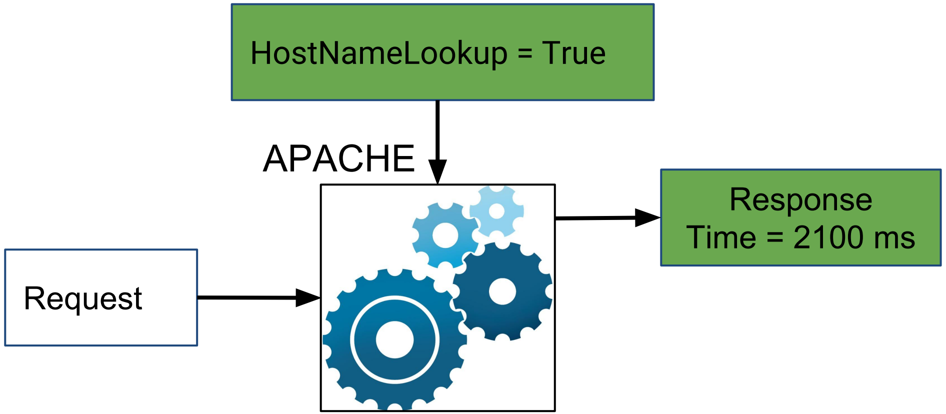
Frugal: Cheaper Methods for SBSE

Vivek Nair

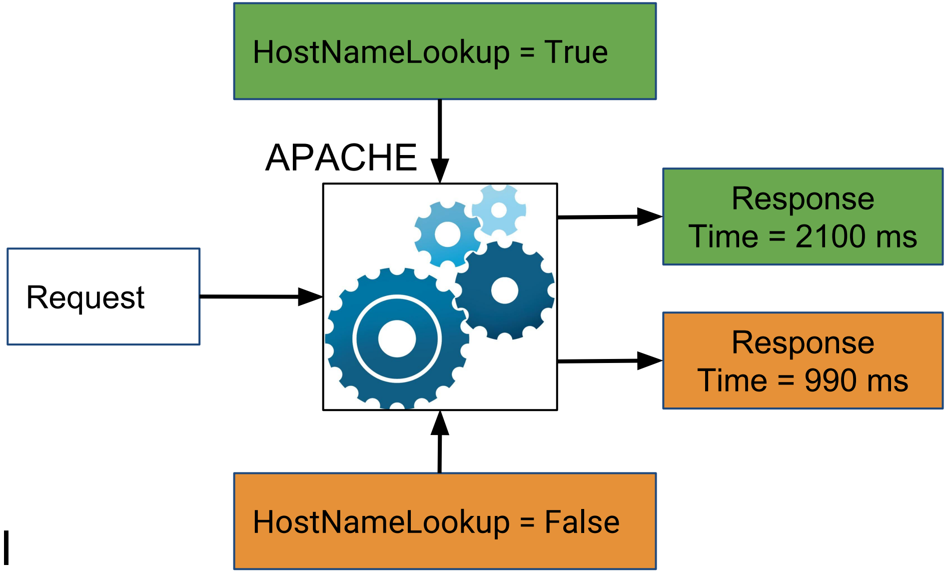
Why configurations are so important?

- Software systems are configurable
- Configurations are parameters to control the behavior of a system
  - Configurations of ***Apache:***
    - HostNameLookups
    - FollowSimLinks
    - ….
- Different configurations of system will result in different performance



Why configurations are so important?

- Software systems are configurable
- Configurations are parameters to control the behavior of a system
  - Configurations of ***Apache:***
    - HostNameLookups
    - FollowSimLinks
    - ….
- Different configurations of system will result in different performance



Example

Conf.	Features							
	x_1	x_2	x_3	..	x_i	..	..	x_N
1	1	0	1	0	0	0	1	1
2	0	1	1	1	1	0	0	1
3	1	0	0	1	0	1	0	0
4	1	1	0	1	0	1	0	1
5	1	0	1	1	0	1	1	0

Find the fastest configuration setting for given a sample program?

Just run it?

Example

Conf.	Features							
	x_1	x_2	x_3	..	x_i	..	..	x_N
1	1	0	1	0	0	0	1	1
2	0	1	1	1	1	0	0	1
3	1	0	0	1	0	1	0	0
4	1	1	0	1	0	1	0	1
5	1	0	1	1	0	1	1	0
.								
.								
3,932,160	1	0	1	1	0	1	1	0

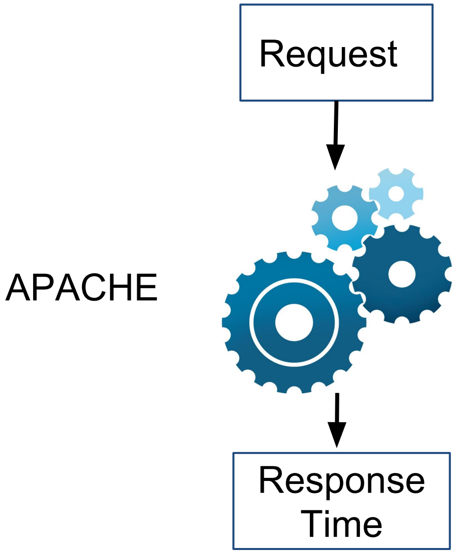
Find the fastest configuration setting for given a sample program?

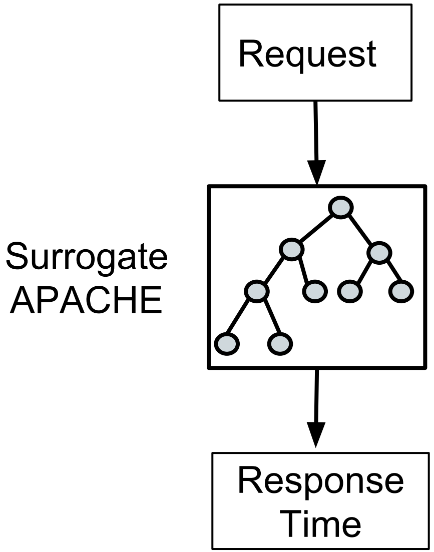
Just run it?

How about now?

We need a Surrogate!

Surrogate is a cheap version of the actual system





Who endorses Surrogates?

Other Communities

- Aerospace
  - Axial compressor blade shape optimization [Samad08]
  - Hydraulic turbine diffuser shape optimization [Marjavaara07]

- Engineering Design
  - Enhanced oil recovery process [Sanchez06]
  - Design of composite materials [Sakata08]
  - Alkaline-surfactant-polymer flooding processes [Zerpa05]

Software Engineering

No surrogates….

Who endorses Surrogates?

Other Communities

- Aerospace
  - Axial compressor blade shape optimization [Samad08]
  - Hydraulic turbine diffuser shape optimization [Marjavaara07]
- Engineering Design
  - Enhanced oil recovery process [Sanchez06]
  - Design of composite materials [Sakata08]
  - Alkaline-surfactant-polymer flooding processes [Zerpa05]

Software Engineering

No surrogates….

Most Similar But ***NOT Surrogates***:

- Heuristic method to predict response times [Siegmund'12]
- Random Sampling to build a prediction model [Guo'13, Sarkar'15]

Our Surrogate Method!

Our method “WHAT” is better than the state of the art

- Similar result using 2 to 10 times less evaluations
- Predictions are more stable

Paper Submitted

Vivek Nair, Tim Menzies, Norbert Siegmund, Sven Apel. Faster Discovery of Faster System Configurations with Spectral Learning. Submitted to FSE - 2016

BACKGROUND

“*Search*” in Software Engineering

What is the: [Harman’12]

- best way to structure this system to enhance its maintainability?
- smallest set of test cases that covers all branches?
- fastest configuration of this system to run this benchmark program?

Software Engineering problems are

- MultiObjective [Mkaouer’15]
  - The are more than one objective to optimize
- Multi-Modal
  - There are more than one optimum solution
- Non-Separability
  - The optimum of one of the objectives is not the optimum for the other objective/s.
- High Dimensions
  - Number of dimensions of the search space is large

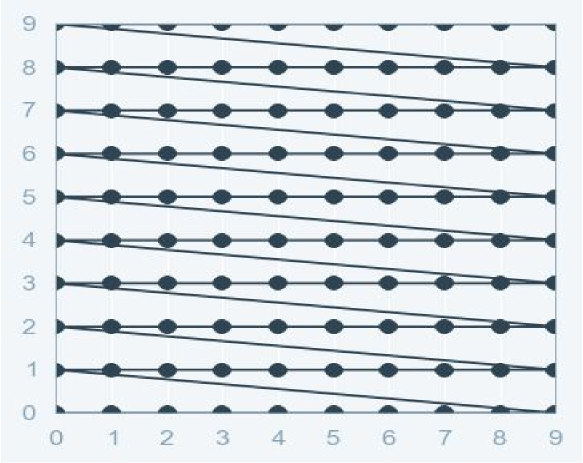
Which optimization algorithms can we use?

Mathematical optimization

- Based on the property of objective function and constraint function:
  - linear programing
  - non-linear programing
- Assumes properties like differentiability etc.

Grid Search

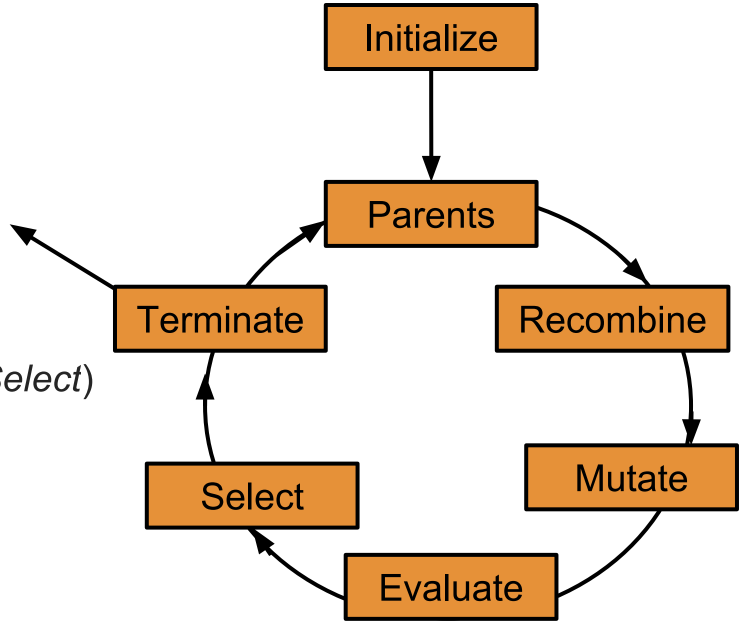
- Divide dimensions into bins
- Choose one from each bin
- Slow and can miss important optimization opportunities



Which optimization algorithms can we use?

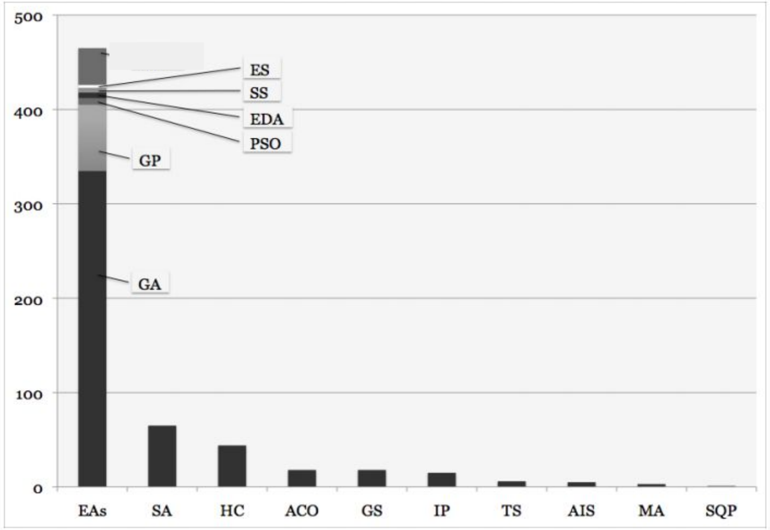
Evolutionary Algorithms

1. Initial Population (*Parent*)
2. While stoppingCriteria is True:
   a. Offspring = Reproduction (*Recombine* + *Mutate*)
   b. Evaluate Fitness (*Evaluate*)
   c. Replace least-fit population with new offspring (*Select*)
3. Return (Population)



Biased towards EA

- Simple implementation
  - Basic EA application can be coded up in 50 lines of python
- Distributed computation
  - Algorithms can be parallelized
- Generation of new ideas that have not been explored before



EA is most explored technique in SBSE [Harman'12]

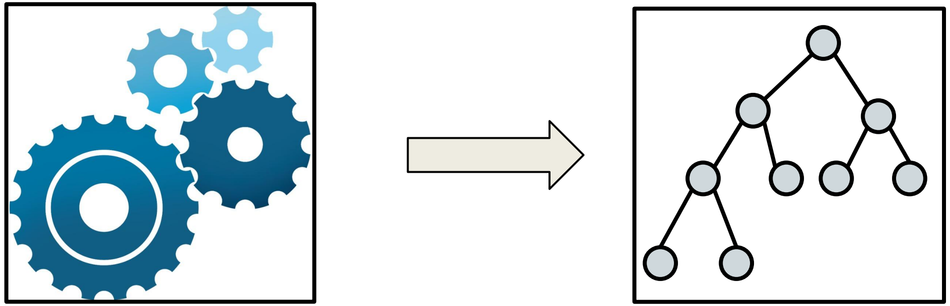
EA is really slow!

EAs require a *high number of objective function evaluations*

- Evaluation of single instance of software /hardware co-design problem can ***take weeks*** [Zuluaga'13]
- Test suite generation using EA can ***take weeks*** [Harman'12]
- Popular EA (NSGA-II) ***taking 7 days*** of execution time for Aviation Models [Krall'15]

Surrogate models might be the answer?

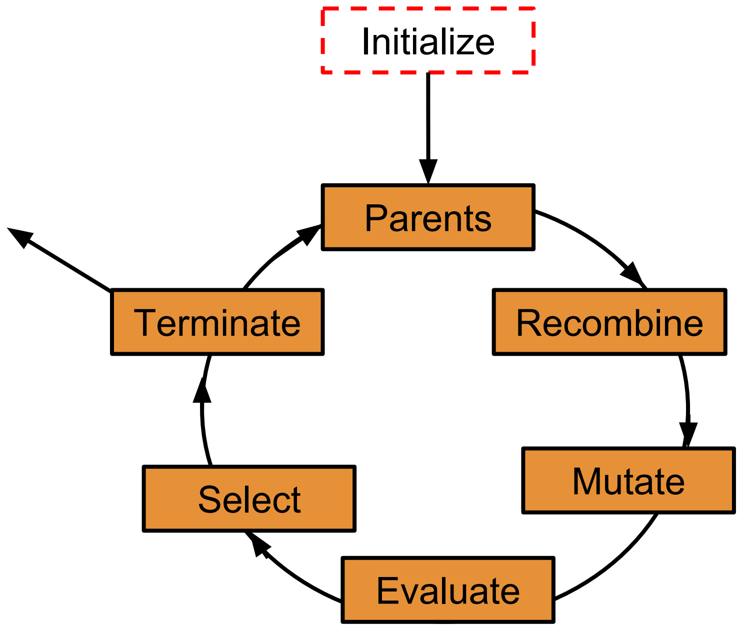
- Surrogates



- Motivation
  - Replacement of expensive function, evaluated many times
  - Widely used in Airfoil design, CFD, reservoir planning etc.
  - No known usage in Software Engineering

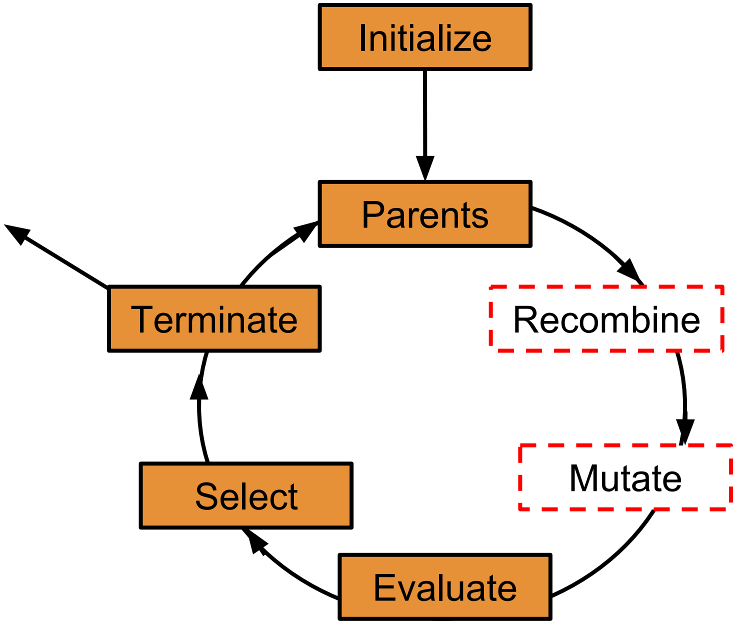
Surrogate can also be used to inform

- Initialization
    - Use only the best candidates evaluated using a surrogate [Rasheed'00]



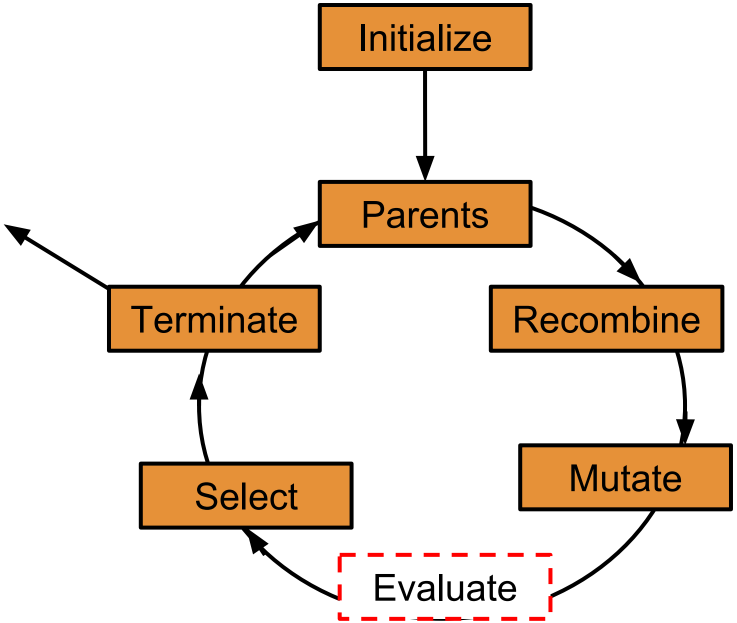
Surrogate can also be used to inform

- Initialization
    - Use only the best candidates evaluated using a surrogate [Rasheed'00]
- Recombination + Mutation
    - Create multiple children and use the fittest of them all [Loshchilov'10]
    - Create local surrogate and and search locally [Abboud'01]

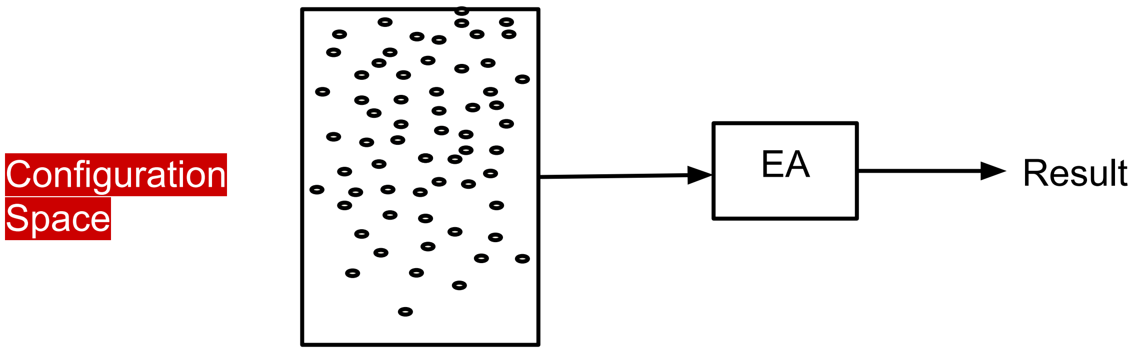


Surrogate can also be used to inform

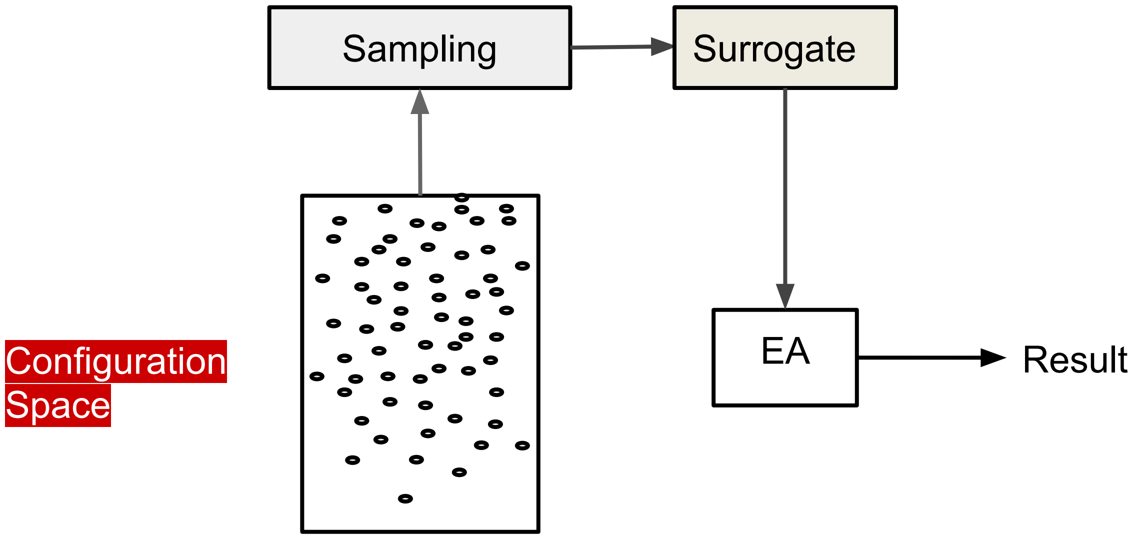
- Initialization
    - Use only the best candidates evaluated using a surrogate [Rasheed00]
- Recombination + Mutation
    - Create multiple children and use the fittest of them all [Loshchilov'10]
    - Create local surrogate and and search locally [Abboud'01]
- Evaluate
    - Multiple Surrogates [Zhou'07]
    - **WHAT** is an evaluate surrogate



