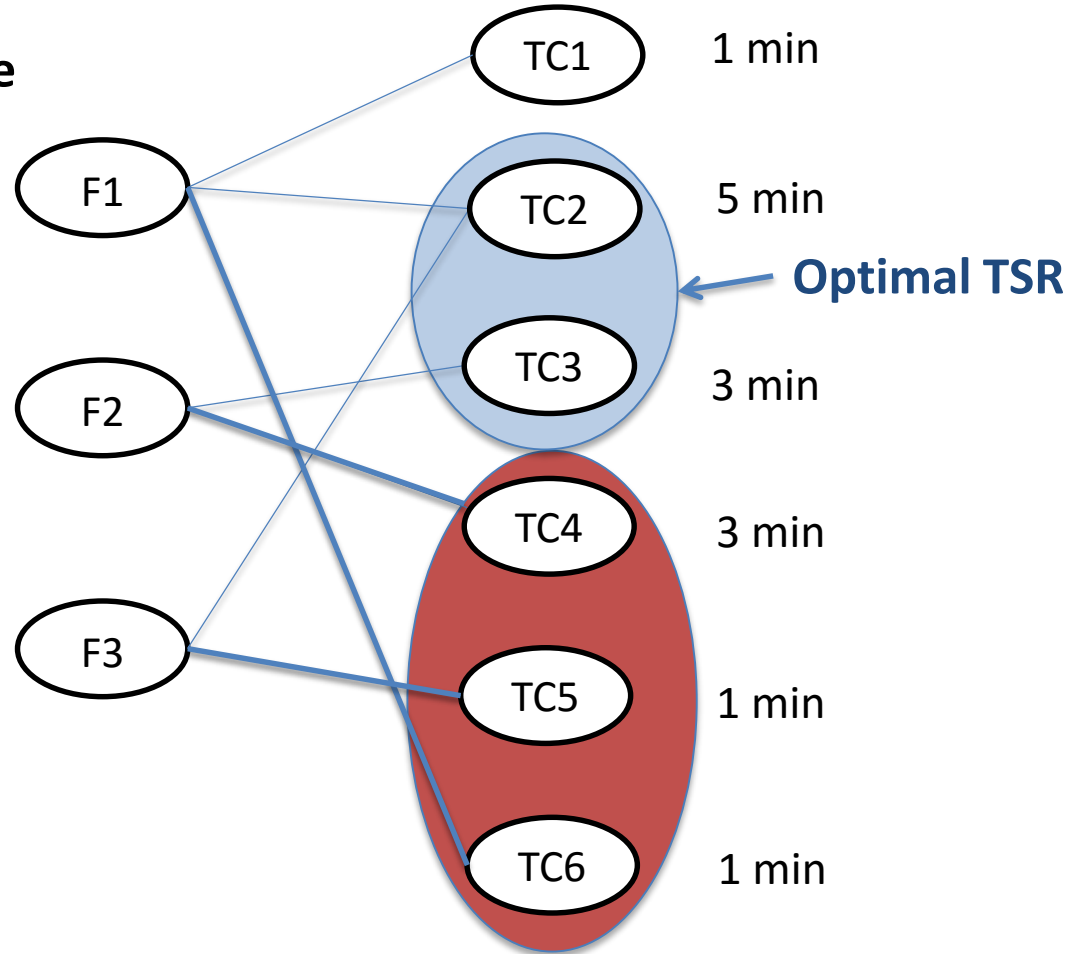
Presolve: Experimental results (2)