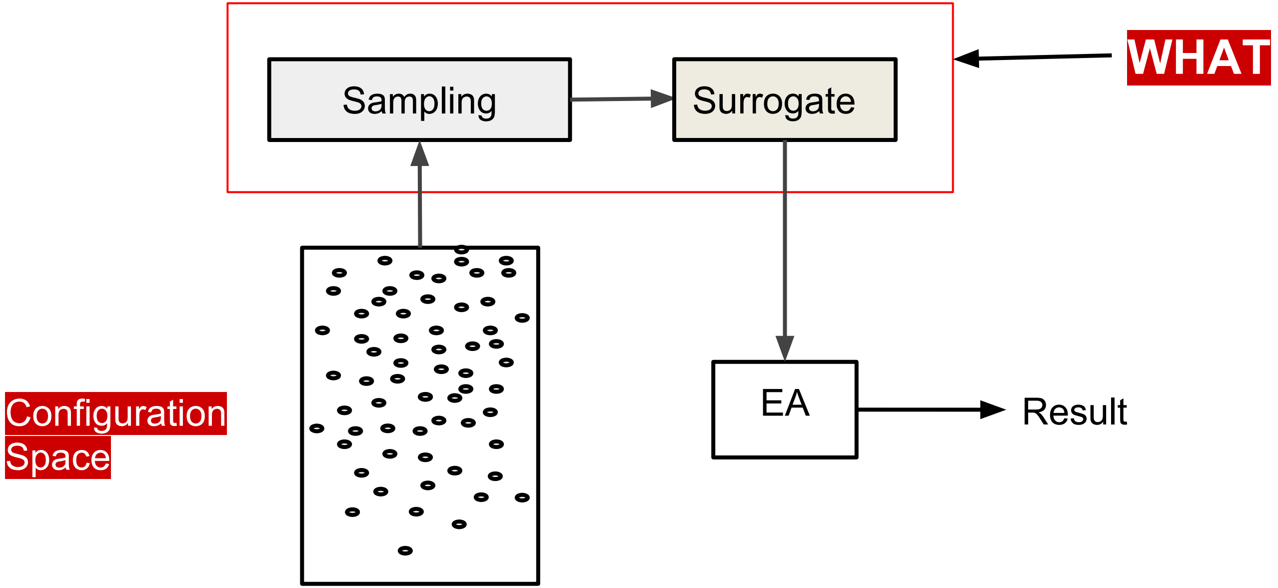
To Summarize



To Summarize



To Summarize

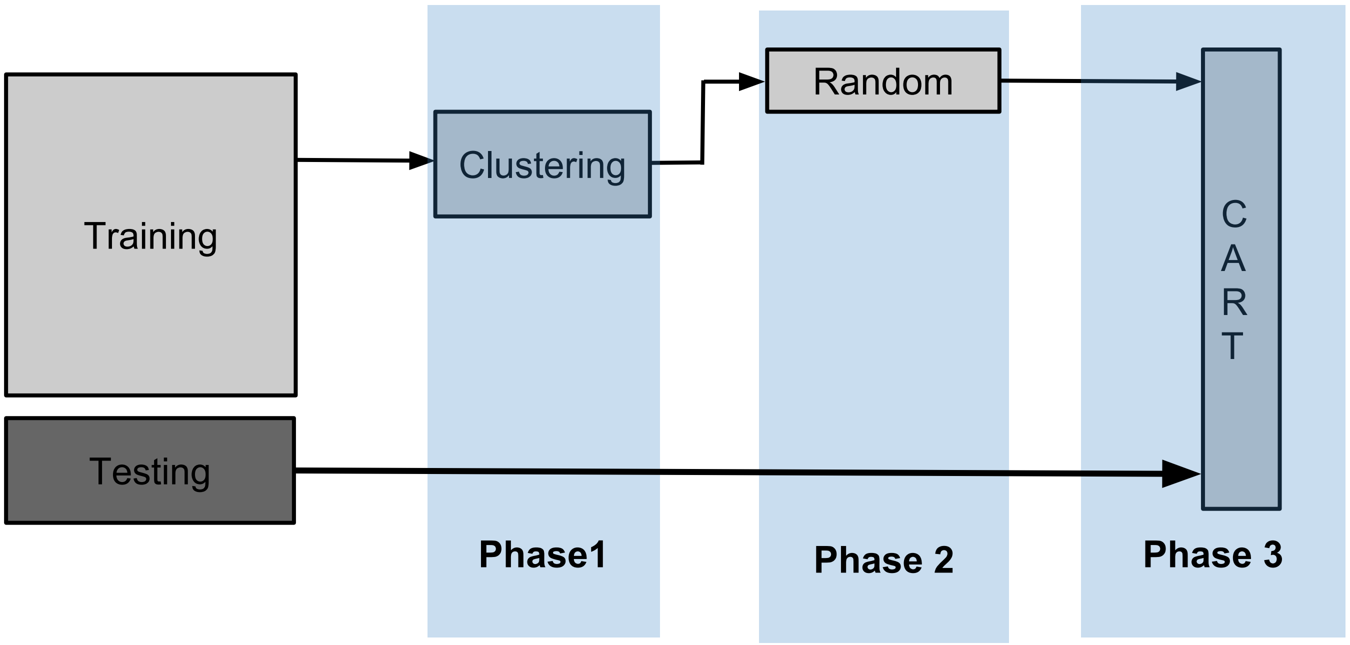


APPROACH

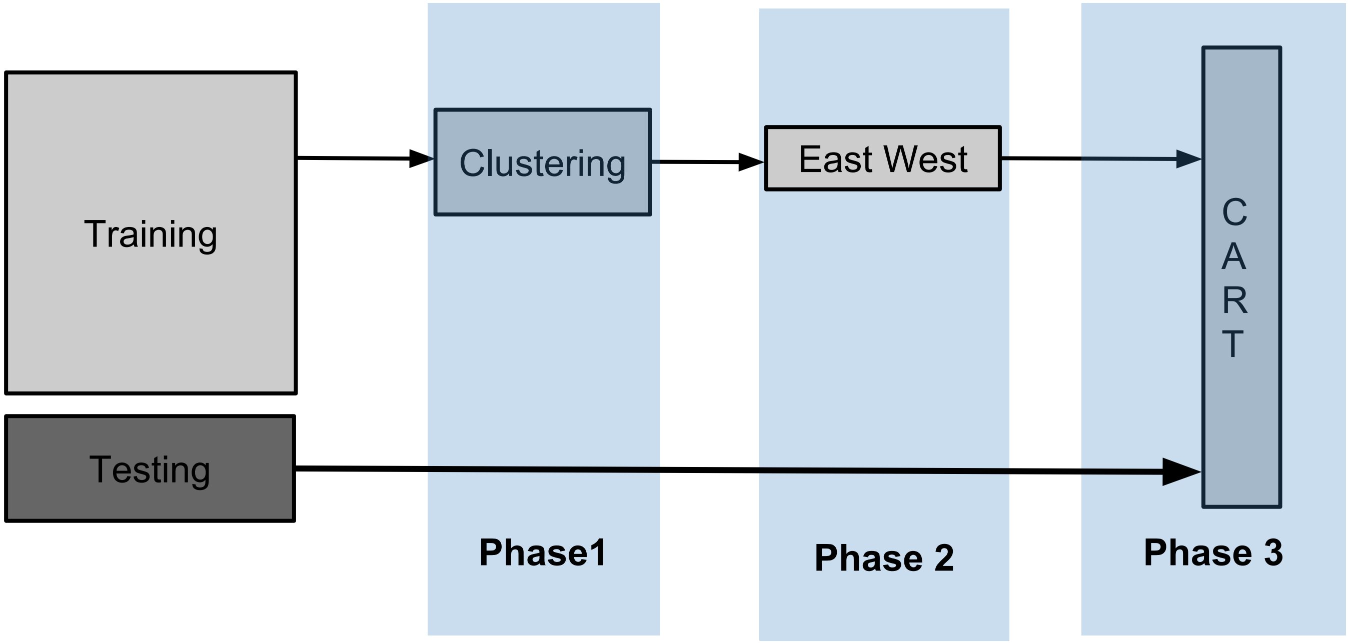
WHAT = Clustering + Sampling

- Phase 1: Clustering
  - WHERE
- Phase 2: Sampling
  - Random Sampling - Select any point at random
  - East West Sampling - Find extreme points on the dimension of highest variance
  - Exemplar - The point with minimum performance measure
- Phase 3: Generate Surrogate - CART
  - Samples selected by our sampler is used to train a CART model

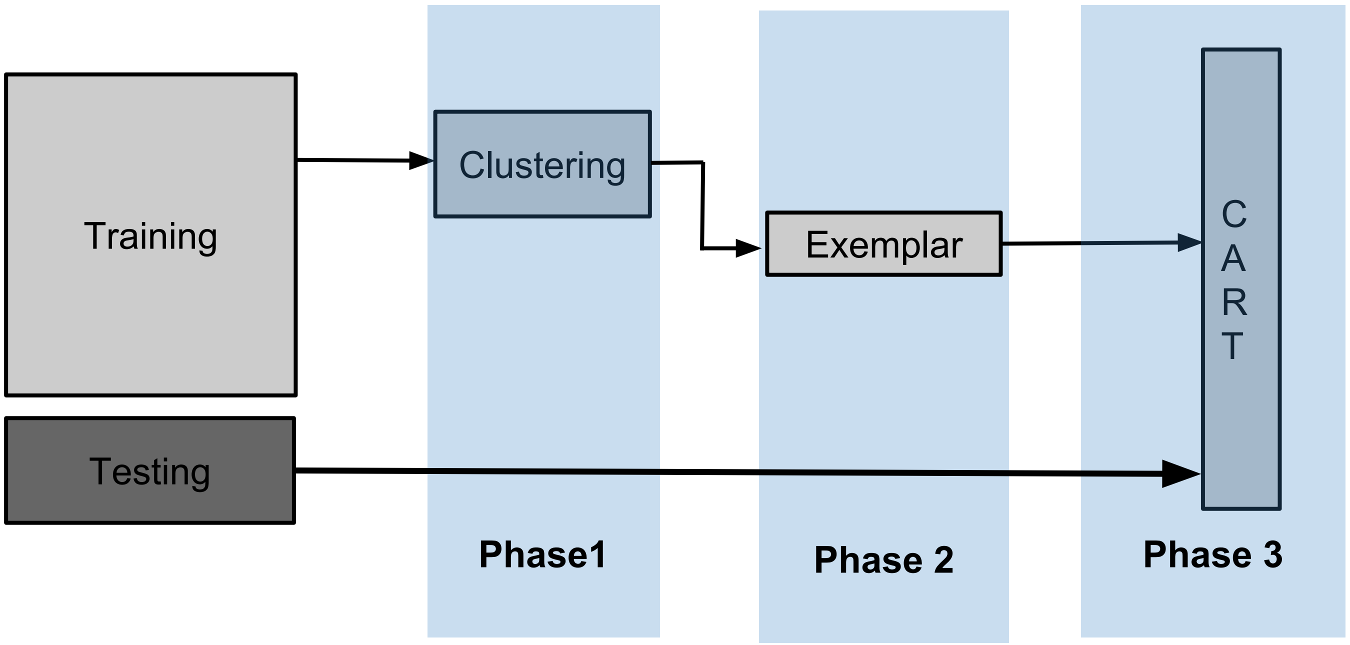
WHAT


Training
Testing
Clustering
Random
CART
Phase1
Phase 2
Phase 3

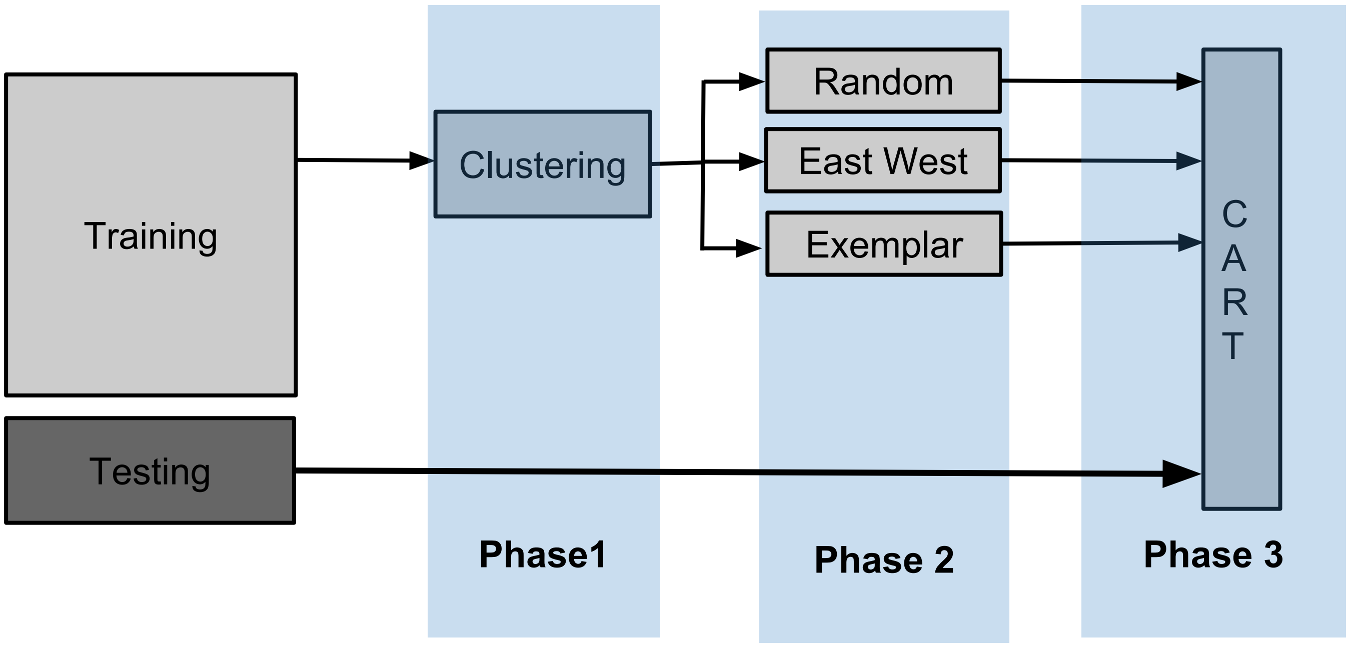
WHAT


Training
Testing
Clustering
East West
CART
Phase1
Phase 2
Phase 3

WHAT


Training
Testing
Clustering
Exemplar
CART
Phase1
Phase 2
Phase 3

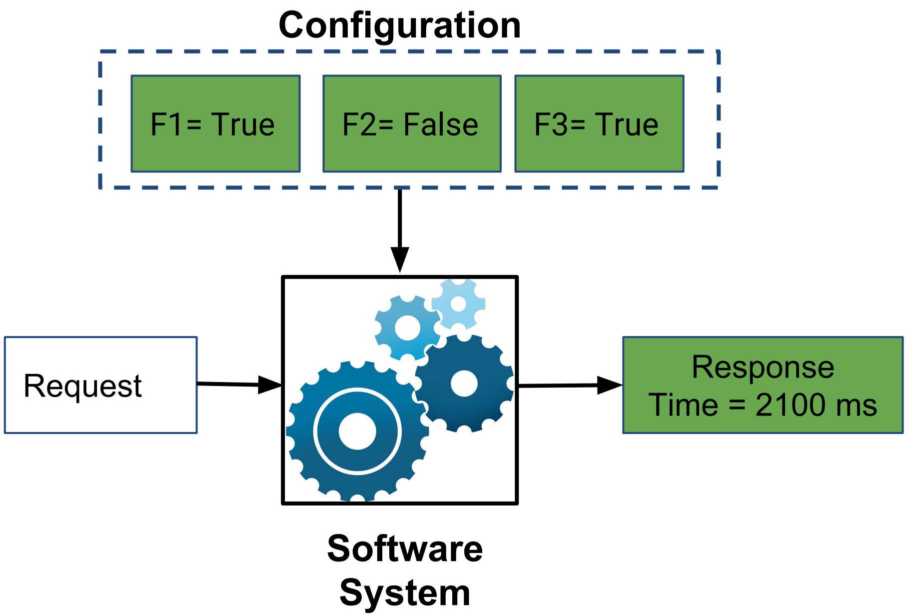
WHAT



Training
Testing
Clustering
Random
East West
Exemplar
C A R T
Phase1
Phase 2
Phase 3

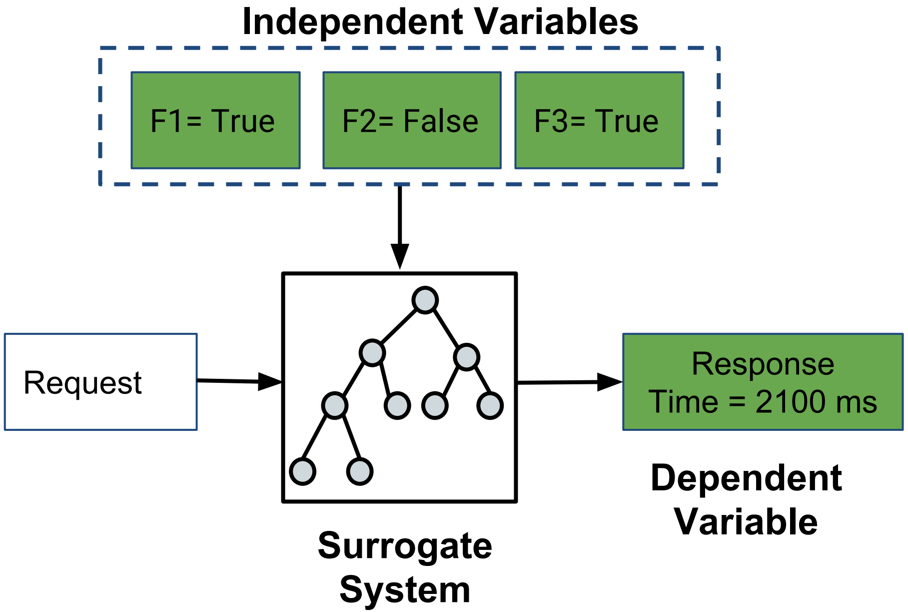
Definition

- Real System
  - Features can be either True or False
  - Configuration is a set of features
  - Each configuration has a corresponding response time or performance measure



Definition

- Real System
  - Features can be either True or False
  - Configuration is a set of features
  - Each configuration has a corresponding response time or performance measure
- Surrogate System
  - Configuration = independent variable
  - Performance measure = dependent variable



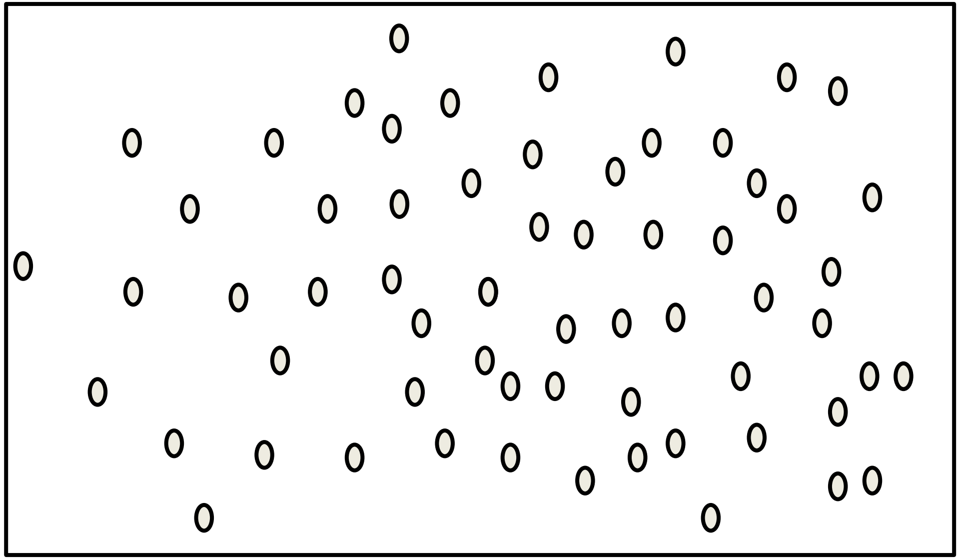
Phase 1: Clustering

- Clustering via WHERE
  - Novel near-linear time spectral learner
  - Exploits underlying lower dimensionality of search space

- In brief:
  - Find a dimension “d” with most variance
  - Project points to “d”
  - Split data at median “d”
  - Recurse
  - Stop when |n| < sqrt(N)

- ***Future work:***
  - Fast Spectral clustering [Yan’09]
- In brief:
  - Polynomial time operations
    - An initial k-means pass
    - O(N^2) operations on the centroids founds by K-means
    - Final pass: map all points to the centroids found in b

- Number of samples (N) = 64

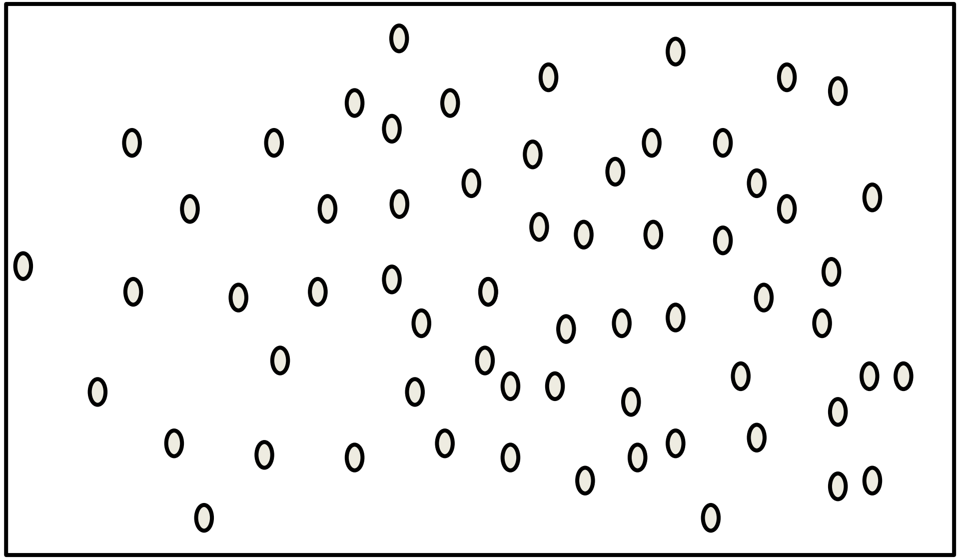


Configuration Space

Algorithm:

- Find a dimension “d” with most variance
- Project points to “d”
- Split data at median “d”
- Recurse
- Stop when |n| < sqrt(N)

- Number of samples (N) = 64

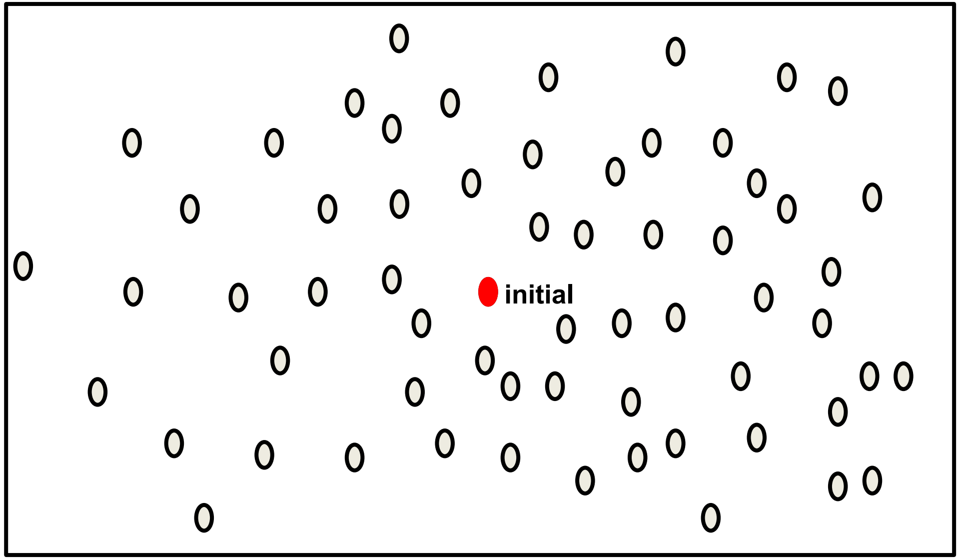


Configuration Space

Algorithm:

- **Find a dimension "d" with most variance**
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

- Number of samples (N) = 64



Configuration Space

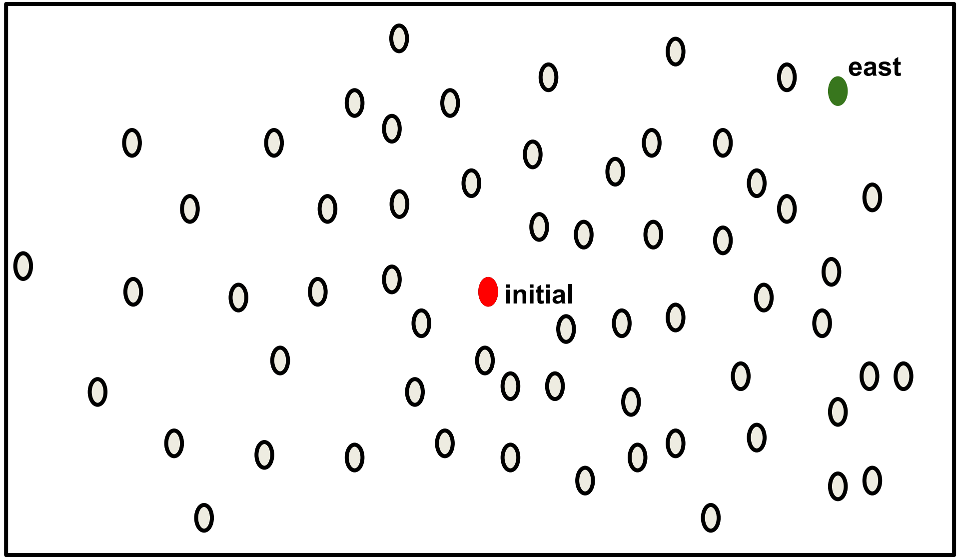
Algorithm:

- **Find a dimension "d" with most variance**
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

- Number of samples (N) = 64

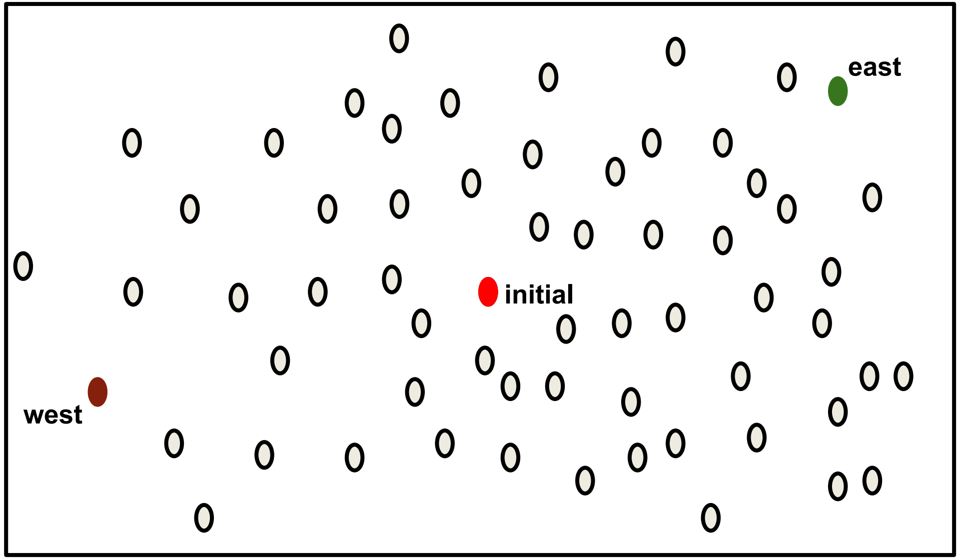
Algorithm:

- **Find a dimension "d" with most variance**
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)



Configuration Space

- Number of samples (N) = 64

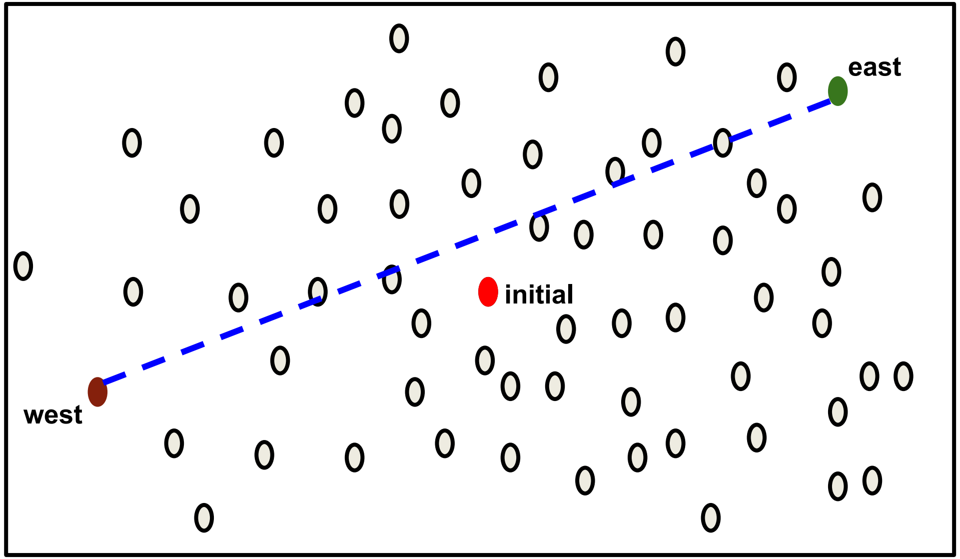


Configuration Space

Algorithm:

- **Find a dimension "d" with most variance**
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

- Number of samples = 64



Configuration Space

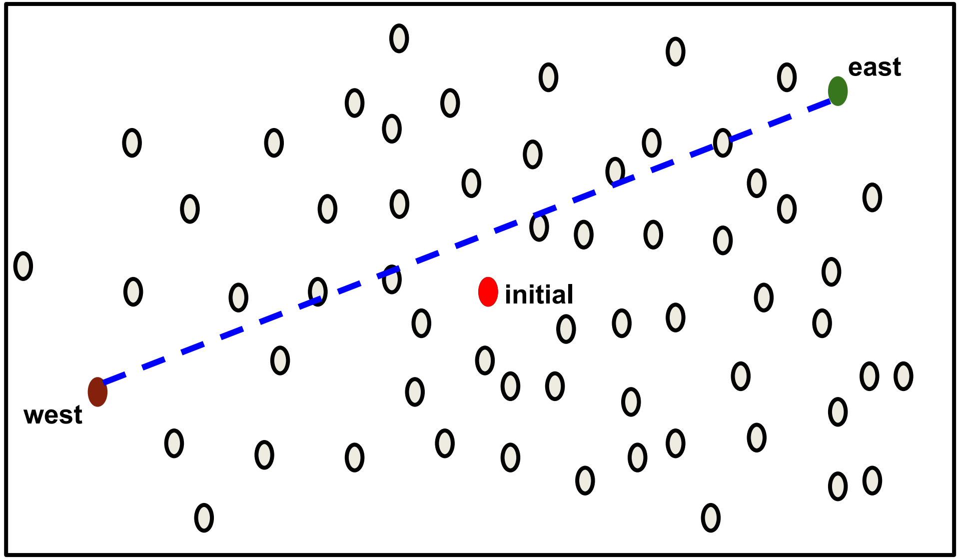
Algorithm:

- **Find a dimension "d" with most variance**
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

- Number of samples (N) = 64

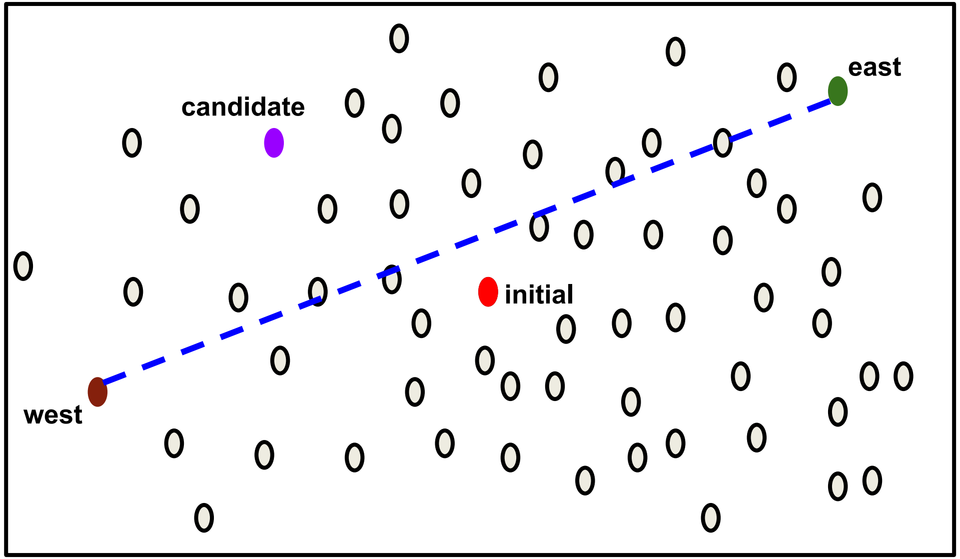
Algorithm:

- Find a dimension “d” with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- **Project points to “d”**
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median “d”
- Recurse
- Stop when |n| < sqrt(N)



Configuration Space

- Number of samples (N) = 64

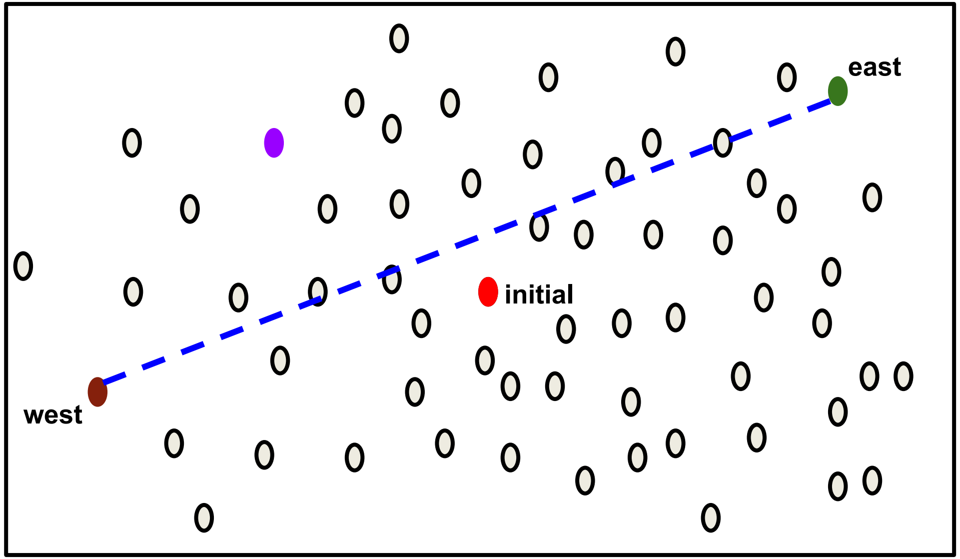


Configuration Space

Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- **Project points to "d"**
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

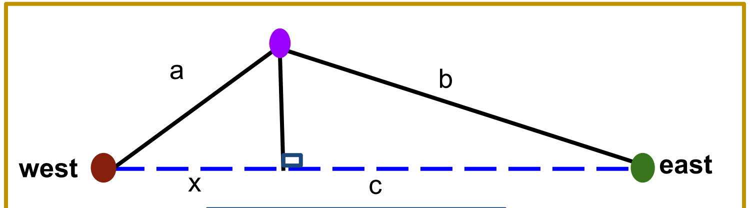
- Number of samples (N) = 64



Configuration Space

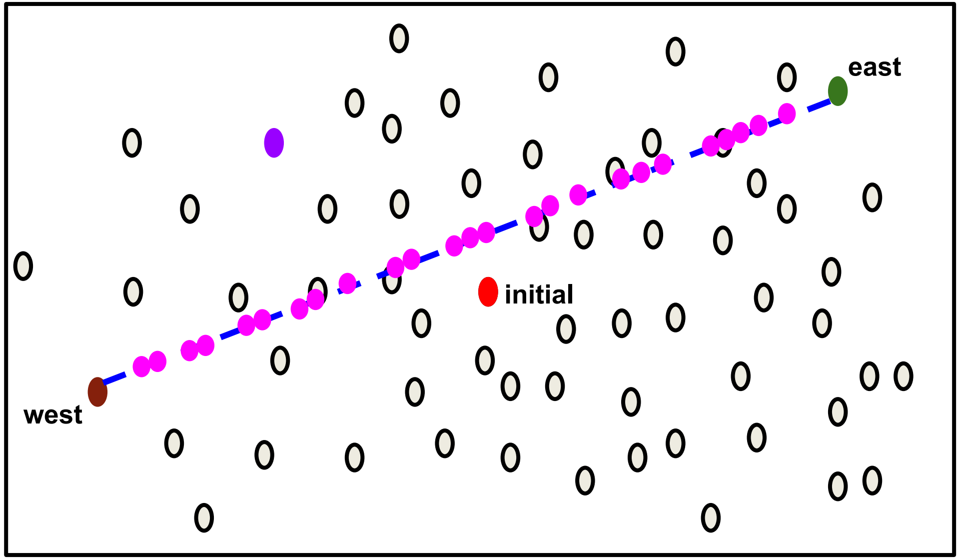
Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- **Project points to "d"**
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)



$$x = \frac{a^2 + c^2 - b^2}{2c}$$

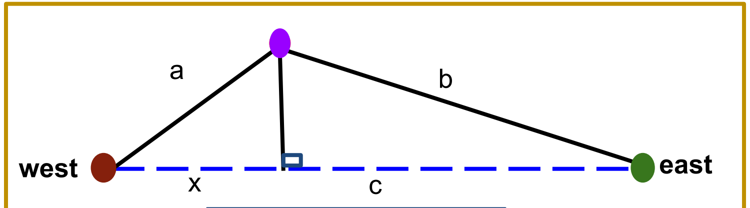
- Number of samples (N) = 64



Configuration Space

Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- **Project points to "d"**
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median "d"
- Recurse
- Stop when |n| < sqrt(N)

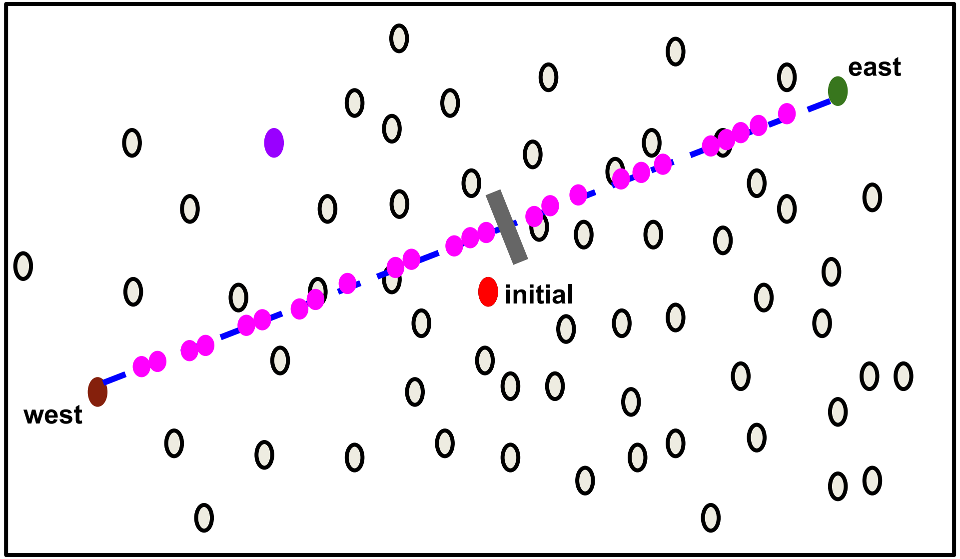


$$x = \frac{a^2 + c^2 - b^2}{2c}$$

- Number of samples (N) = 64

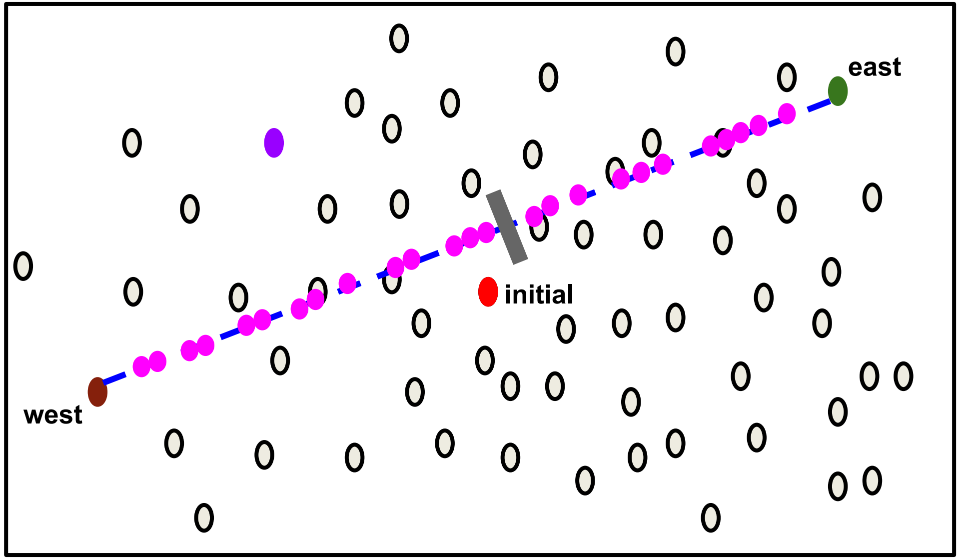
Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- **Split data at median of "d"**
- Recurse
- Stop when |n| < sqrt(N)

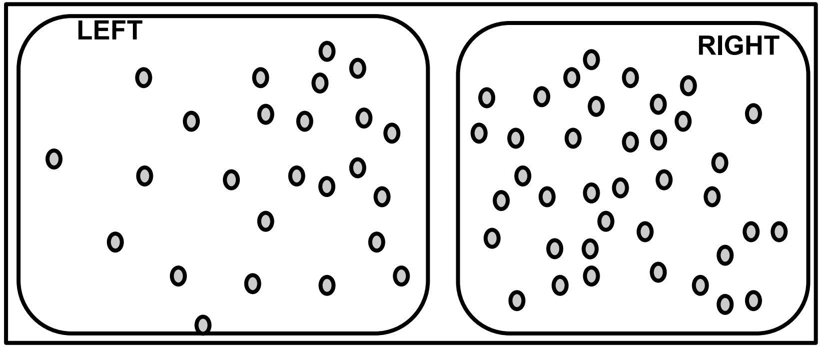


Configuration Space

- Number of samples (N) = 64



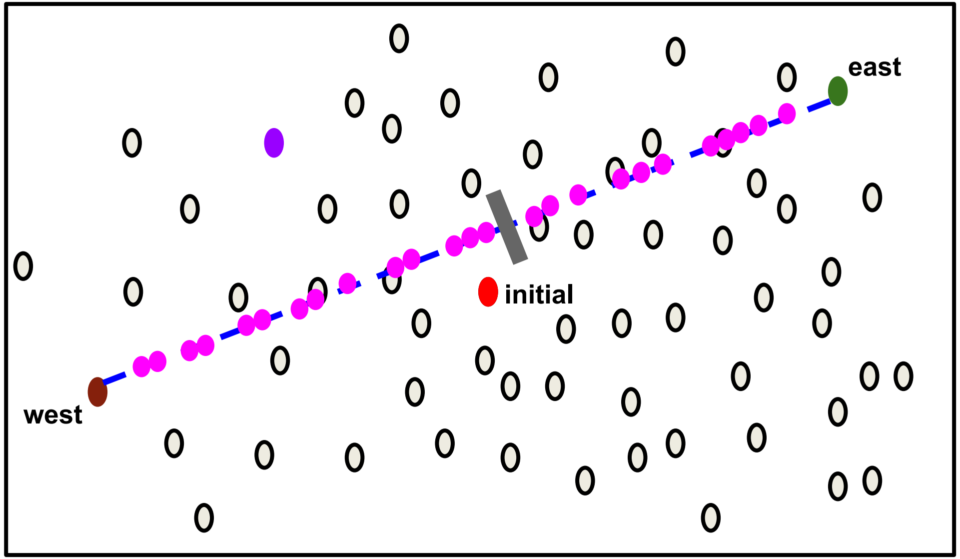
Configuration Space



Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- **Split data at median of "d"**
- Recurse
- Stop when |n| < sqrt(N)

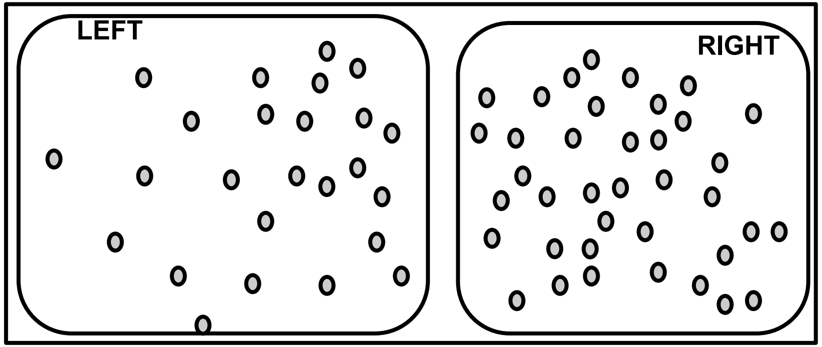
- Number of samples (N) = 64



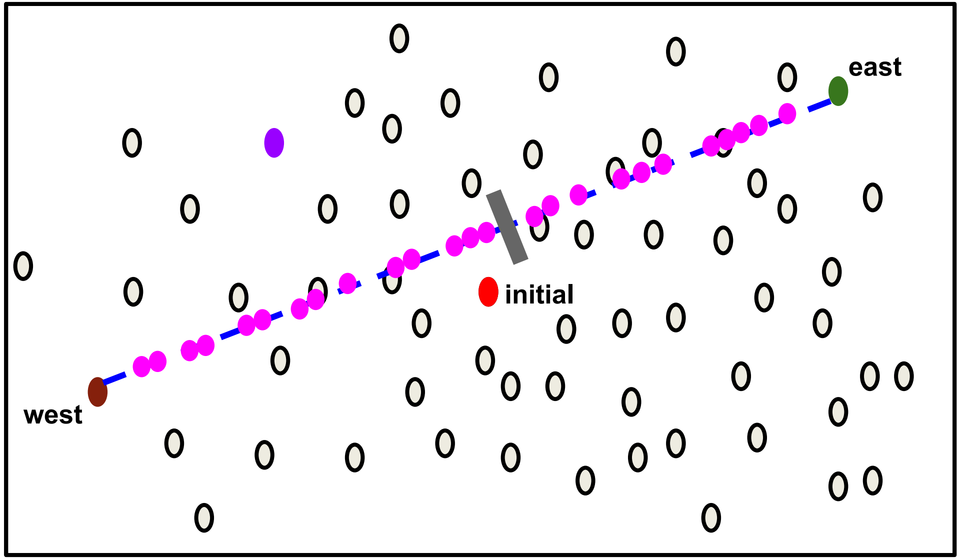
Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median of "d"
- **Recurse**
- Stop when |n| < sqrt(N)

Configuration Space



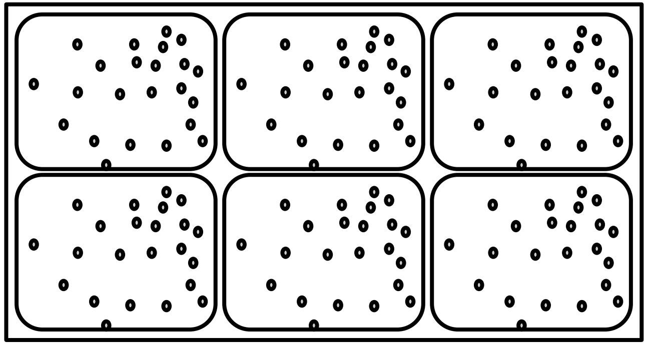
- Number of samples (N) = 64



Algorithm:

- Find a dimension "d" with most variance
  - Choose point at random (initial)
  - Find furthest point (east)
  - Find furthest point from east (west)
- Project points to "d"
  - For all points
    - Choose a point (candidate)
    - Calculate position on d dimension
- Split data at median of "d"
- Recurse
- **Stop when |n| < sqrt(N)**

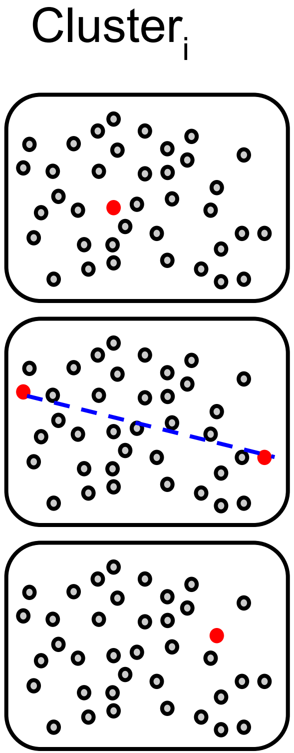
Configuration Space



Phase 2: Sampling

Choosing representative candidates from clusters

- Random
  - Choose a candidate at random
  - Number of evaluations/Cluster = 1
  - Point selected/Cluster = 1
- East-West
  - Choose extreme points in dimension of maximum variance
  - Number of evaluations/Cluster = 2
  - Point selected/Cluster = 2
- Exemplar
  - Choose the best candidate from the cluster
  - Number of evaluations/Cluster = n
  - Point selected/Cluster = 1



Phase 3: Generate Surrogate

- Use the configuration/s sampled from each cluster
- Run the configuration
  - In this work, we performed a table lookup
- Train a CART decision tree learner using:
  - Configurations (Independent Variable)
  - Performance Measure (Dependent Variable)

Experiments

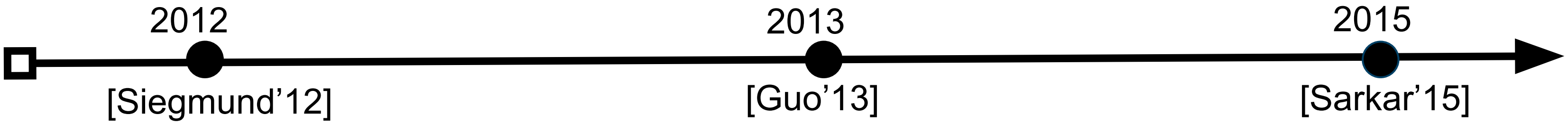
Collecting “Ground Truth” = 26 days of computation

Experiments

- **Datasets Used**:
  - Apache - *open-source Web server*
  - Berkeley DB C (**BDBC**) - *embedded database system written in C*
  - Berkeley DB Java (**BDBJ**) - *BDBC in Java with SQL support*
  - LLVM - *a compiler infrastructure written in C++*
  - SQLite - *embedded database system*
  - X264 - *is a video encoder in C*
- **Surrogate Used**: CART
- **Techniques compared against**:
  - Siegmund et al.
  - Guo et al.
  - Sarkar et al.
- **Performance Measure:**
  - MRE: Mean Relative Error

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

Techniques compared against



Techniques compared against

2012 — **[Siegmund'12]**

2013 — [Guo'13]

2015 — [Sarkar'15]

Uses Feature Wise heuristics:

- Find
  - a pair of configuration (C_1 and C_2)
  - has same features except for one (F_i)
- Performance score (PS) of F_i
  $PS(F_i) = PS(C_1) - PS(C_2)$
- Performance of a new C_i
  $PS(Ci) = \sum PS(F_i) \ \forall \ F_i \in C_i$

Techniques compared against

2012 — [Siegmund'12]

2013 — [Guo'13]

2015 — [Sarkar'15]

Progressive Sampling Approach:

While terminationCriteria() is True:

- Random Sampling
- Samples in step of |F|
- Build a CART tree

Techniques compared against

2012 [Siegmund'12]

2013 [Guo'13]

2015 **[Sarkar'15]**

Uses Feature Frequencies:

- Projective sampling to decide number of configurations to sample
- Random Sampling
- Build a CART tree

Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations?

RQ 2: Do less data cause larger variances in predicted values?

RQ 3: Can "good" surrogate models (to be used in optimizers) be built using WHAT?

RQ 4: How good is WHAT compared to the state of the art predictors?