Multi-objectives Test Suite Reduction

Optimal TSR: the core problem

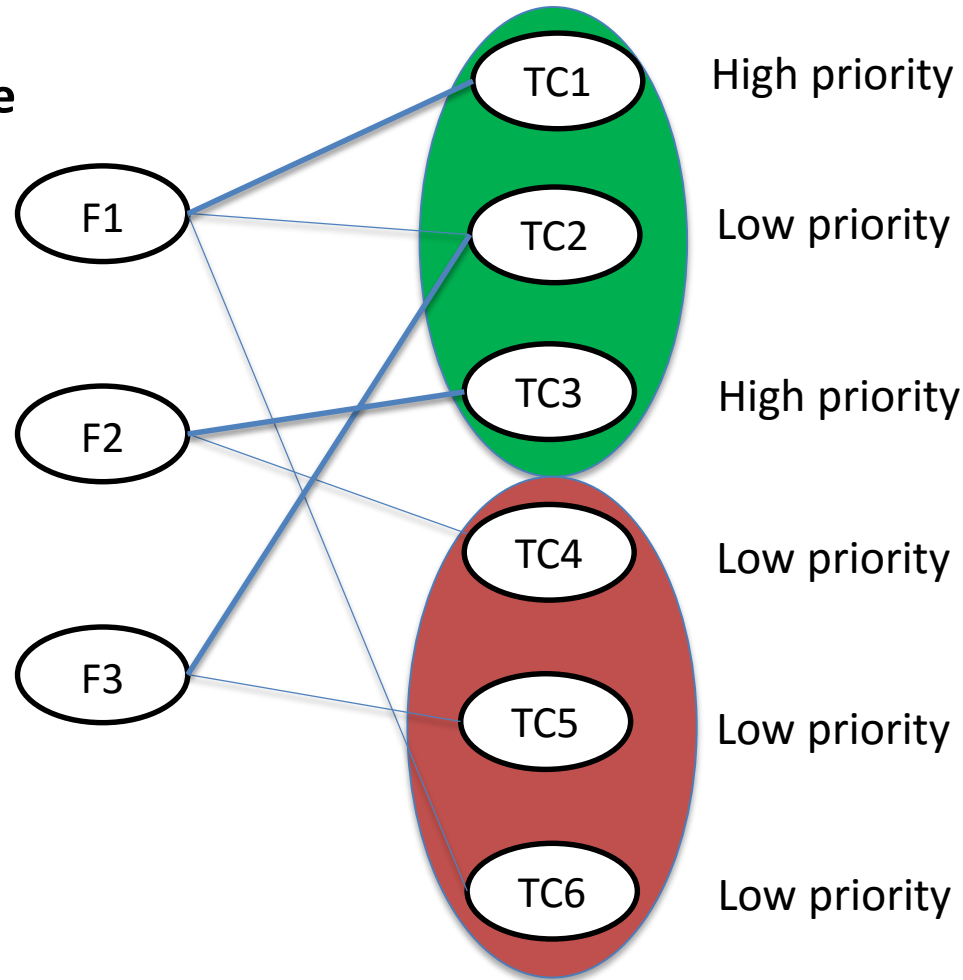
Requirements coverage is always a prerequisite but other criteria than the size of the test suite are also sought:



Execution time!

Optimal TSR: the core problem

Requirements coverage is always a prerequisite but other criteria than the size of the test suite are also sought:



Fault revealing capabilities!

Proposed approaches

1. Actual multi-objectives optimization with search-based algorithms (Pareto Front)

S. Wang, D. Buchmann, S. Ali, A. Gotlieb, D. Pradhan, and M. Liaaen. Multi-objective test prioritization in software product line testing: An industrial case study. In *Software Product Line Conference (SPLC'14)*, Florence, Italy, 2014.

Aggregated cost function using RW-algo, URW-algo, and many others

Based on computed values

S. Wang, S. Ali, and A. Gotlieb. Random-weighted search-based multi-objective optimization revisited. In *Int. Symp. on Search-Based Software Engineering (SSBSE'14)*, Fortaleza, Brazil, 2014.

No constraint model!

2. Cost-based single-objective constrained optimization

Based on a CP model with global constraints

Constrained optimization model!

Flower/C: An extension of Flower with costs

R_1, \dots, R_n : Requirements

t_1, \dots, t_m : Test cases - Each test case t_i is associated a unitary cost c_i

O_1, \dots, O_m : Occurrences variables

Minimize TotalCost

s.t

$\text{gcc}((R_1, \dots, R_n), (t_1, \dots, t_m), (O_1, \dots, O_m))$

for $i=1$ to m do $B_i = (O_i > 0)$

$\text{scalar_product}((B_1, \dots, B_m), (c_1, \dots, c_m), \text{TotalCost})$

where scalar_product encodes $B_1 * c_1 + \dots + B_m * c_m = \text{TotalCost}$

Industrial Application

TITAN

Multisite features

- Easy 9000/9100 high Conference (9000: 223 Model)
- Full individual audio and video transcoding
- Individual operators in Multisite CP (lines out soft view)
- H.323 SIP/VSP in the same conference
- Support for Presentation (223/224 CP) from any party
- Band Transparen Automatic CP (sends)
- H.264, Encrypted, Dual Stream from any site
- RTP Overlapping
- Dual In Dual out
- Additional bandwidth call (transcode required)
- Conference rates up to 16 Mbps