RQ1 + RQ2

RQ1 + RQ2 explore

- if WHAT can generate good predictors with low variance
- how much of data should WHAT reflect upon

Comparison between:

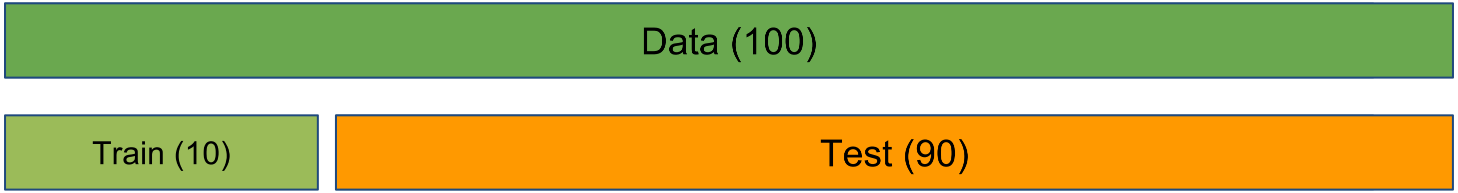
- Baseline (using all the data)
- WHERE + Random
- WHERE + EAST-West
- WHERE + Exemplar

Design of Experiment

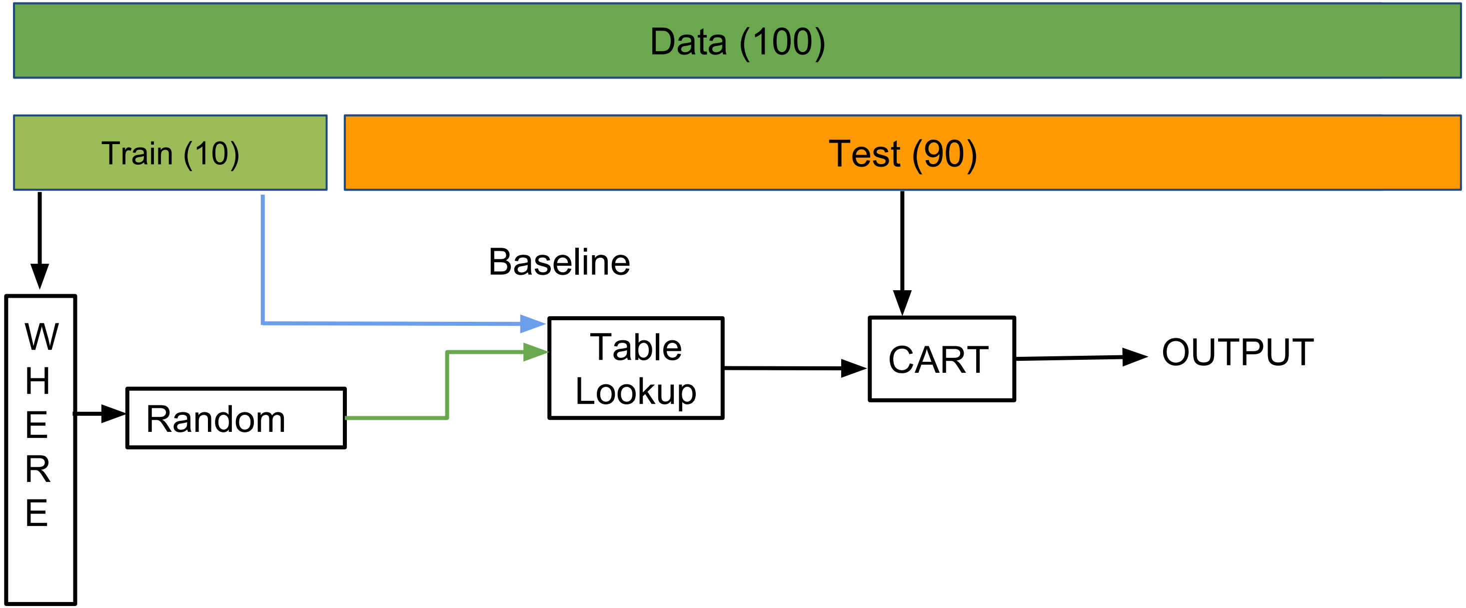
Design of Experiment

Data (100)

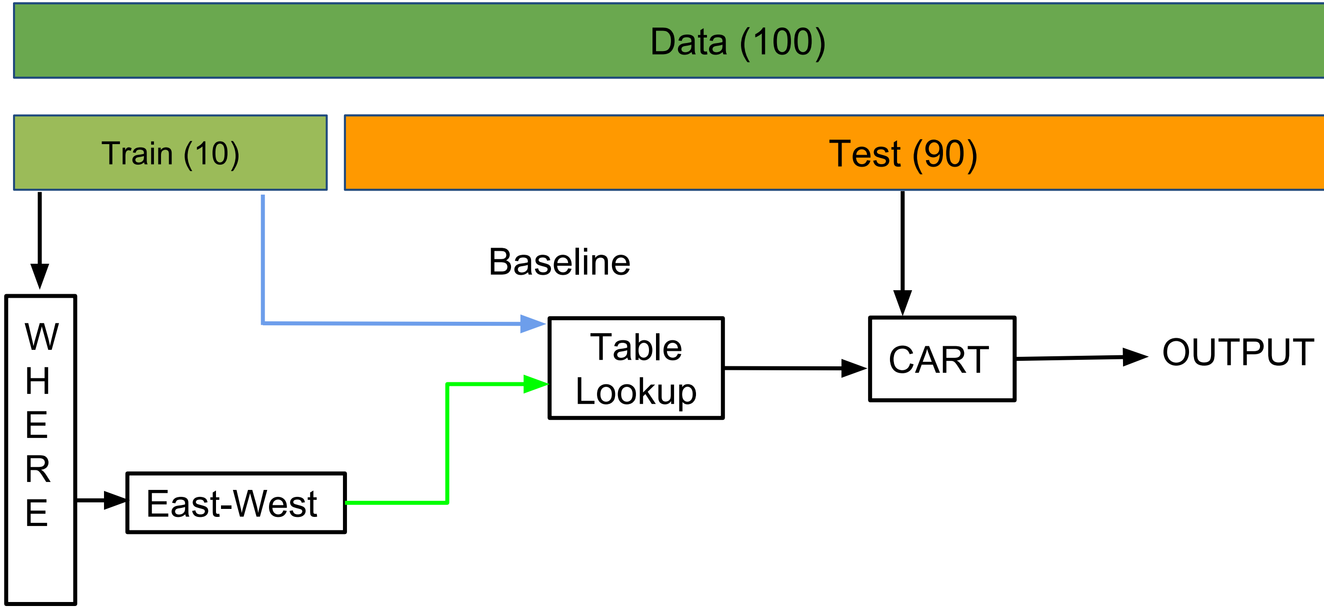
Design of Experiment



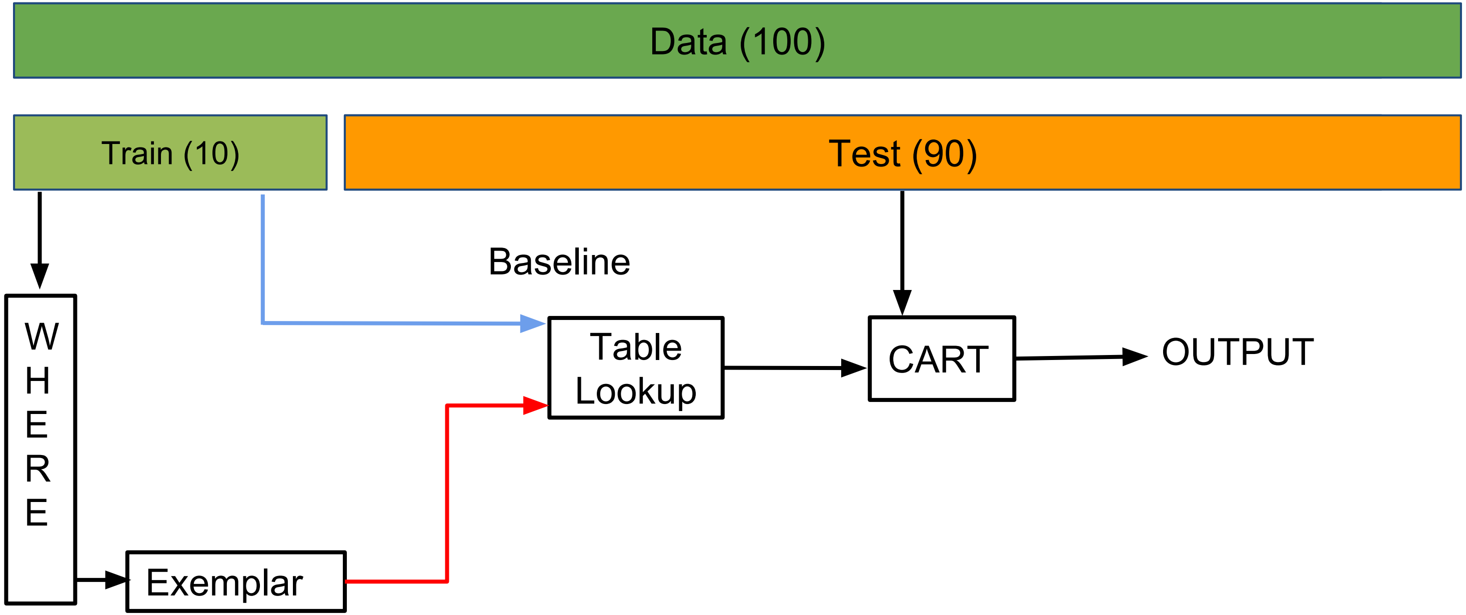
Design of Experiment



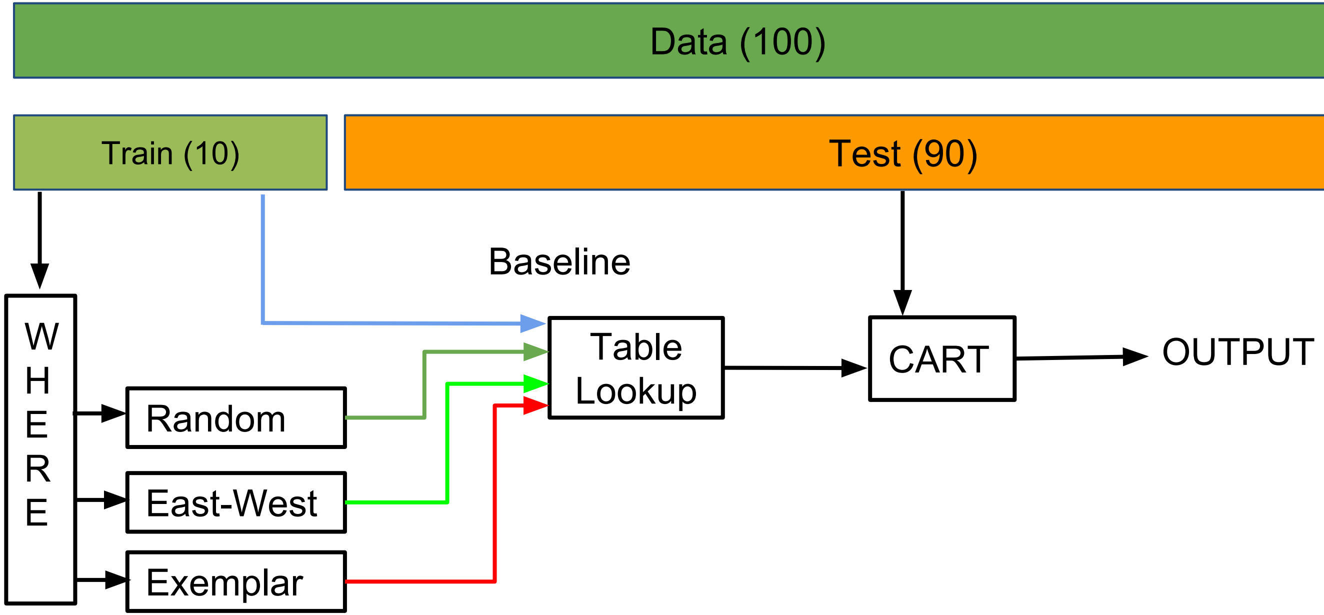
Design of Experiment



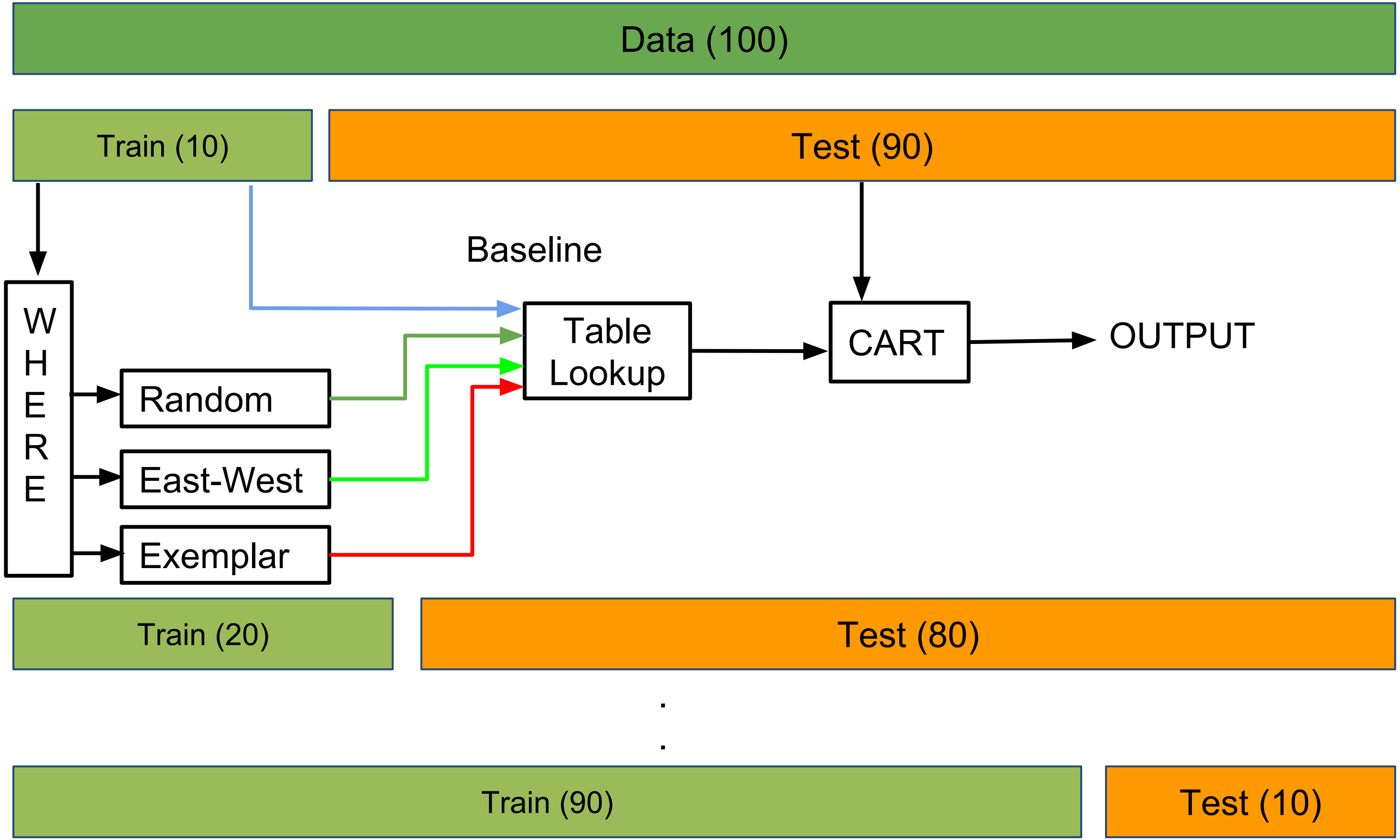
Design of Experiment



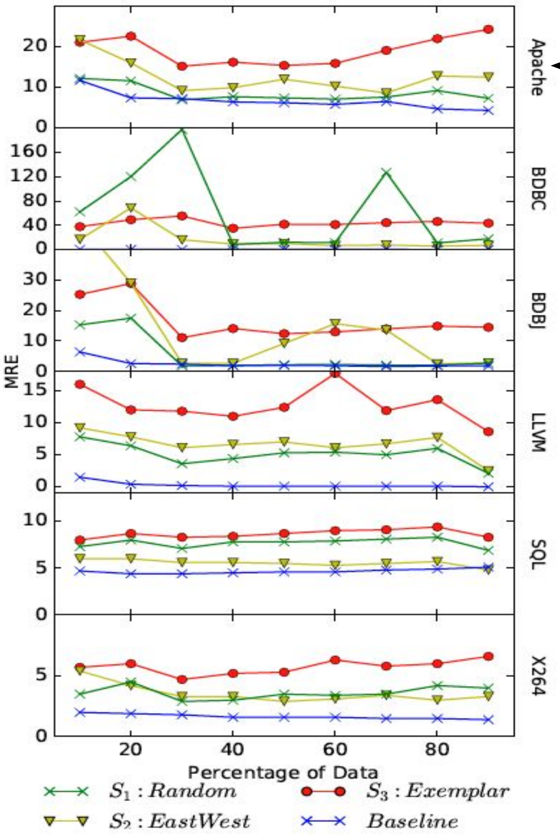
Design of Experiment



Design of Experiment

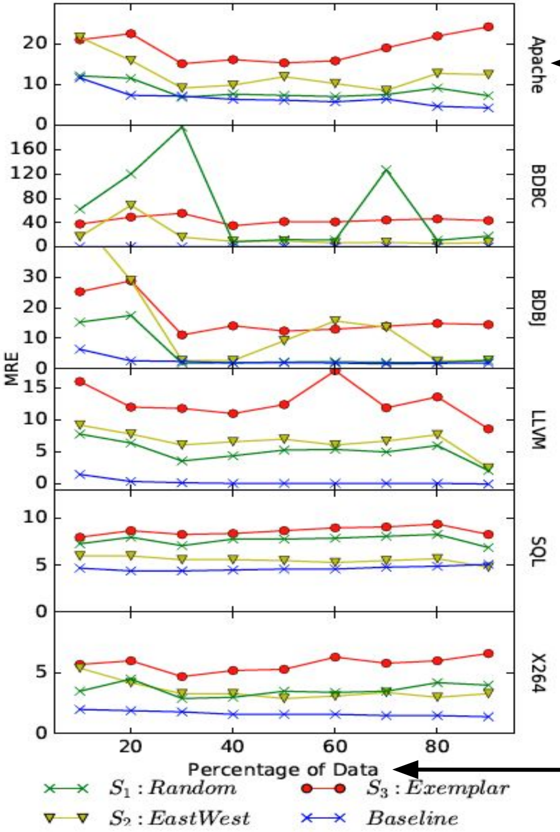


RQ1: Can WHAT generate good predictions using only a small number of configurations?



Software Systems

RQ1: Can WHAT generate good predictions using only a small number of configurations?



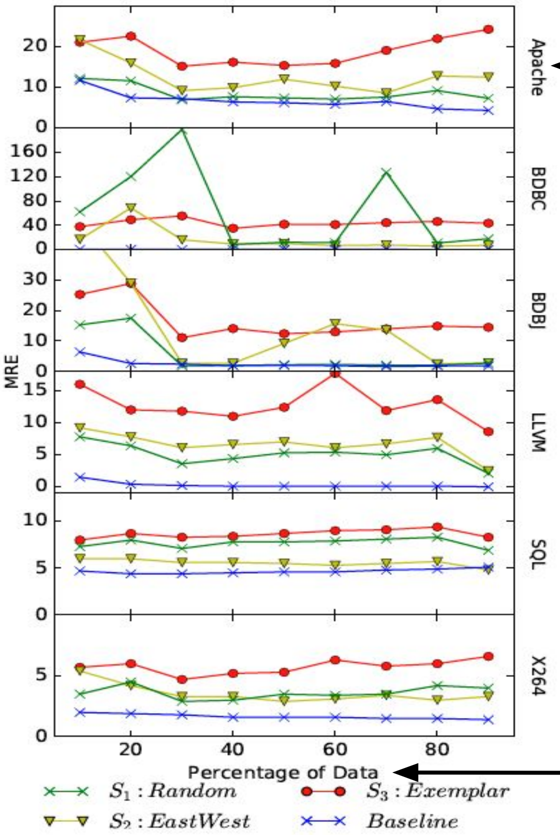
Software Systems

% of data reflected upon

RQ1: Can WHAT generate good predictions using only a small number of configurations?

Software Systems

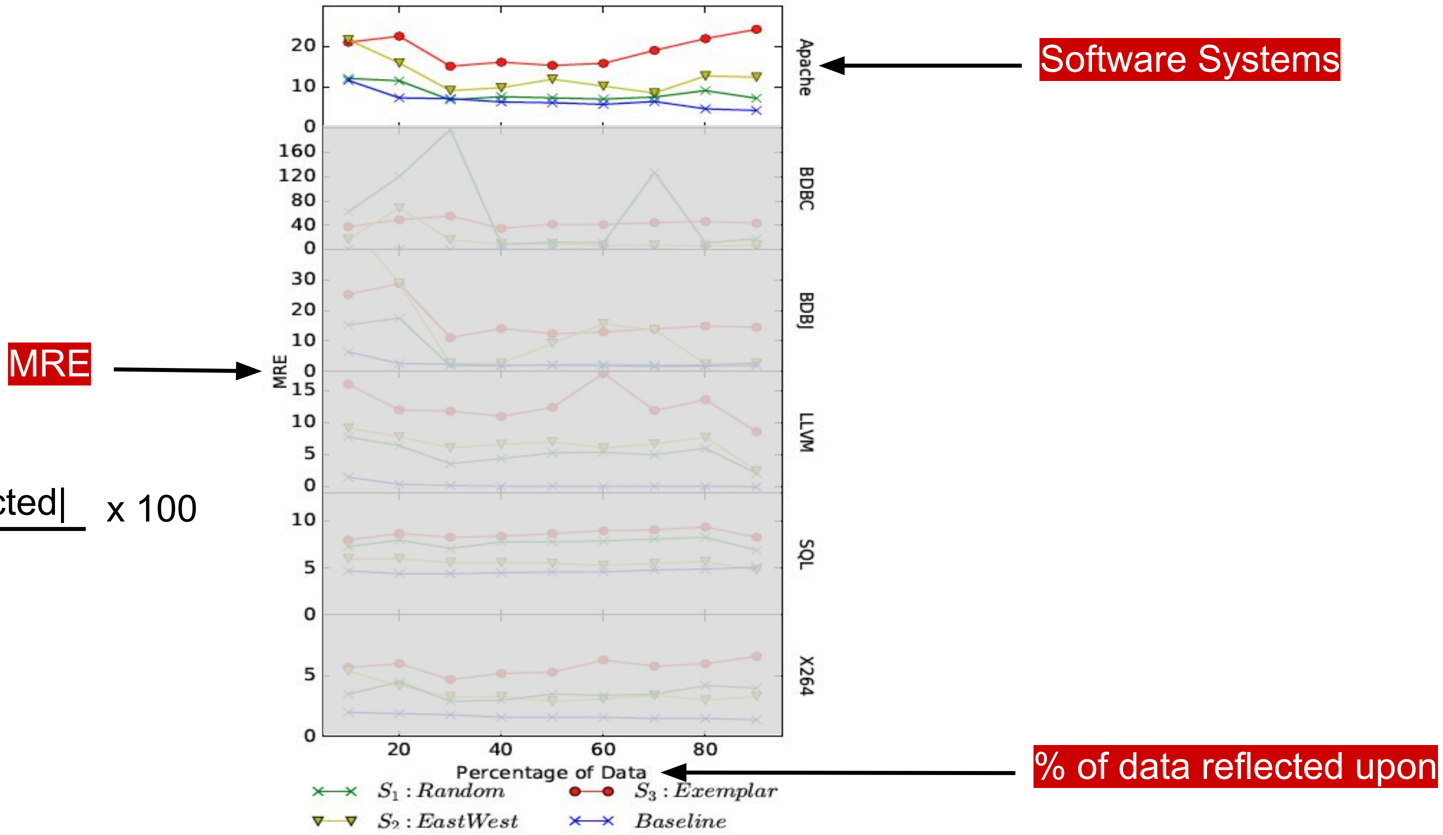
MRE



% of data reflected upon

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

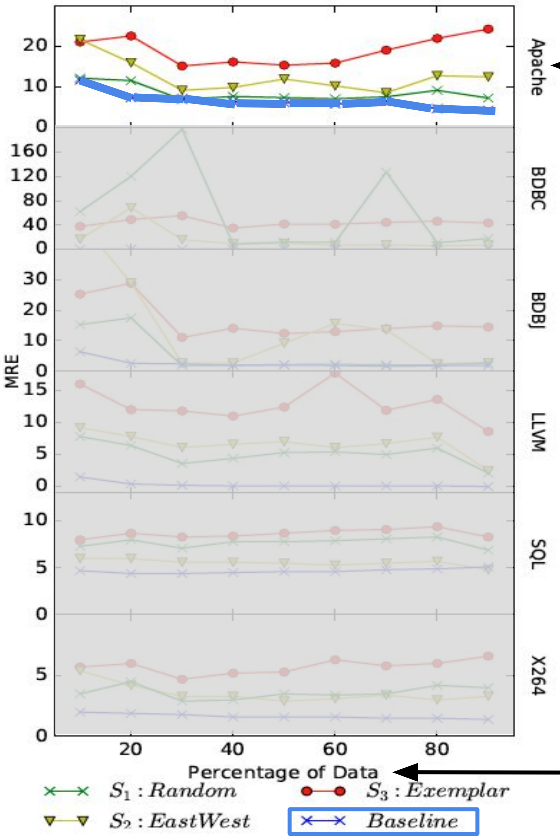
RQ1: Can WHAT generate good predictions using only a small number of configurations?



$$\text{MRE} = \frac{|\text{actual} - \text{predicted}|}{\text{actual}} \times 100$$

RQ1: Can WHAT generate good predictions using only a small number of configurations?

MRE

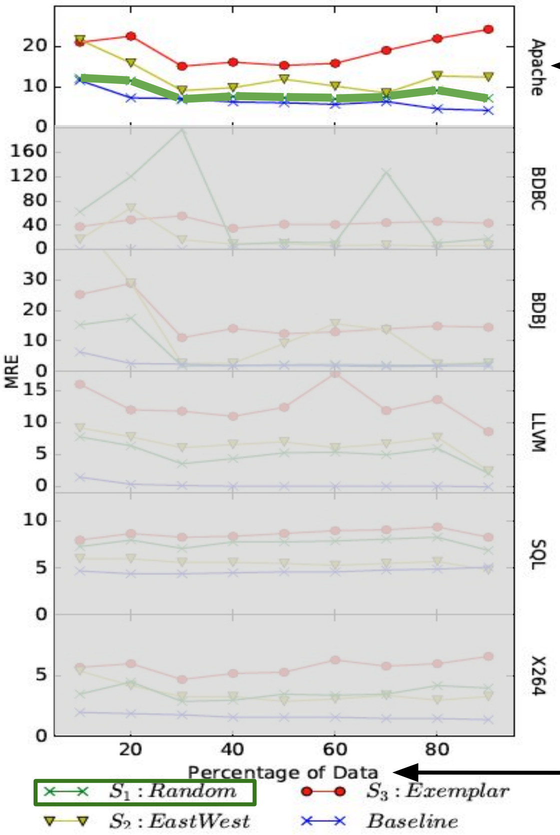


Software Systems

% of data reflected upon

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

RQ1: Can WHAT generate good predictions using only a small number of configurations?



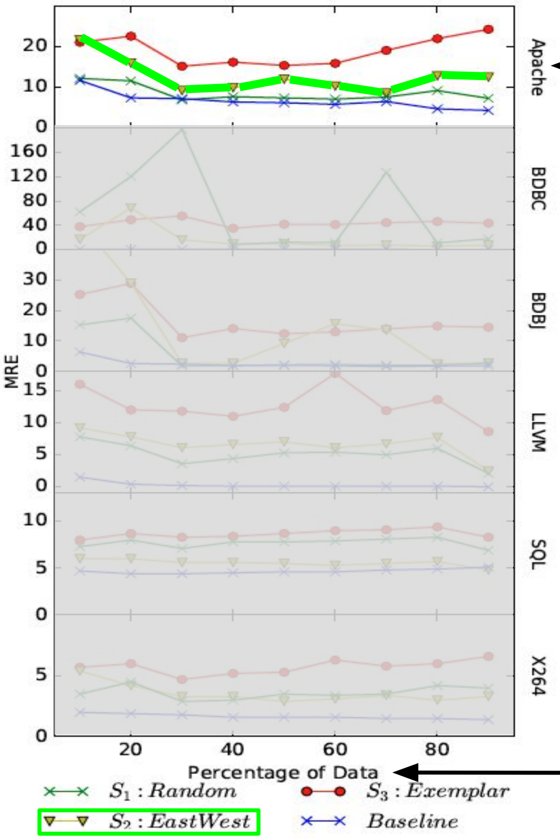
Software Systems

MRE

% of data reflected upon

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

RQ1: Can WHAT generate good predictions using only a small number of configurations?



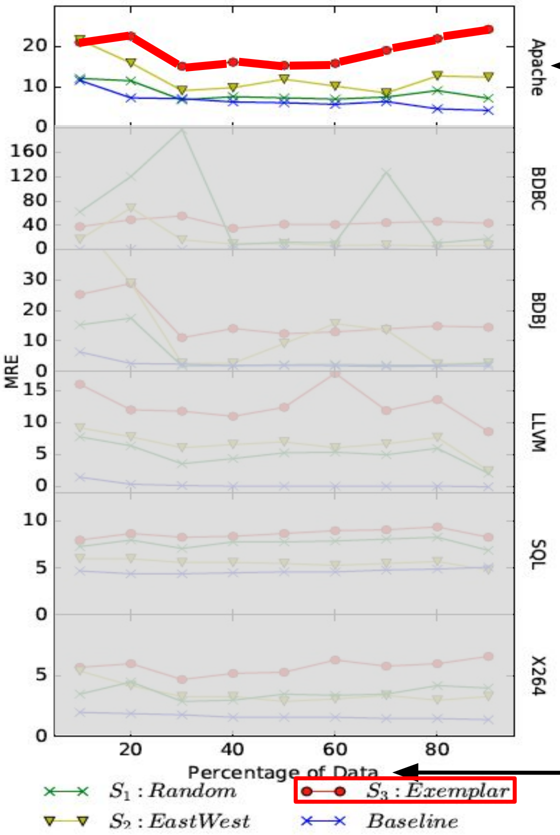
Software Systems

MRE

% of data reflected upon

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

RQ1: Can WHAT generate good predictions using only a small number of configurations?



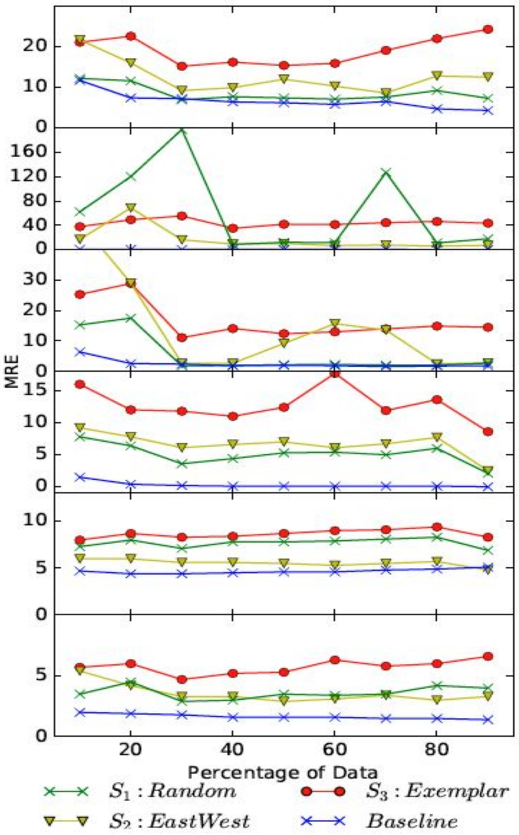
Software Systems

MRE

% of data reflected upon

$$MRE = \frac{|actual - predicted|}{actual} \times 100$$

RQ1: Can WHAT generate good predictions using only a small number of configurations?

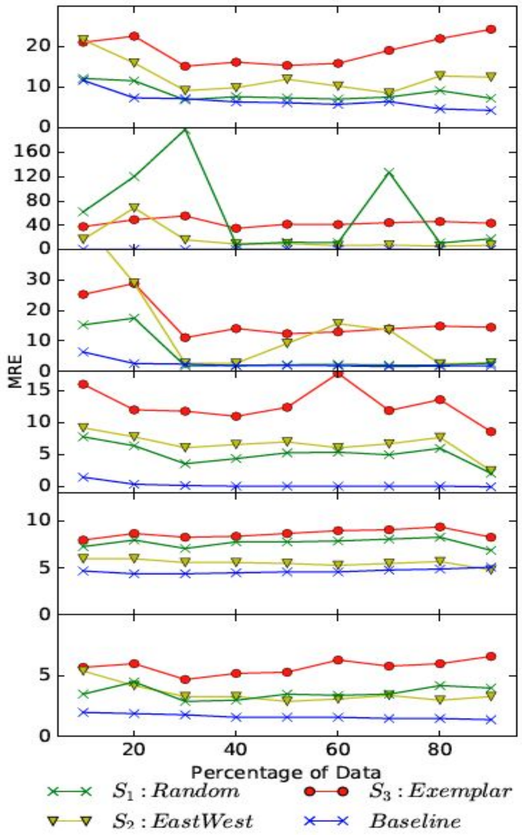


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	?	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	?	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	?	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

RQ1: Can WHAT generate good predictions using only a small number of configurations?

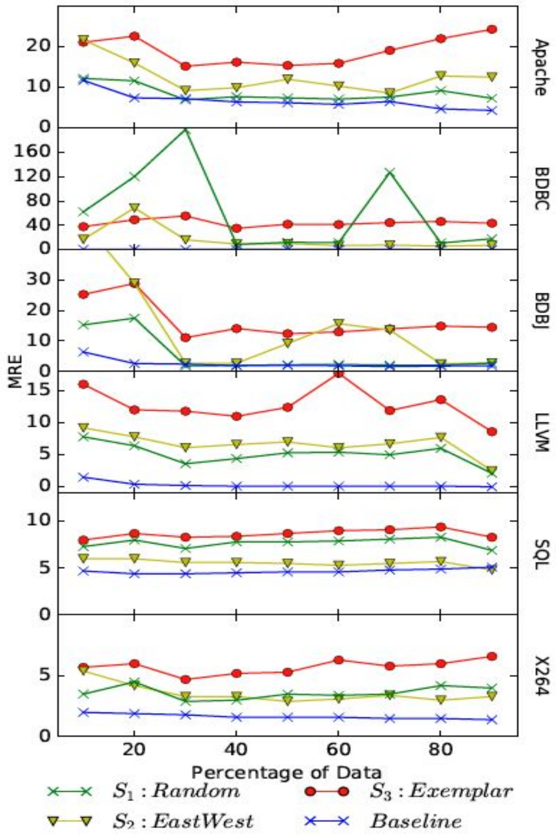


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✕	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✕	?	?	?	?	?
Standard Deviation	?	?	?	?	?	?

RQ1: Can WHAT generate good predictions using only a small number of configurations?

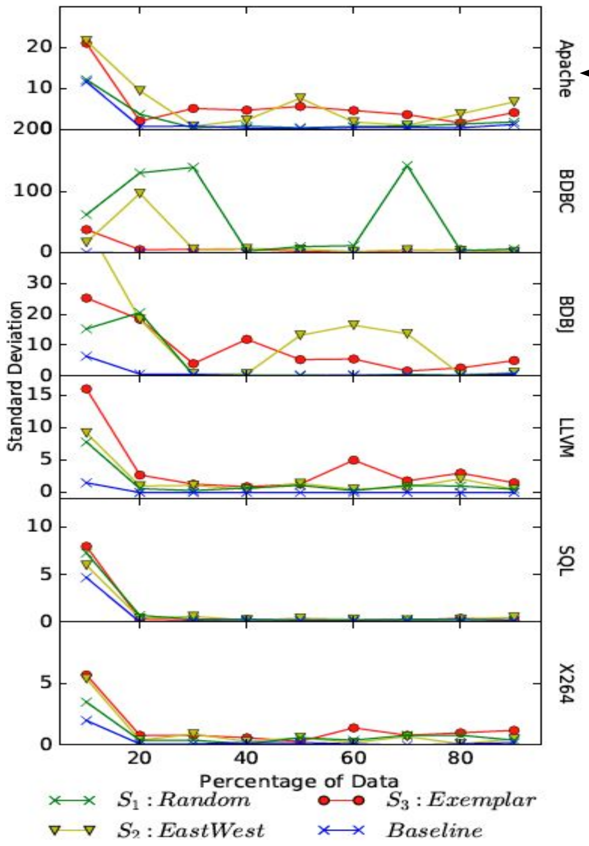


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✗	✔	✔	✗	✔
Standard Deviation	?	?	?	?	?	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✔	✗	✗	✔	✔
Standard Deviation	?	?	?	?	?	?

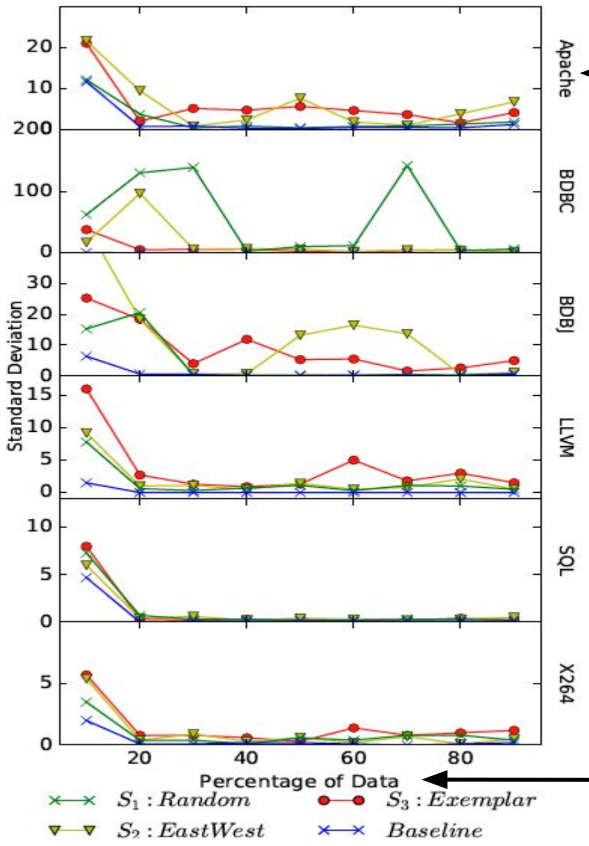
Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✗	✗	✗	✗	✗
Standard Deviation	?	?	?	?	?	?

RQ2: Do less data cause larger variances in predicted values?



Software Systems

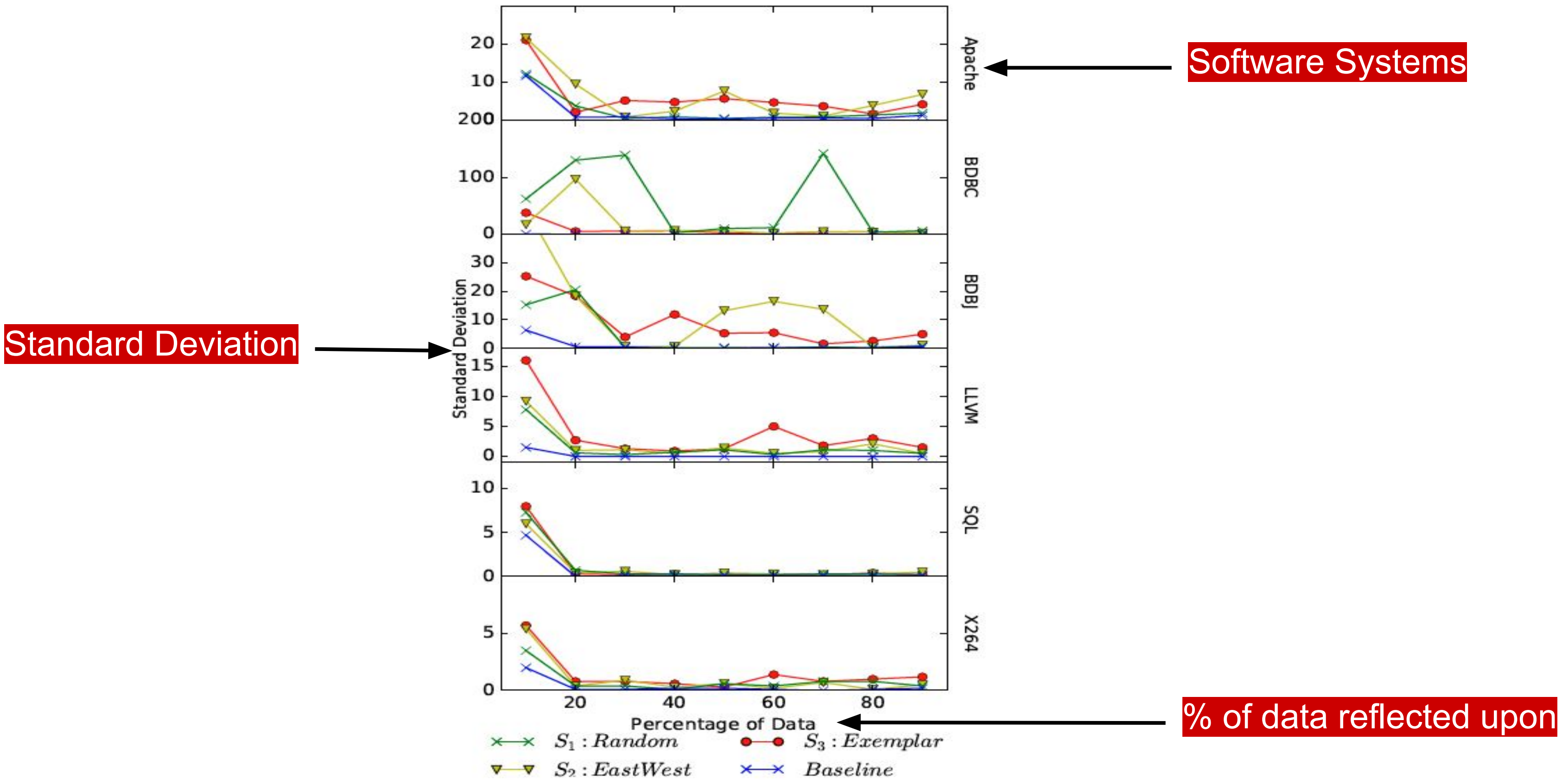
RQ2: Do less data cause larger variances in predicted values?



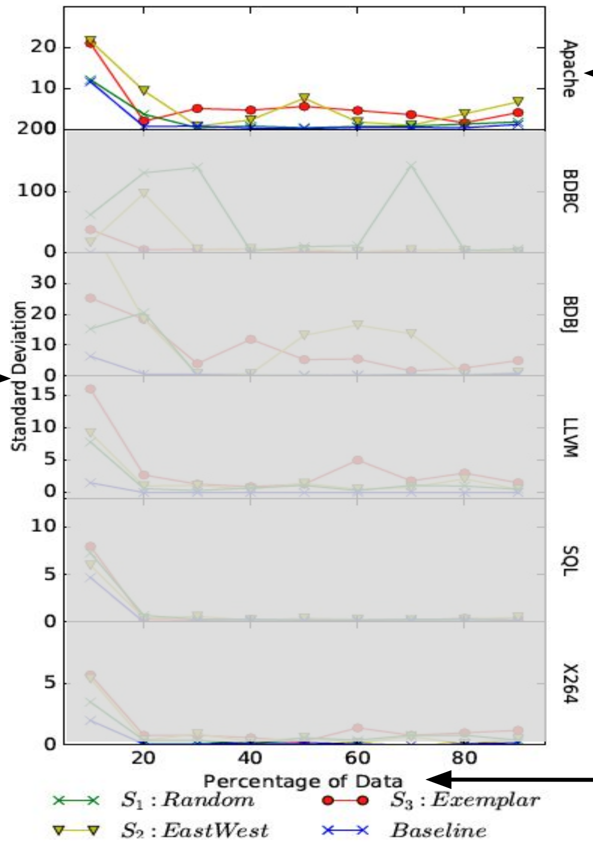
Software Systems

% of data reflected upon

RQ2: Do less data cause larger variances in predicted values?



RQ2: Do less data cause larger variances in predicted values?

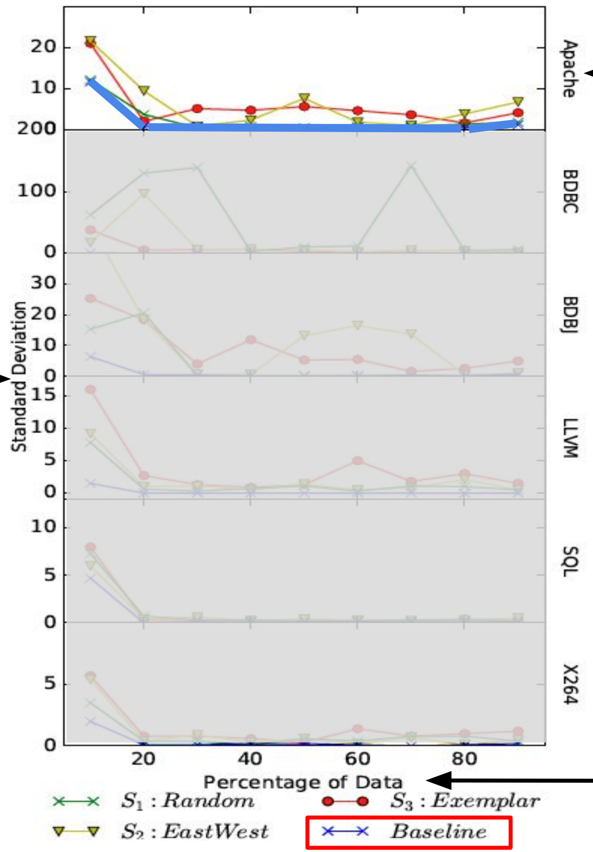


Software Systems

Standard Deviation

% of data reflected upon

RQ2: Do less data cause larger variances in predicted values?

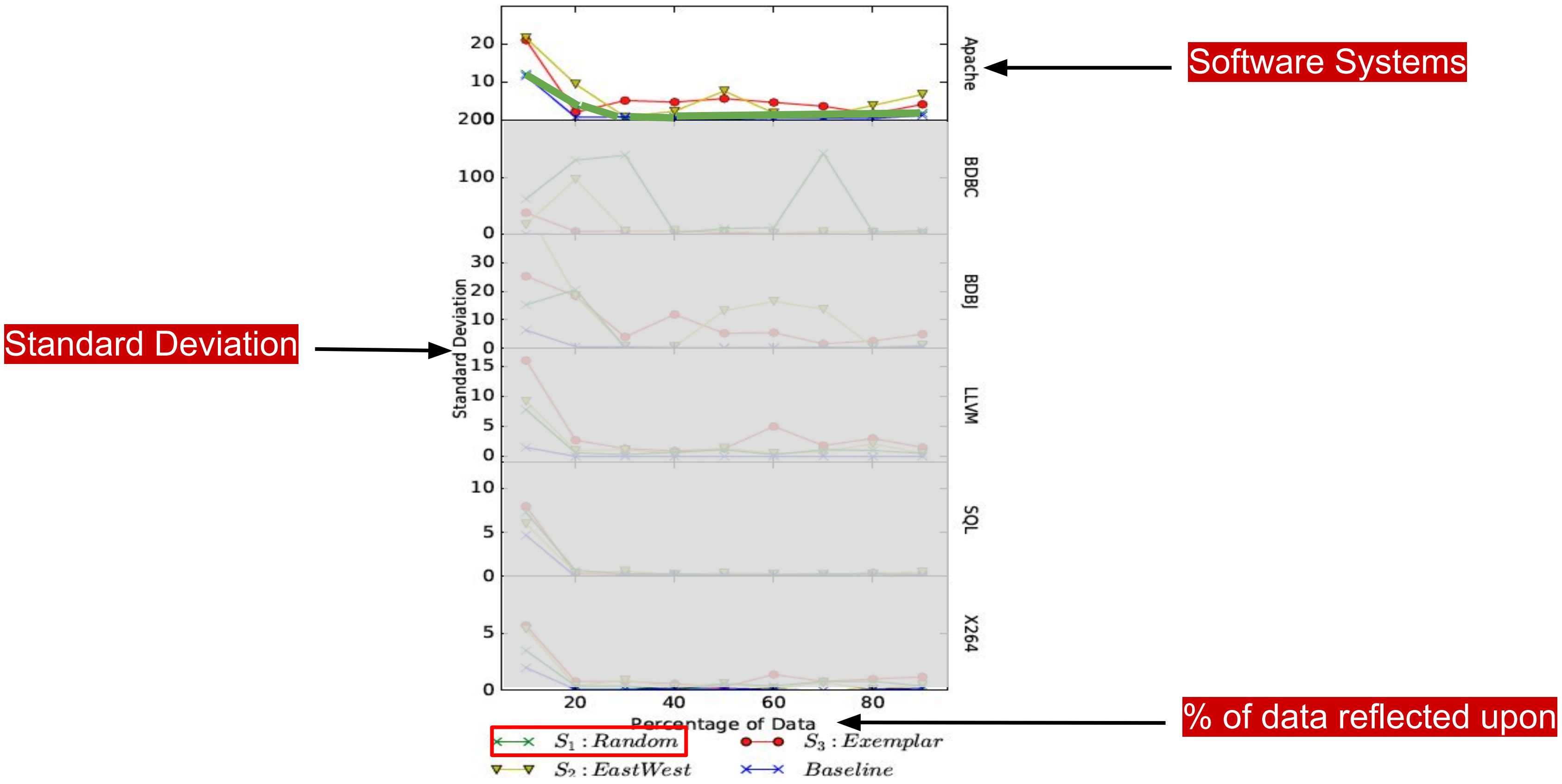


Standard Deviation

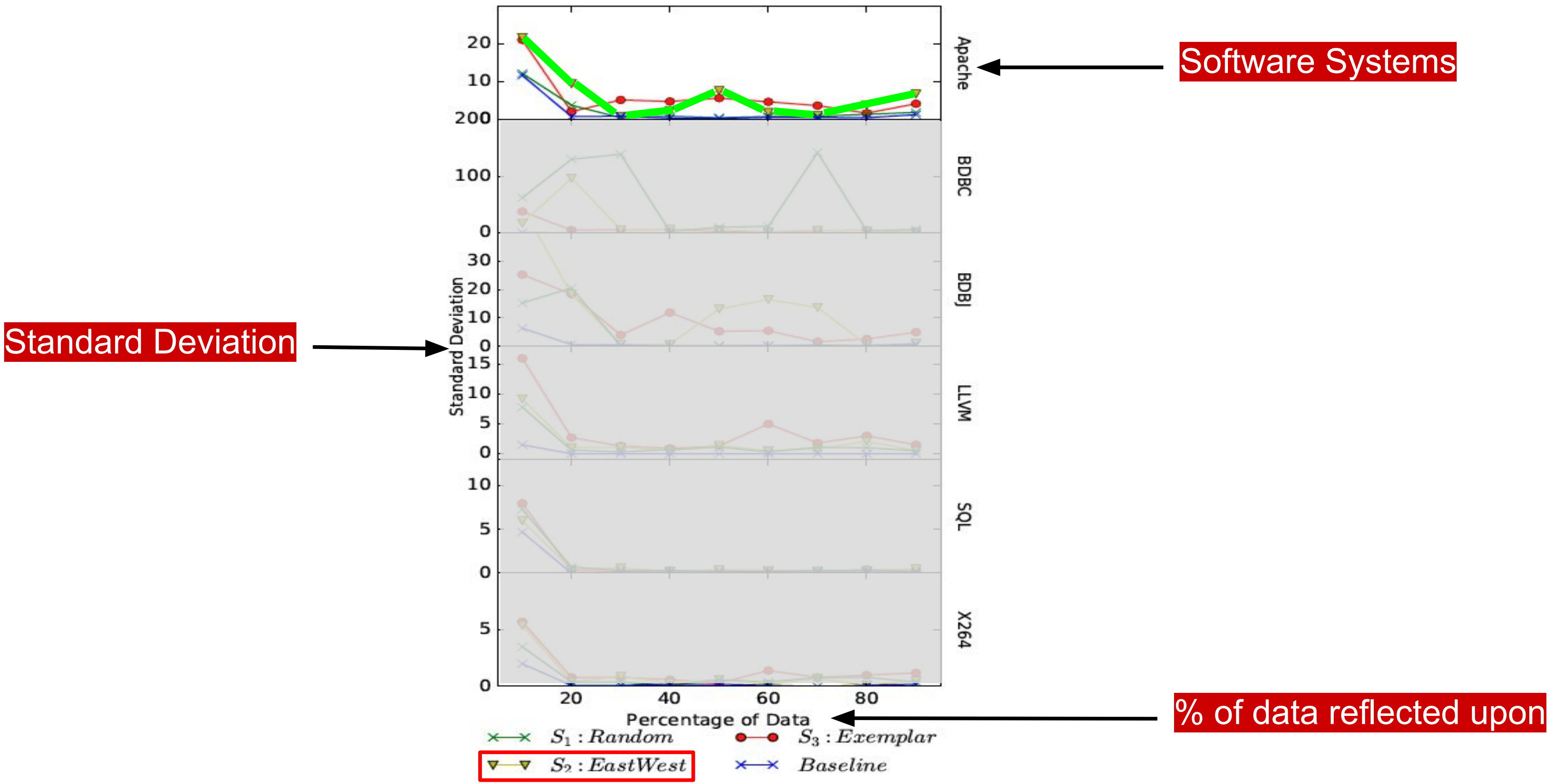
Software Systems

% of data reflected upon

RQ2: Do less data cause larger variances in predicted values?