Audio features

- CD Quality (20KHz) Mono and Stereo
- 6/8 separate acoustic echo cancellers
- 8-point Audio mixer
- Automatic Gain Control (AGC)
- Automatic Noise Reduction
- Active for synchronization

Audio standards

- G.711 (S, F22, G, F22, 54 Kbps) 12K Mpa MPEG4 AAC-LC, AAC-LC Stereo

IP network features

- DNS lookup for service configuration
- Differentiated Services (Diffs)
- RPI adaptive bandwidth management (on)
- Auto gatekeeper discovery
- Dynamic dialtone and trigger buffering
- H.245 STMF series in H.323
- Data and Time support via SIP
- Packet loss based Overlapping
- UDP Damping
- TCPSP
- DSCP
- RAS to Network authentication
- ClassPath

Live video res.

- 150 x 144@30 Ip (H.264)
- 200 x 200@30 Ip (H.264)
- 352 x 288@30 Ip (H.264)
- 512 x 448@30 Ip (H.264)
- 768 x 576@30 Ip (H.264)
- 1024 x 768@30 Ip (H.264)
- 1280 x 1024@30 Ip (H.264)
- 1536 x 1368@30 Ip (H.264)
- 2048 x 1536@30 Ip (H.264)
- 2880 x 1920@30 Ip (H.264)
- 3840 x 2880@30 Ip (H.264)
- 512 x 288@30 Ip (H.264)
- 768 x 448@30 Ip (H.264)
- 1024 x 768@30 Ip (H.264)
- 1280 x 1024@30 Ip (H.264)
- 1536 x 1368@30 Ip (H.264)
- 2048 x 1536@30 Ip (H.264)
- 2880 x 1920@30 Ip (H.264)
- 3840 x 2880@30 Ip (H.264)
- 512 x 288@30 Ip (H.264)
- 768 x 448@30 Ip (H.264)
- 1024 x 768@30 Ip (H.264)
- 1280 x 1024@30 Ip (H.264)
- 1536 x 1368@30 Ip (H.264)
- 2048 x 1536@30 Ip (H.264)
- 2880 x 1920@30 Ip (H.264)
- 3840 x 2880@30 Ip (H.264)
- 512x288 from 1472Mbps

Security features

- Management via HTTPS and SSH
- IP Administration Firewall
- Menu Administration Password
- CloudIP services
- Network Settings protection