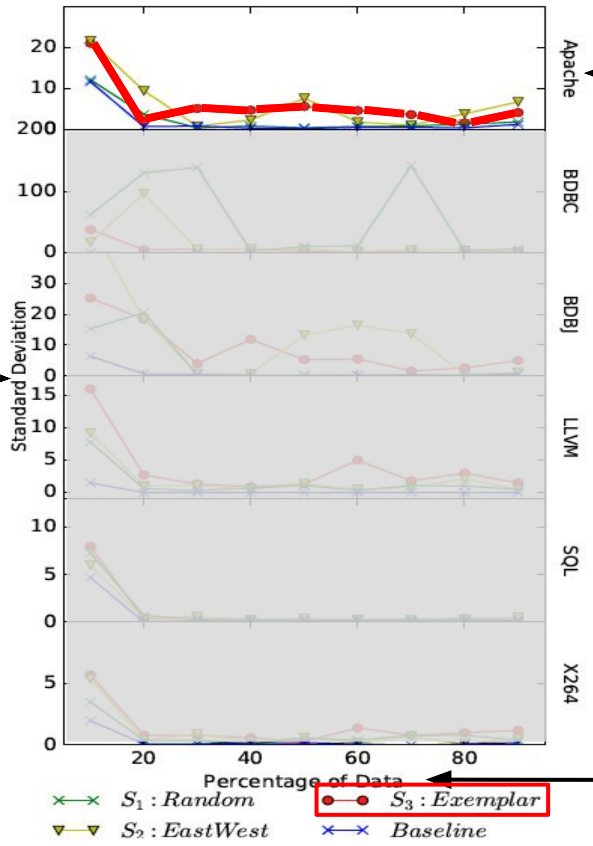
RQ2: Do less data cause larger variances in predicted values?



RQ2: Do less data cause larger variances in predicted values?

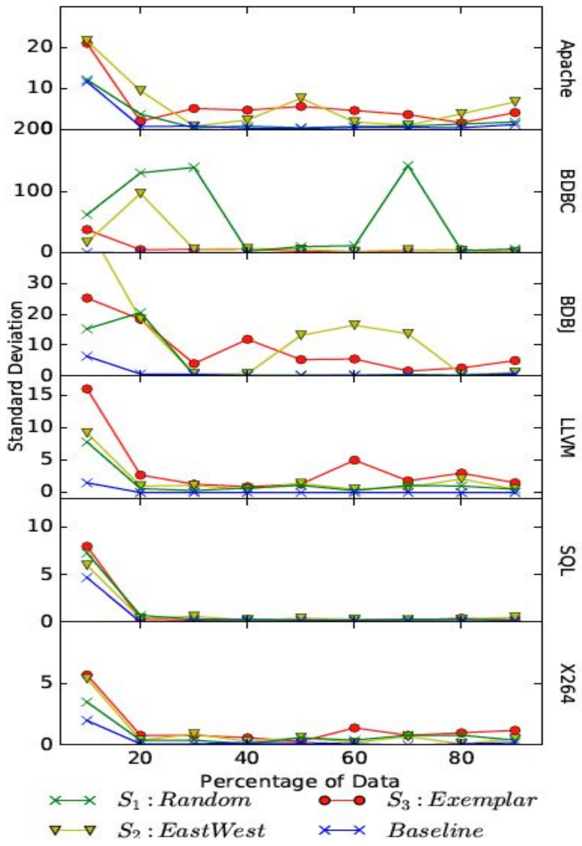
Software Systems

Standard Deviation



% of data reflected upon

RQ2: Do less data cause larger variances in predicted values?

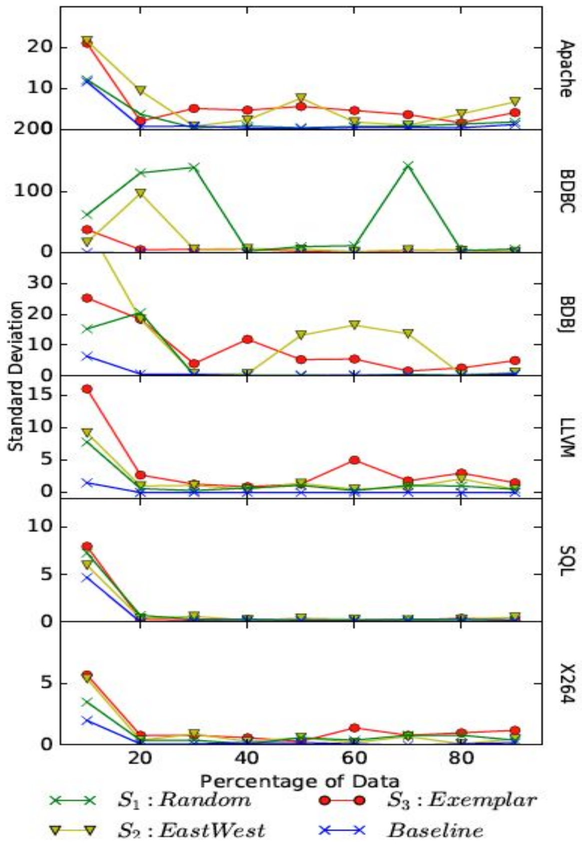


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✘	✔	✔	✘	✔
Standard Deviation	?	?	?	?	?	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✔	✘	✘	✔	✔
Standard Deviation	?	?	?	?	?	?

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✘	✘	✘	✘	✘
Standard Deviation	?	?	?	?	?	?

RQ2: Do less data cause larger variances in predicted values?

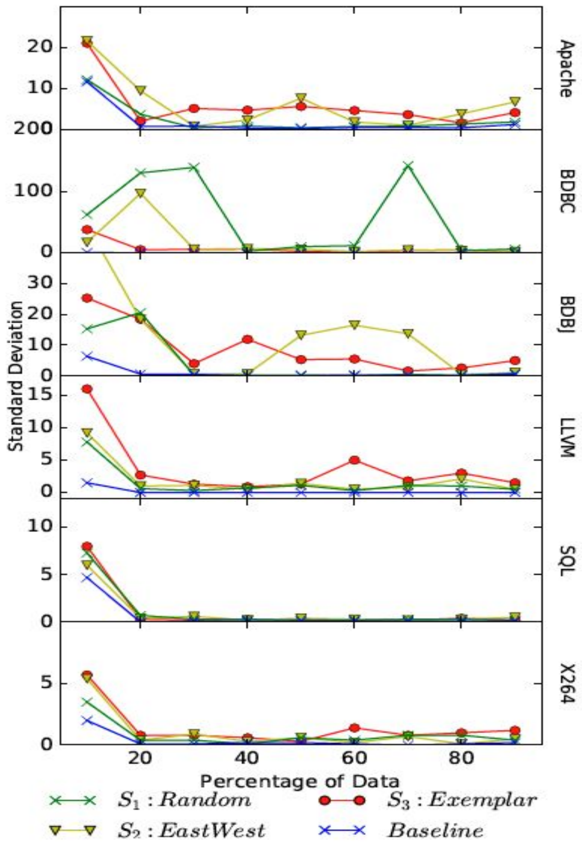


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✗	✔	✔	✗	✔
Standard Deviation	?	✗	?	?	?	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✔	✗	✗	✔	✔
Standard Deviation	?	✔	?	?	?	?

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✗	✗	✗	✗	✗
Standard Deviation	?	✔	?	?	?	?

RQ2: Do less data cause larger variances in predicted values?

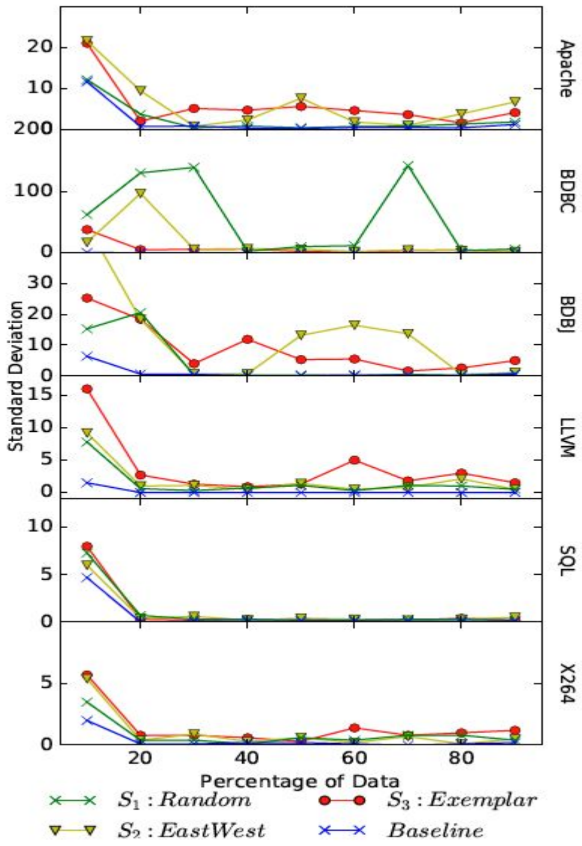


Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✘	✔	✔	✘	✔
Standard Deviation	?	✘	?	?	✔	?

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✔	✘	✘	✔	✔
Standard Deviation	?	✔	?	?	✔	?

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✘	✘	✘	✘	✘
Standard Deviation	?	✔	?	?	✔	?

RQ2: Do less data cause larger variances in predicted values?



Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✘	✔	✔	✘	✔
Standard Deviation	✔	✘	✔	✔	✔	✔

East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✔	✘	✘	✔	✔
Standard Deviation	✘	✔	✘	✔	✔	✔

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✘	✘	✘	✘	✘	✘
Standard Deviation	✘	✔	✘	✘	✔	✔

RQ1 + RQ2: Observations

- Baseline results is the best
  - It uses 100% of data
- Results plateaued after **40%**
- WHERE + Exemplar
  - largest Mean MRE
  - **Not Recommended**
- WHERE + East-West
  - MRE 3/6 times better/similar
  - Standard deviation is low
  - **Recommended**
- WHERE + Random
  - MRE 4/6 times better/similar
  - Standard deviation is low
  - **Recommended**

Random						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✔	✗	✔	✔	✗	✔
Standard Deviation	✔	✗	✔	✔	✔	✔

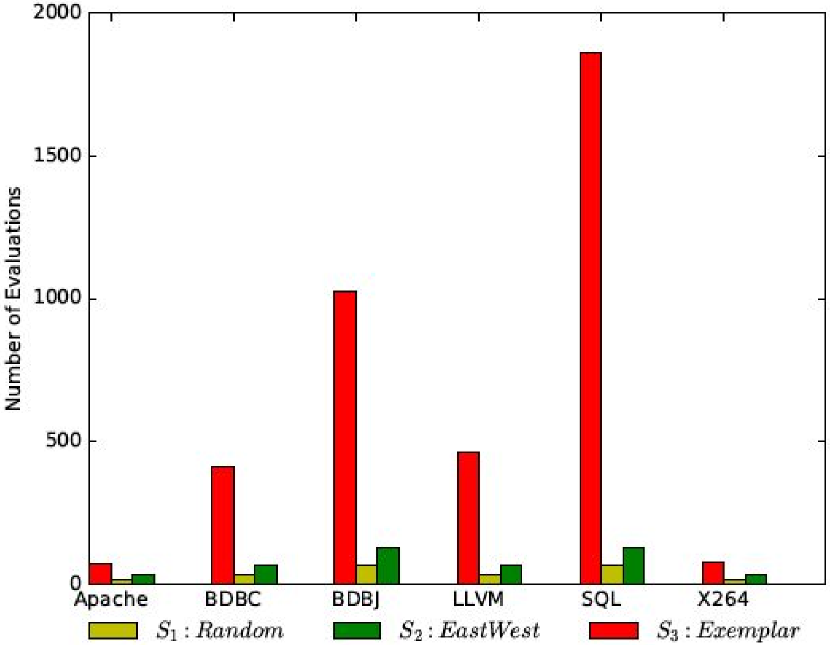
East-West						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✔	✗	✗	✔	✔
Standard Deviation	✗	✔	✗	✔	✔	✔

Exemplar						
Software System	Apache	BDBC	BDBJ	LLVM	SQLite	X264
Mean MRE	✗	✗	✗	✗	✗	✗
Standard Deviation	✗	✔	✗	✗	✔	✔

RQ1 + RQ2: Evaluation

- WHERE + East-West
  - MRE 3/6 times better/similar
  - Standard deviation is low
  - **Recommended**

- WHERE + Random
  - MRE 4/6 times better/similar
  - Standard deviation is low
  - **Recommended**



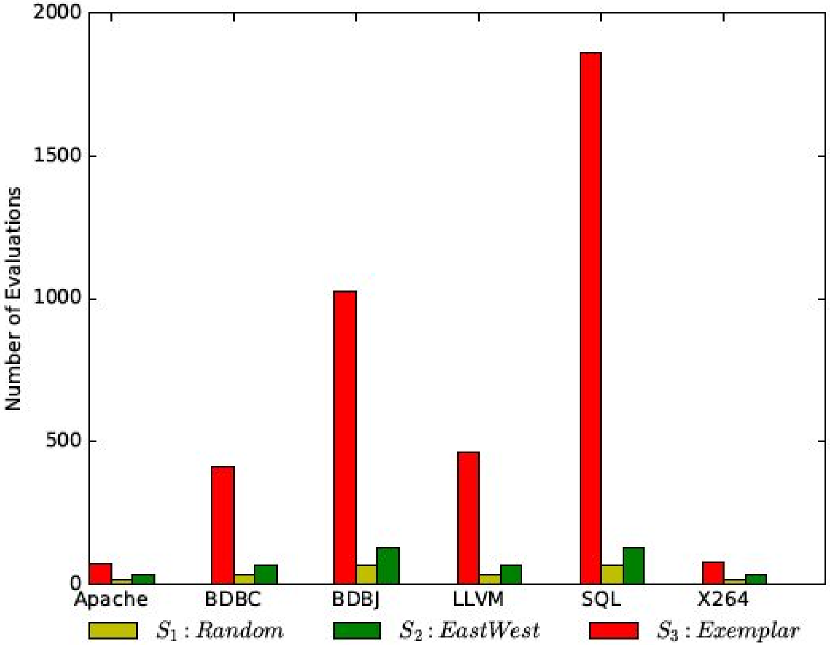
Software Systems

RQ1 + RQ2: Evaluation

- WHERE + East-West
    - MRE 3/6 times better/similar
    - Standard deviation is low
    - **Recommended**
- WHERE + Random
    - MRE 4/6 times better/similar
    - Standard deviation is low
    - **Recommended**

# of Evaluations

(When Training Data = 40%)



Software Systems

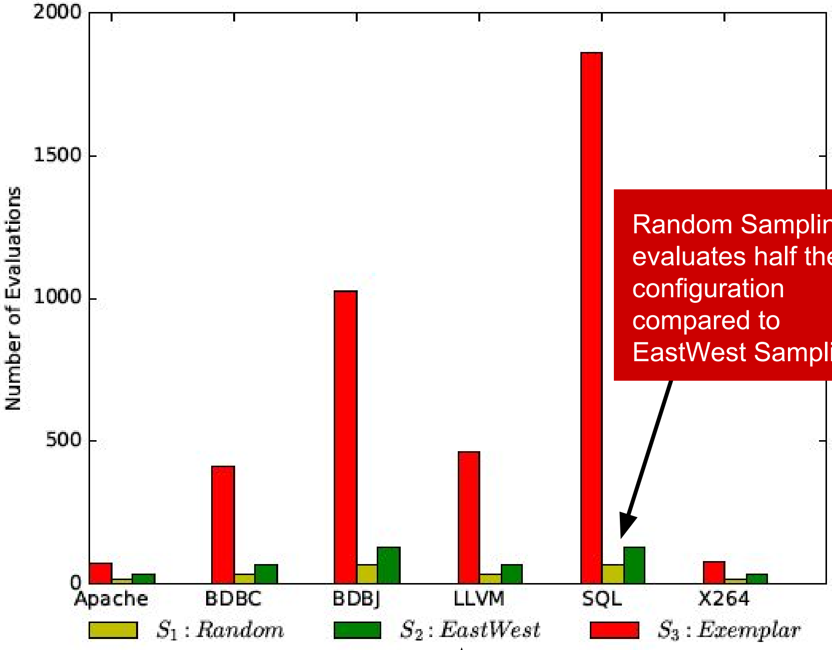
RQ1 + RQ2: Evaluation

- WHERE + East-West
  - MRE 3/6 times better/similar
  - Standard deviation is low
  - ~~**Recommended**~~

# of Evaluations

(When Training Data = 40%)

- WHERE + Random
  - MRE 4/6 times better/similar
  - Standard deviation is low
  - **Recommended**



Random Sampling evaluates half the configuration compared to EastWest Sampling

Software Systems

RQ1 + RQ2: Evaluation

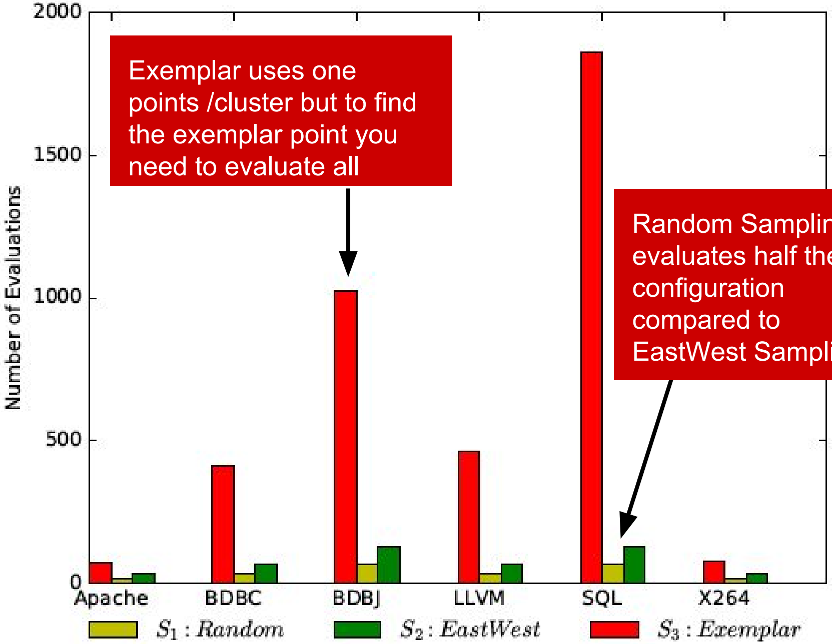
- WHERE + East-West
  - MRE 3/6 times better/similar
  - Standard deviation is low
  - **Recommended**
- WHERE + Random
  - MRE 4/6 times better/similar
  - Standard deviation is low
  - **Recommended**

# of Evaluations

(When Training Data = 40%)

Exemplar uses one points /cluster but to find the exemplar point you need to evaluate all

Random Sampling evaluates half the configuration compared to EastWest Sampling



Software Systems

RQ 3: Can "good" surrogate models (to be used in optimizers) be built using WHAT?

RQ 3 explore

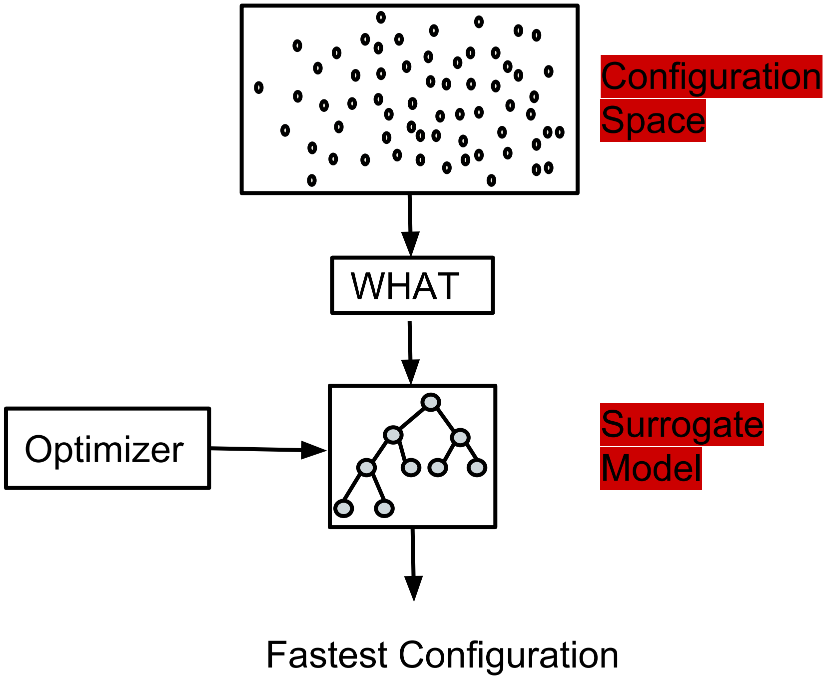
- if predictors generated using samples from WHAT can find faster performance scores (eg. Response time)

Optimization Goal

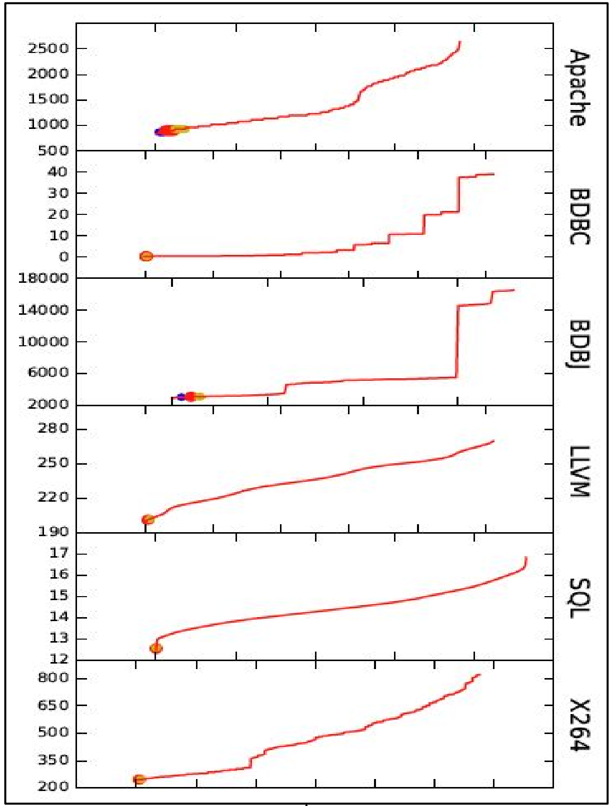
- Minimize the performance score of the system

Comparison between:

- GALE [Krall'15]
- DE [Storn'95]
- NSGA-II [Deb'02]



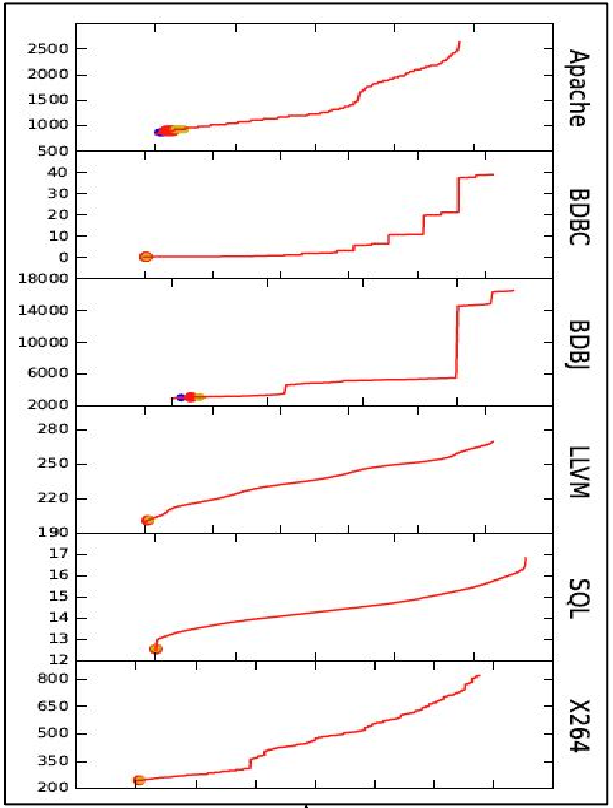
RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?



Instances sorted based on Performance Scores

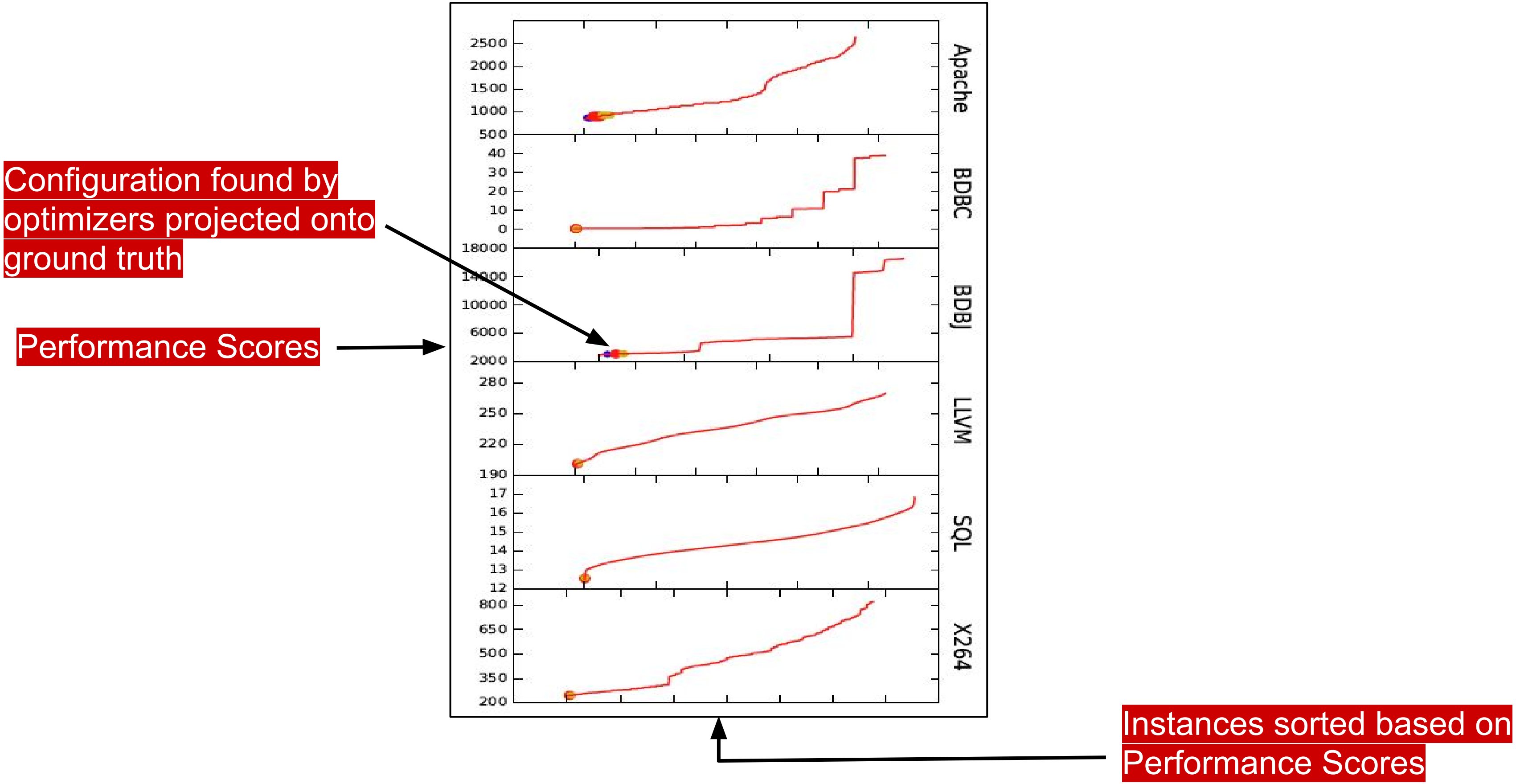
RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

Performance Scores



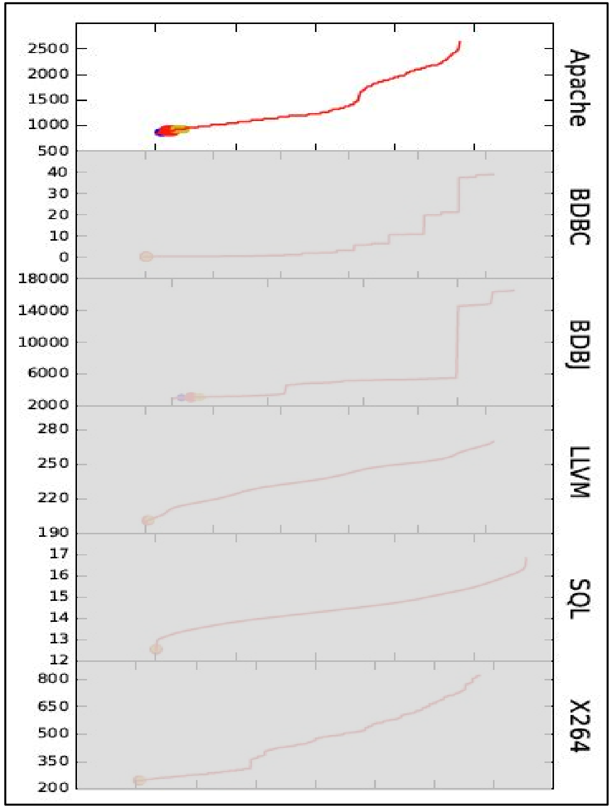
Instances sorted based on Performance Scores

RQ 3: Can "good" surrogate models (to be used in optimizers) be built using WHAT?



RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

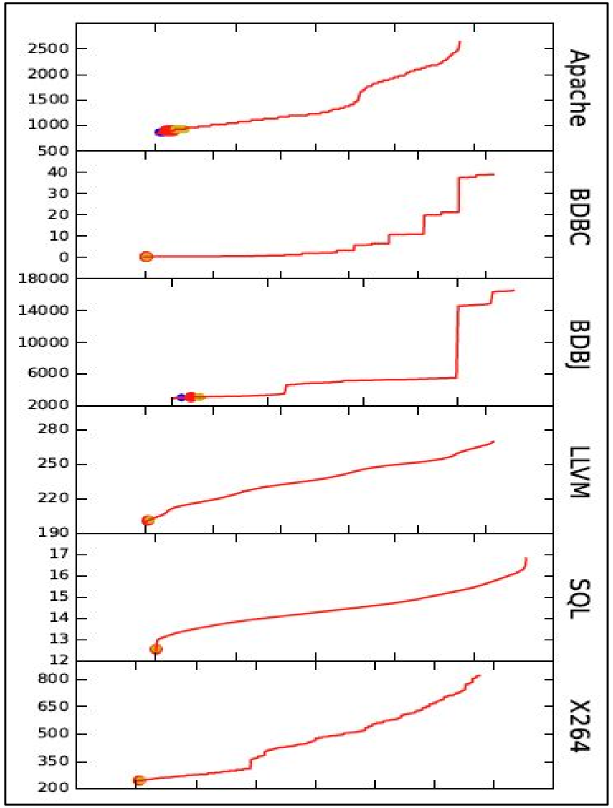
- Optimization Goal: Minimization



RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

- Optimization Goal: Minimization

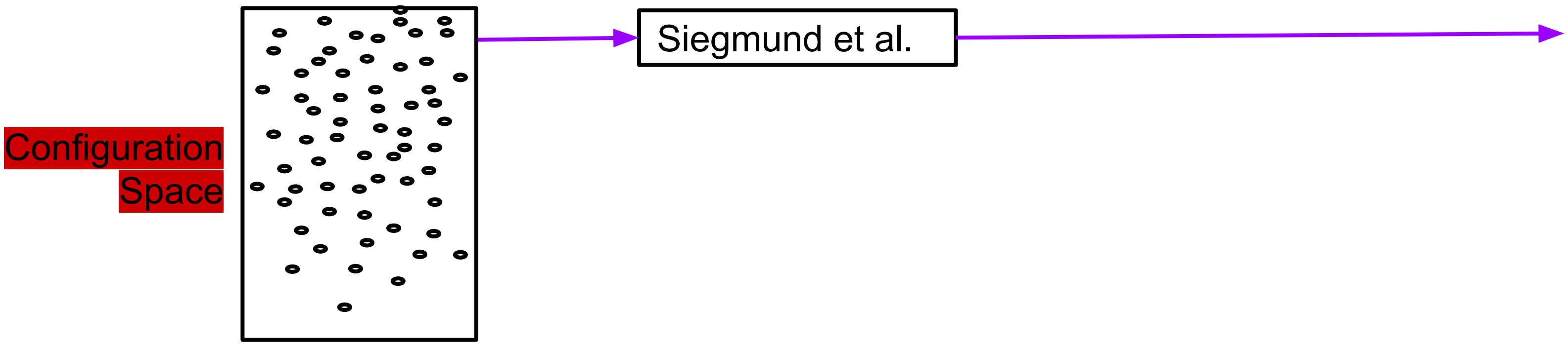
- Optimized configurations
  - within 1% of the fastest configuration



RQ 4: How good is WHAT compared to the state of the art predictors?

RQ 4 explores

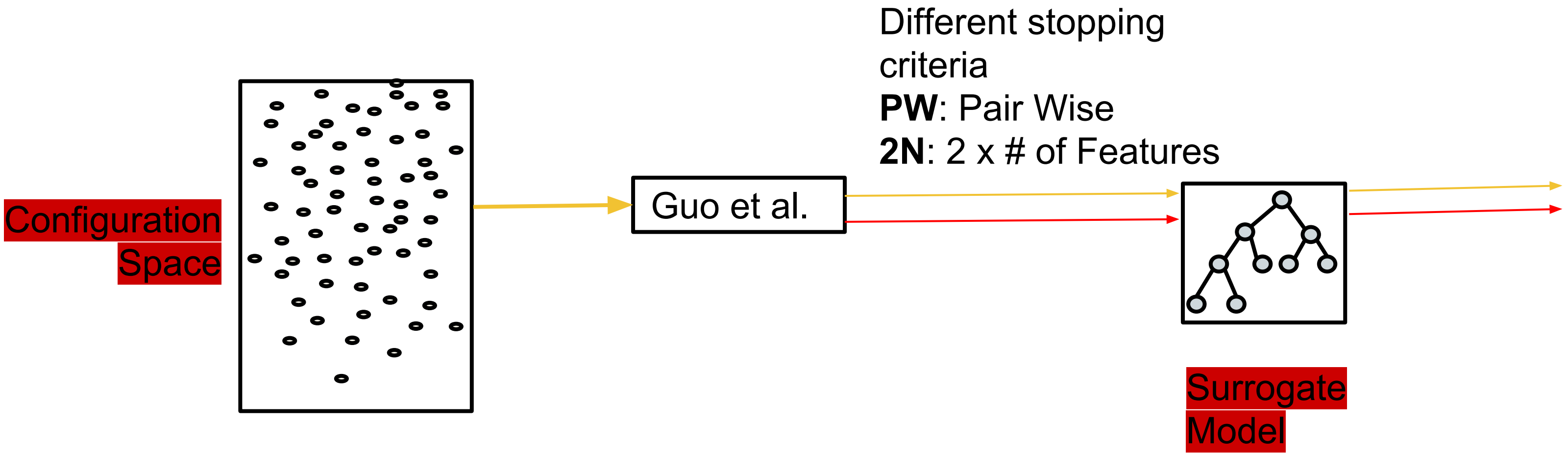
- If WHAT is better than state-of-the-art techniques
  - Siegmund et al. - FW heuristics
  - Guo et al. - Progressive Sampling
  - Sarkar et al. - Random Sampling + Feature-wise heuristics



RQ 4: How good is WHAT compared to the state of the art predictors?

RQ 4 explores

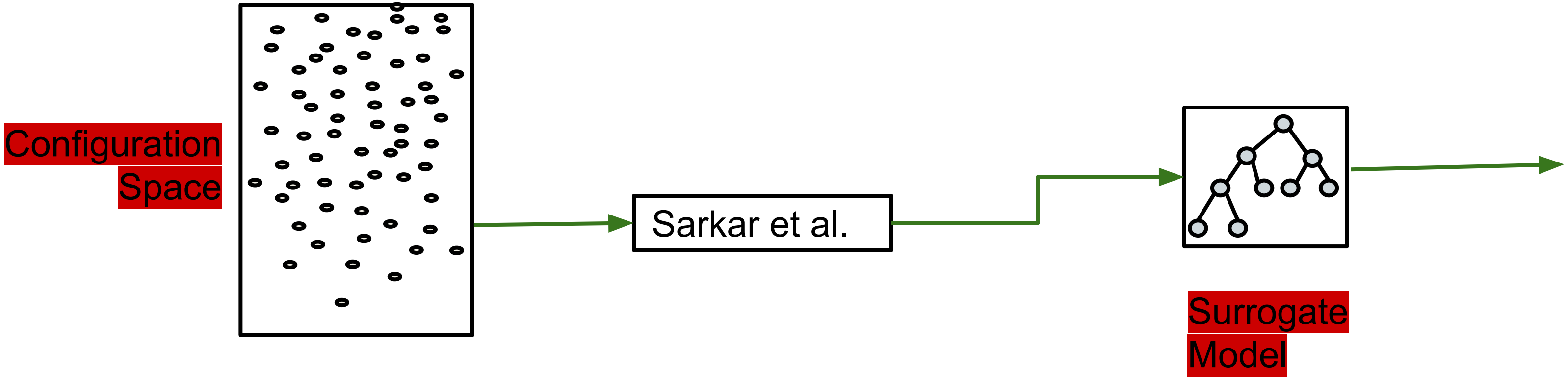
- If WHAT is better than state-of-the-art techniques
  - Siegmund et al. - FW heuristics
  - Guo et al. - Progressive Sampling
  - Sarkar et al. - Random Sampling + Feature-wise heuristics



RQ 4: How good is WHAT compared to the state of the art predictors?

RQ 4 explores

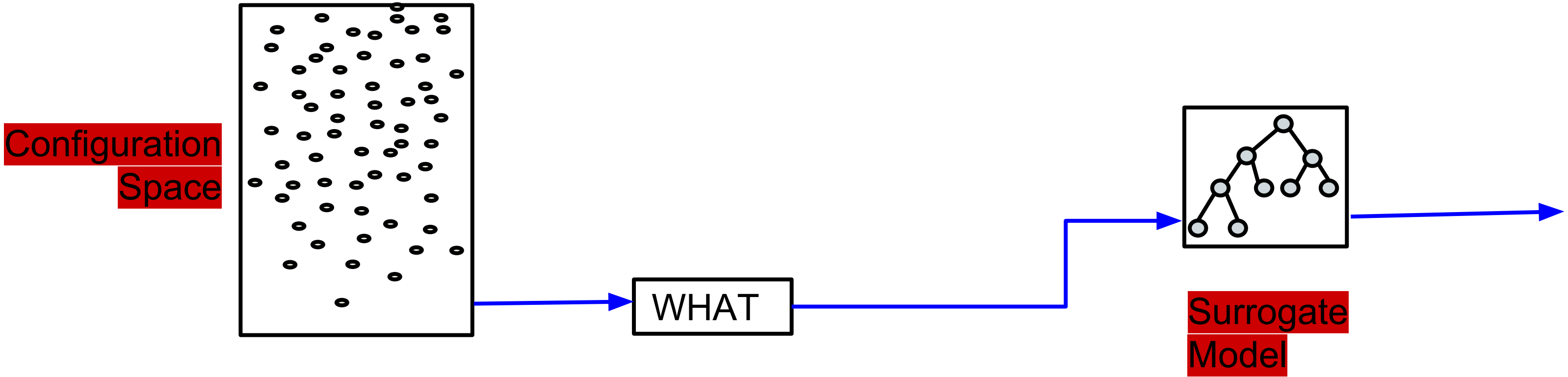
- If WHAT is better than state-of-the-art techniques
  - Siegmund et al. - FW heuristics
  - Guo et al. - Progressive Sampling
  - Sarkar et al. - Random Sampling + Feature-wise heuristics



RQ 4: How good is WHAT compared to the state of the art predictors?

RQ 4 explores

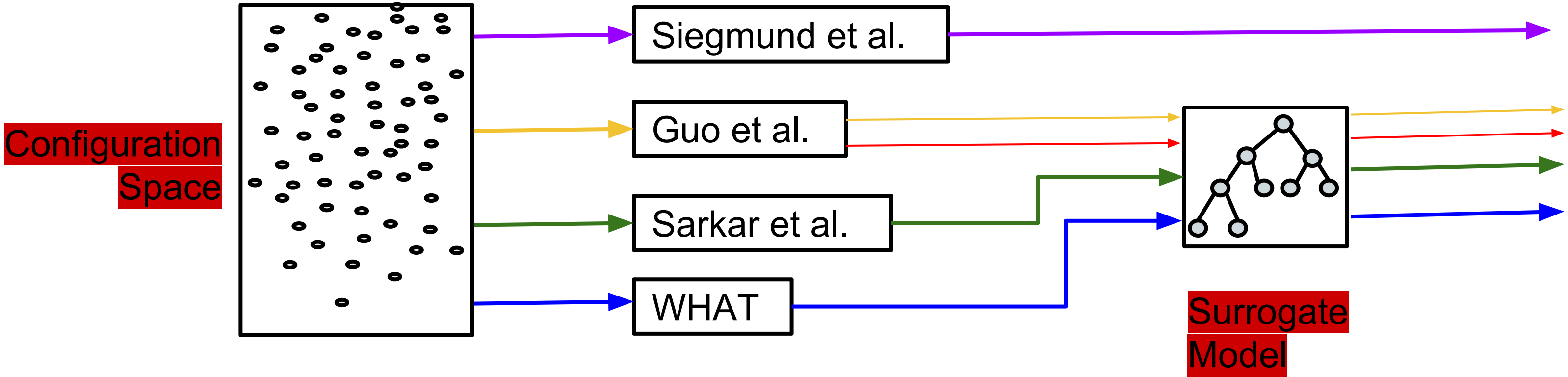
- If WHAT is better than state-of-the-art techniques
  - Siegmund et al. - FW heuristics
  - Guo et al. - Progressive Sampling
  - Sarkar et al. - Random Sampling + Feature-wise heuristics



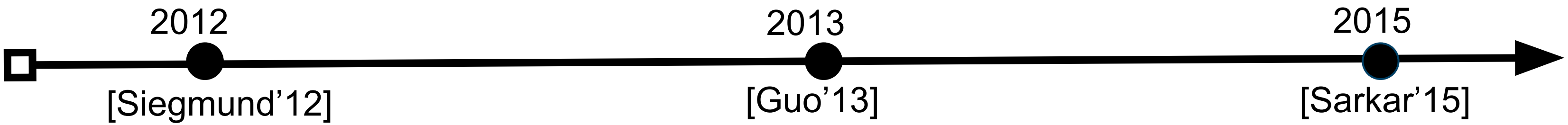
RQ 4: How good is WHAT compared to the state of the art predictors?

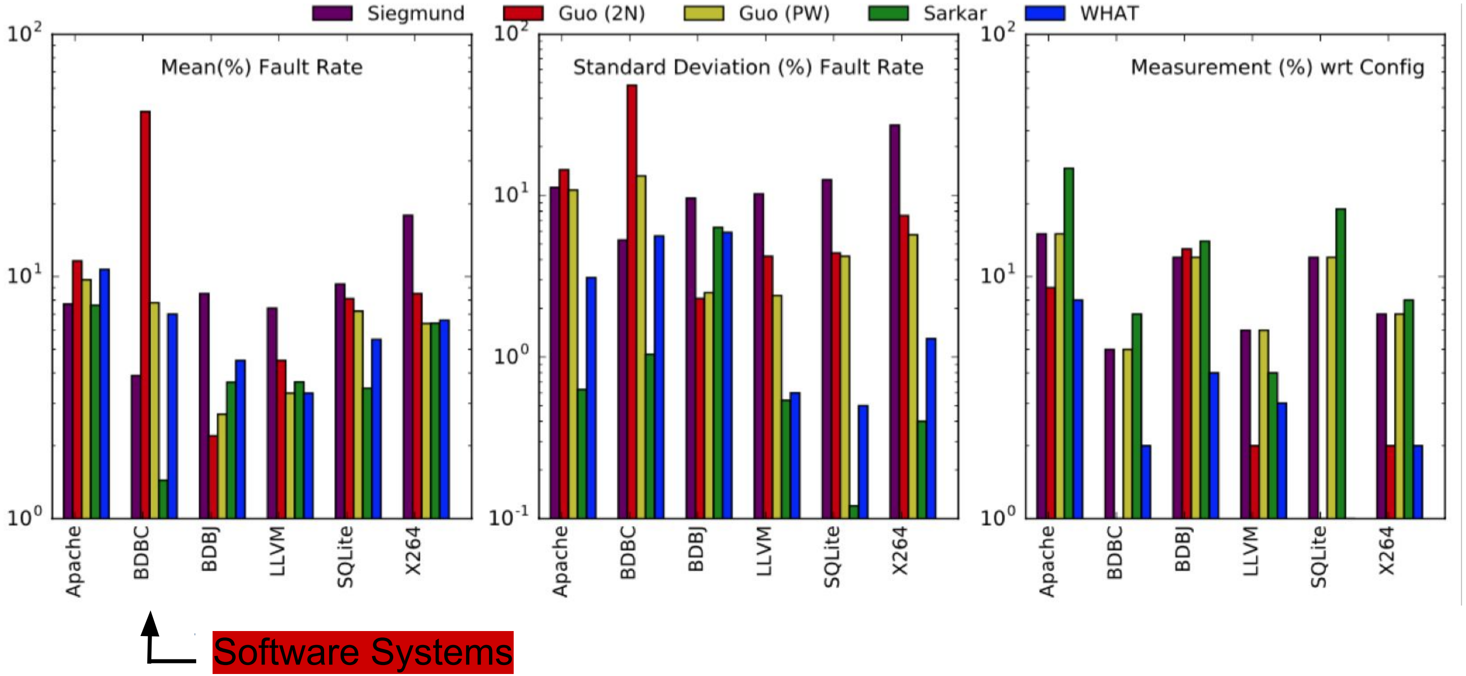
RQ 4 explores

- If WHAT is better than state-of-the-art techniques
  - Siegmund et al. - FW heuristics
  - Guo et al. - Progressive Sampling
  - Sarkar et al. - Random Sampling + Feature-wise heuristics

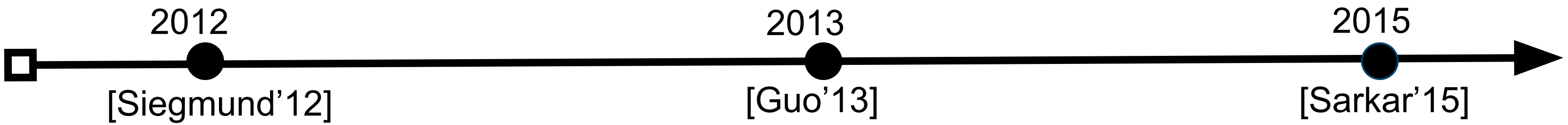


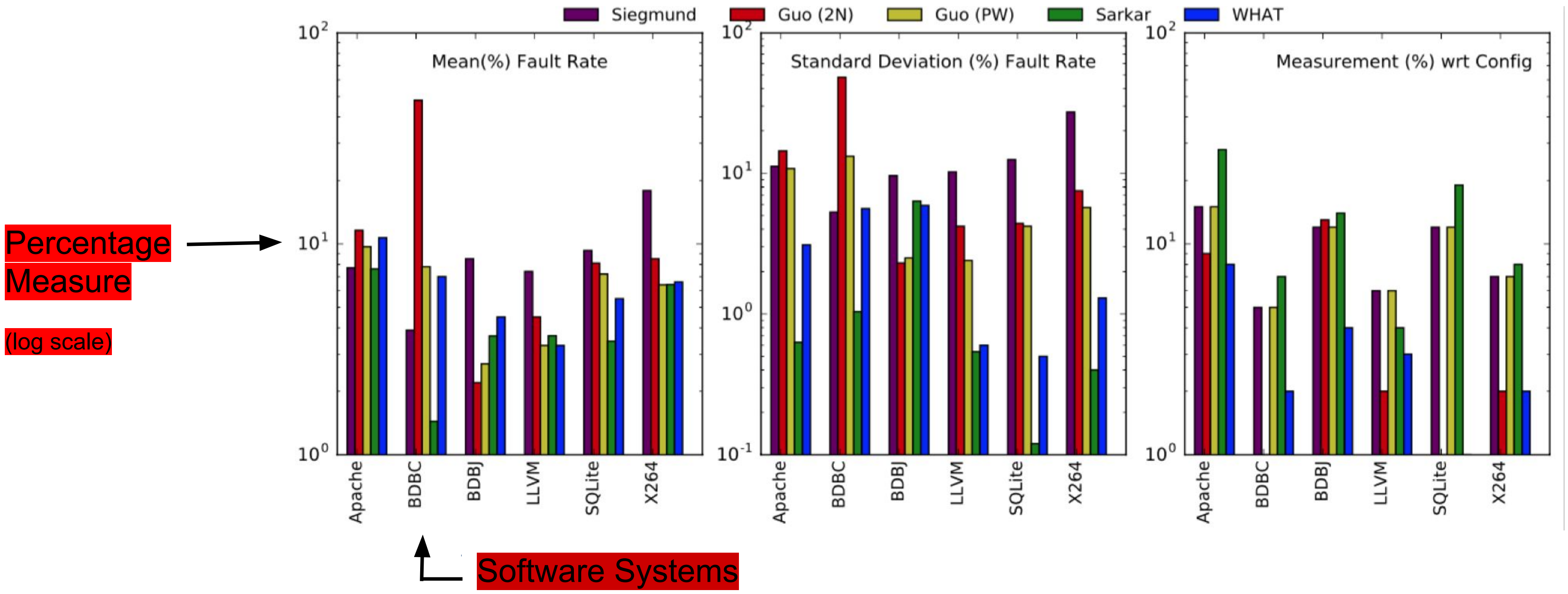
RQ 4: How good is WHAT compared to the state of the art predictors?