Bandwidth

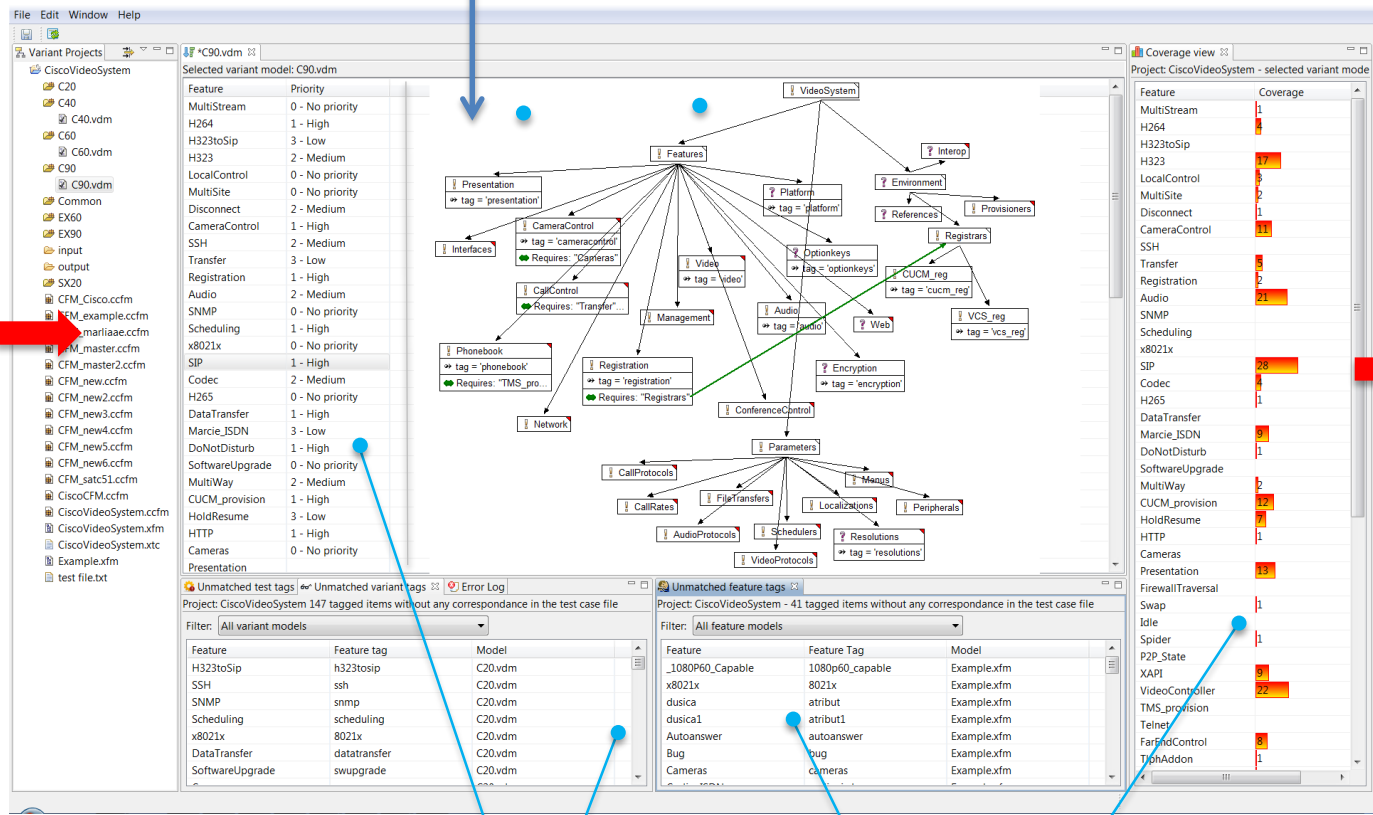
- H.323/SIP up to 16 Mbps point to point
- Up to 16 Mbps total Multisite bandwidth

Protocols

- H.323
- SIP



Variability model to describe a software product line



The screenshot displays the TITAN software interface. On the left, a file explorer shows a project named 'CiscoVideoSystem' with various variant models (C20, C40, C60, C90, etc.). A red arrow points from this list to the central feature model tree. The tree shows a hierarchy of features like 'Presentation', 'Video', 'Audio', and 'Parameters', with sub-features and their dependencies. Below the tree are diagnostic views showing 'Unmatched test tags' and 'Unmatched feature tags'. On the right, a 'Coverage view' table shows the coverage for various features across different test cases, with a red arrow pointing from the table to a stack of green documents representing an optimized test suite.

Feature	Coverage
MultiStream	1
H264	1
H323toSip	1
H323	17
LocalControl	5
MultiSite	2
Disconnect	1
CameraControl	11
SSH	5
Transfer	3
Registration	21
Audio	5
SNMP	1
Scheduling	1
x8021x	1
SIP	28
Codec	4
H265	1
DataTransfer	1
Marcie_JSDN	9
DoNotDisturb	1
SoftwareUpgrade	1
MultiWay	1
CUCM_provision	12
HoldResume	1
HTTP	1
Cameras	13
Presentation	1
FirewallTraversal	1
Swap	1
Idle	1
Spider	1
P2P_State	1
XAPI	9
VideoController	22
TMS_provision	1
Telnet	1
FarEndControl	9
TShAddOn	1

Unoptimized test suite

Optimized (reduced/prioritized) test suite

Diagnostic views, feature coverage

TITAN

- Reusing Pure::Variants plug-in for feature modelling and editing
- Desktop version + web-based service
- Patent under advisement in the US
- Deployed at Cisco Systems
- Commercial development (funded under the RCN's FORNY program)

Conclusions

- Global constraints and CP can efficiently and effectively tackle difficult software validation problems – experimental results and initial industrial case studies
- So far, the links between feature modelling and software product line engineering and software validation has been little studied
- There is room for Research and Innovation in that area!