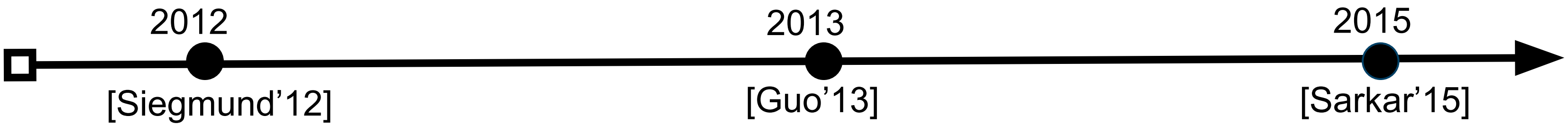


RQ 4: How good is WHAT compared to the state of the art predictors?



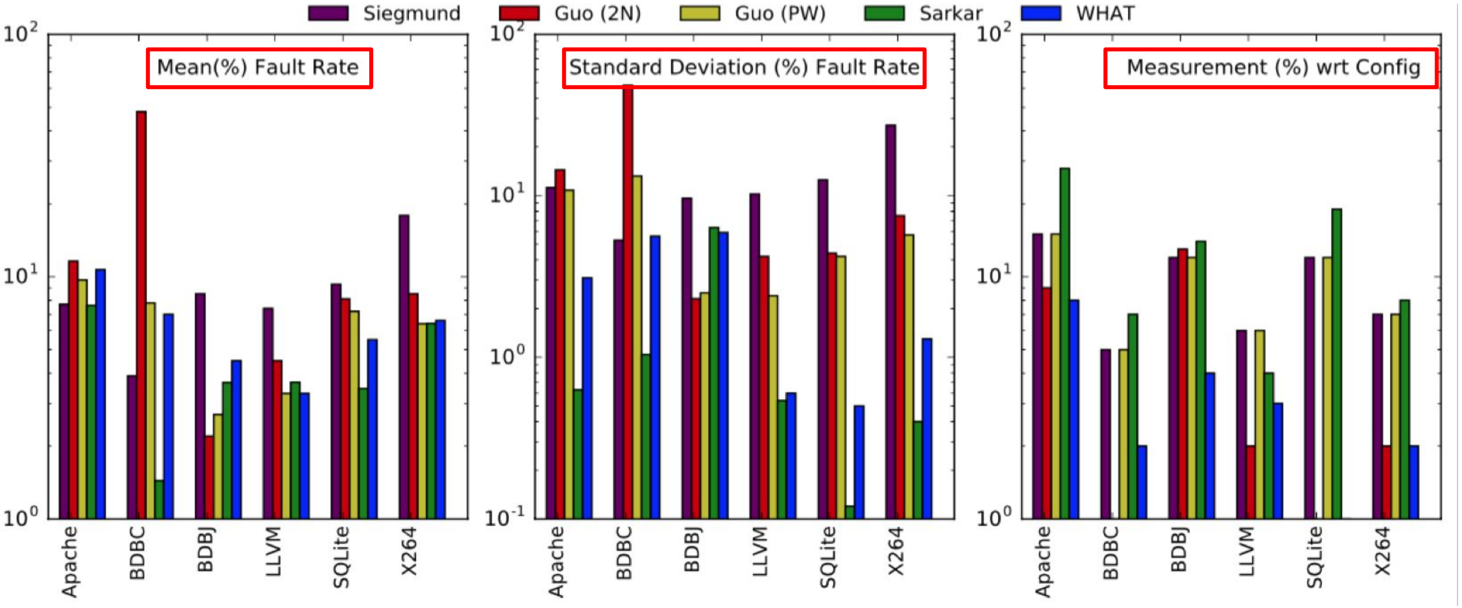


RQ 4: How good is WHAT compared to the state of the art predictors?



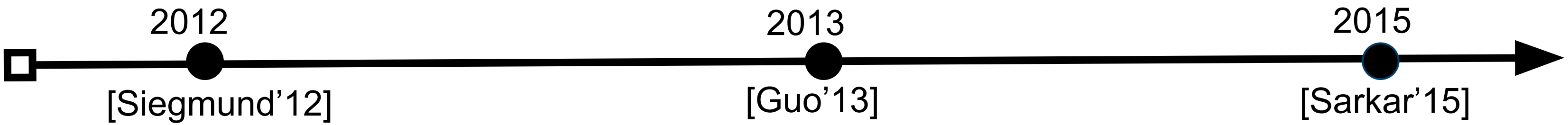
Percentage Measure

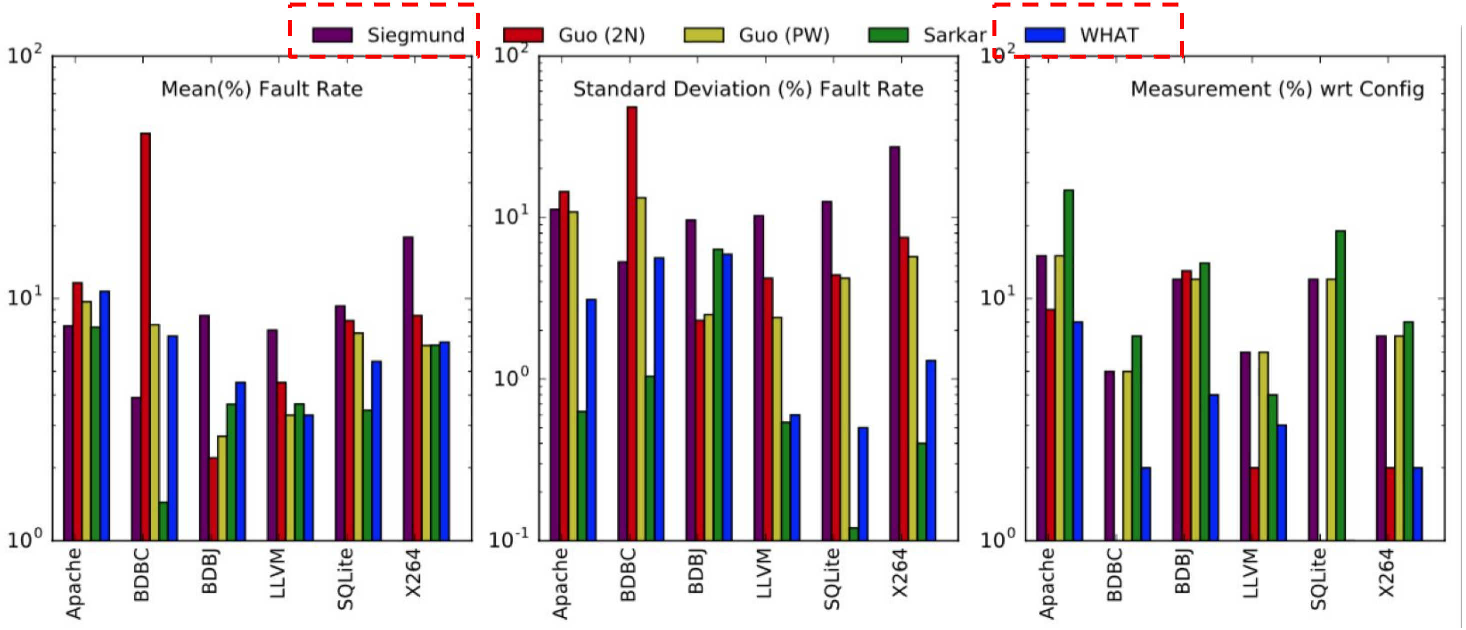
(log scale)



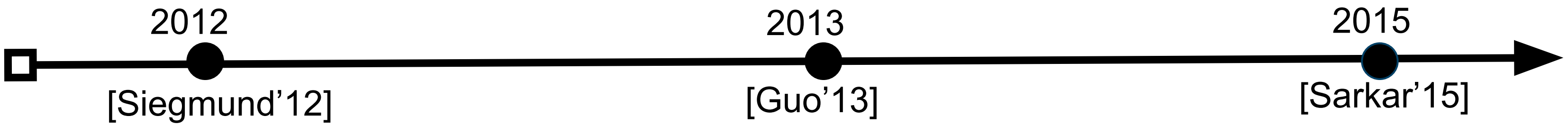
Software Systems

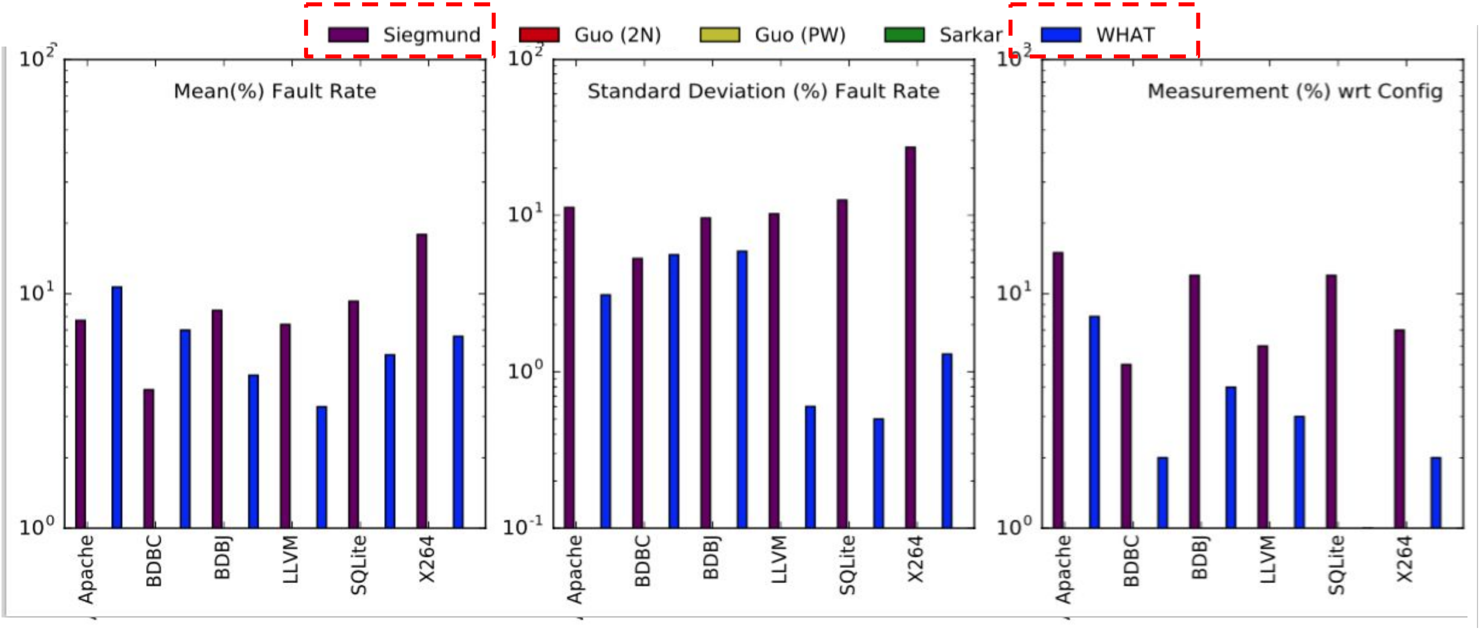
RQ 4: How good is WHAT compared to the state of the art predictors?



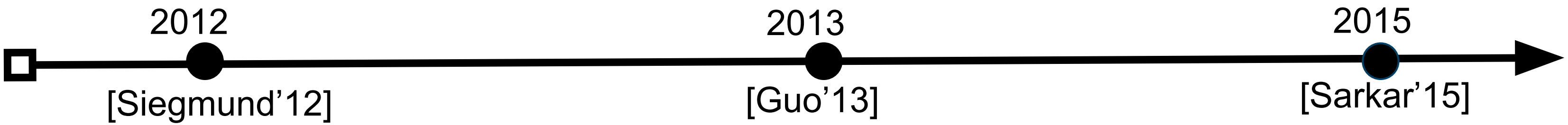


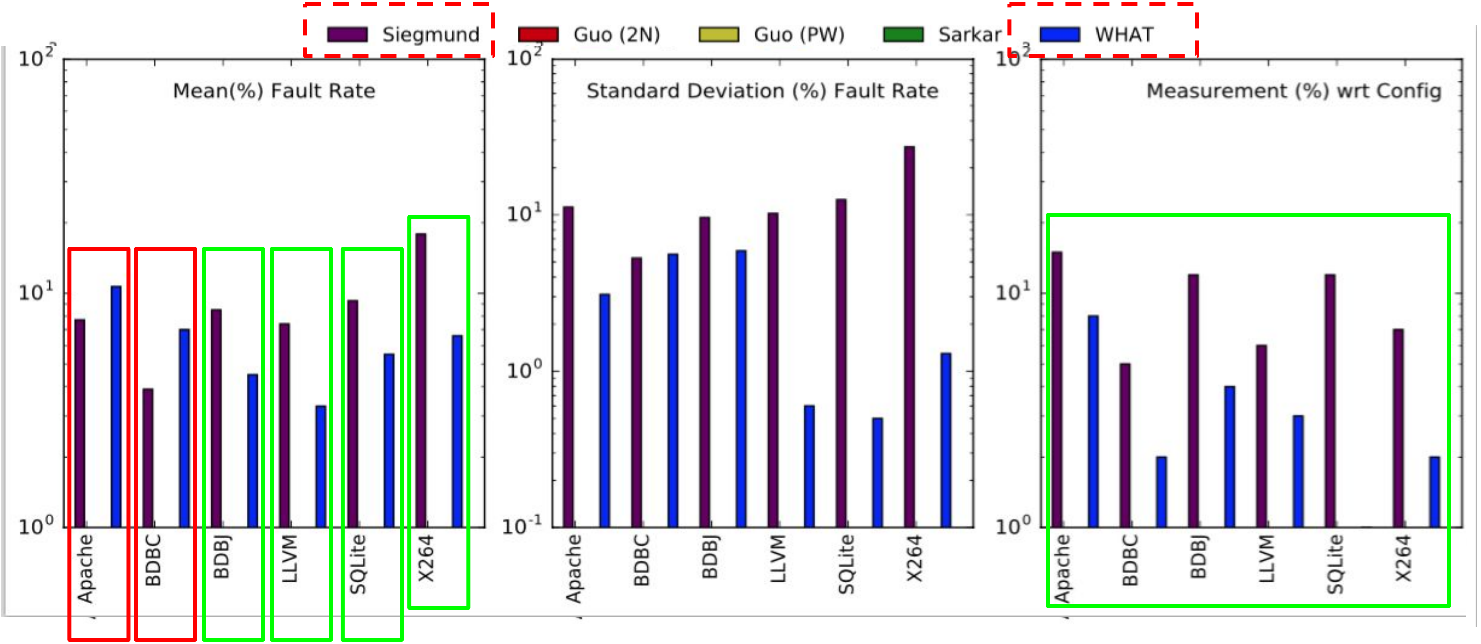
RQ 4: How good is WHAT compared to the state of the art predictors?



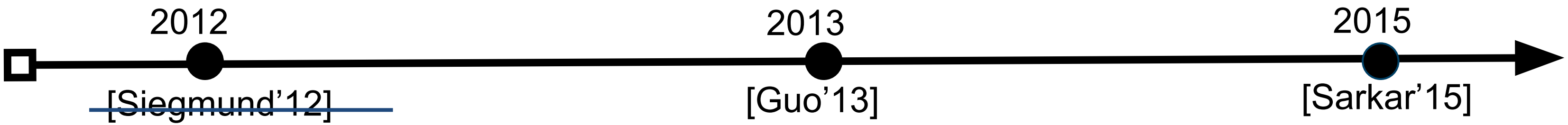


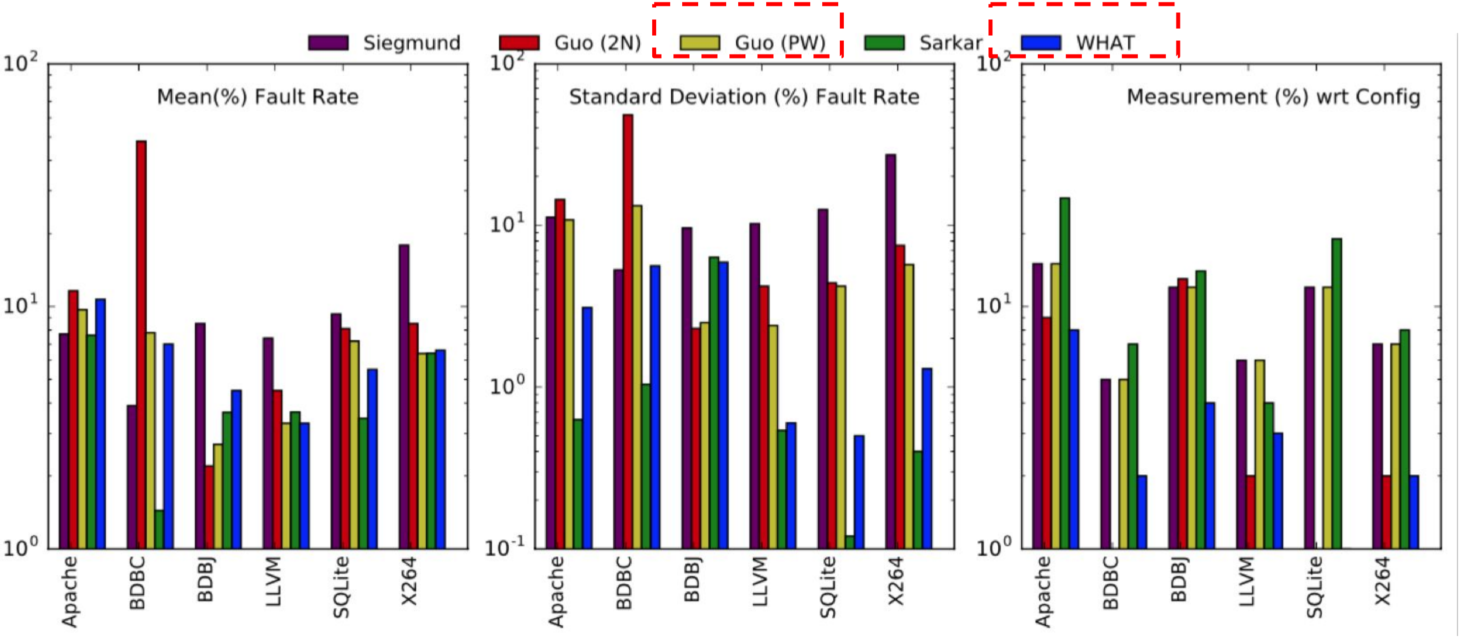
RQ 4: How good is WHAT compared to the state of the art predictors?



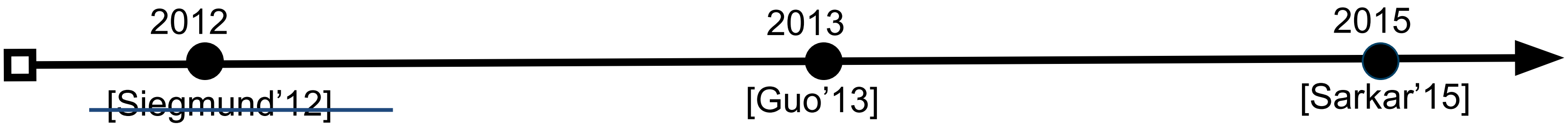


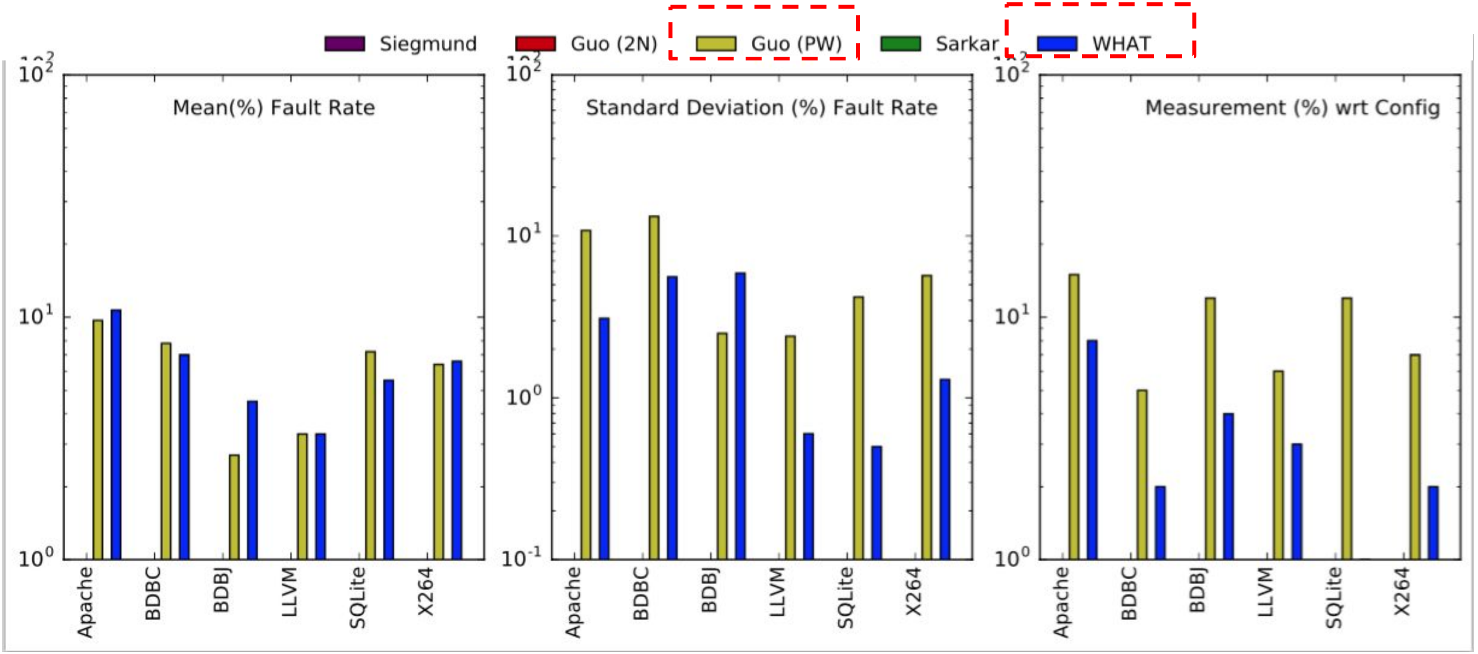
RQ 4: How good is WHAT compared to the state of the art predictors?





RQ 4: How good is WHAT compared to the state of the art predictors?



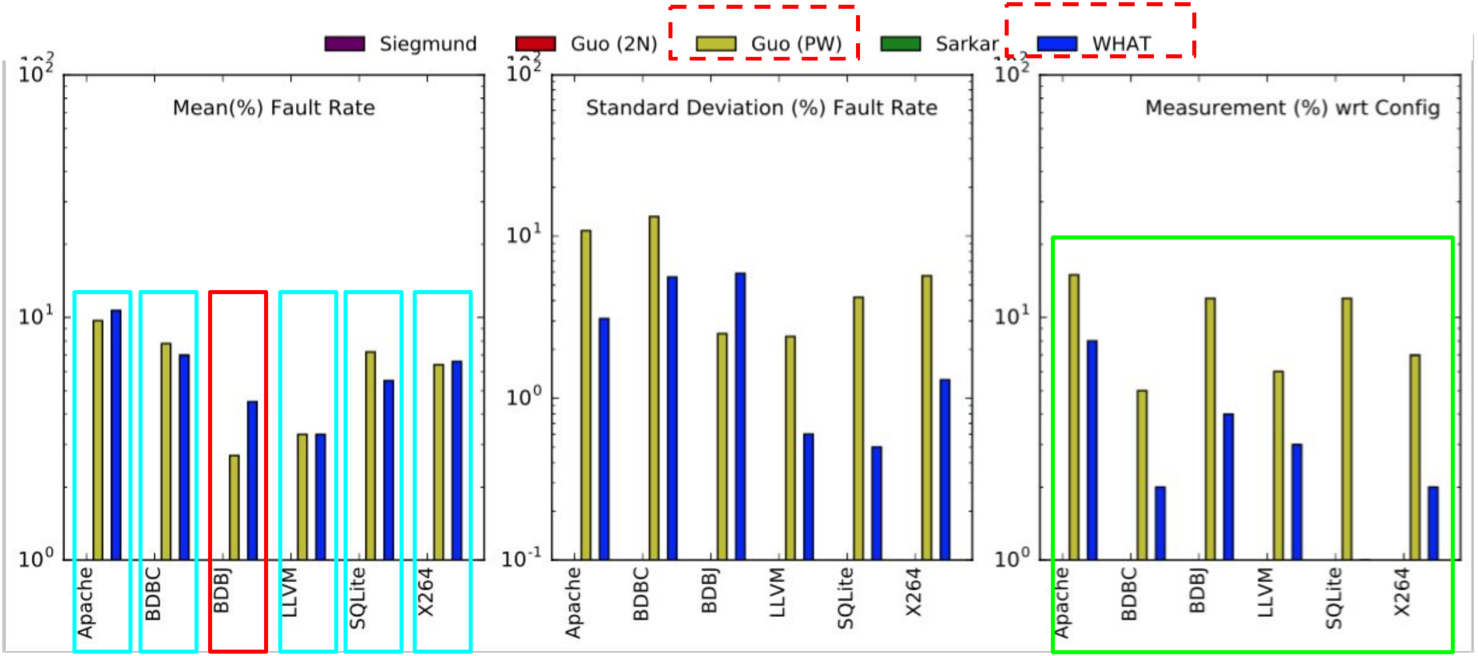


RQ 4: How good is WHAT compared to the state of the art predictors?

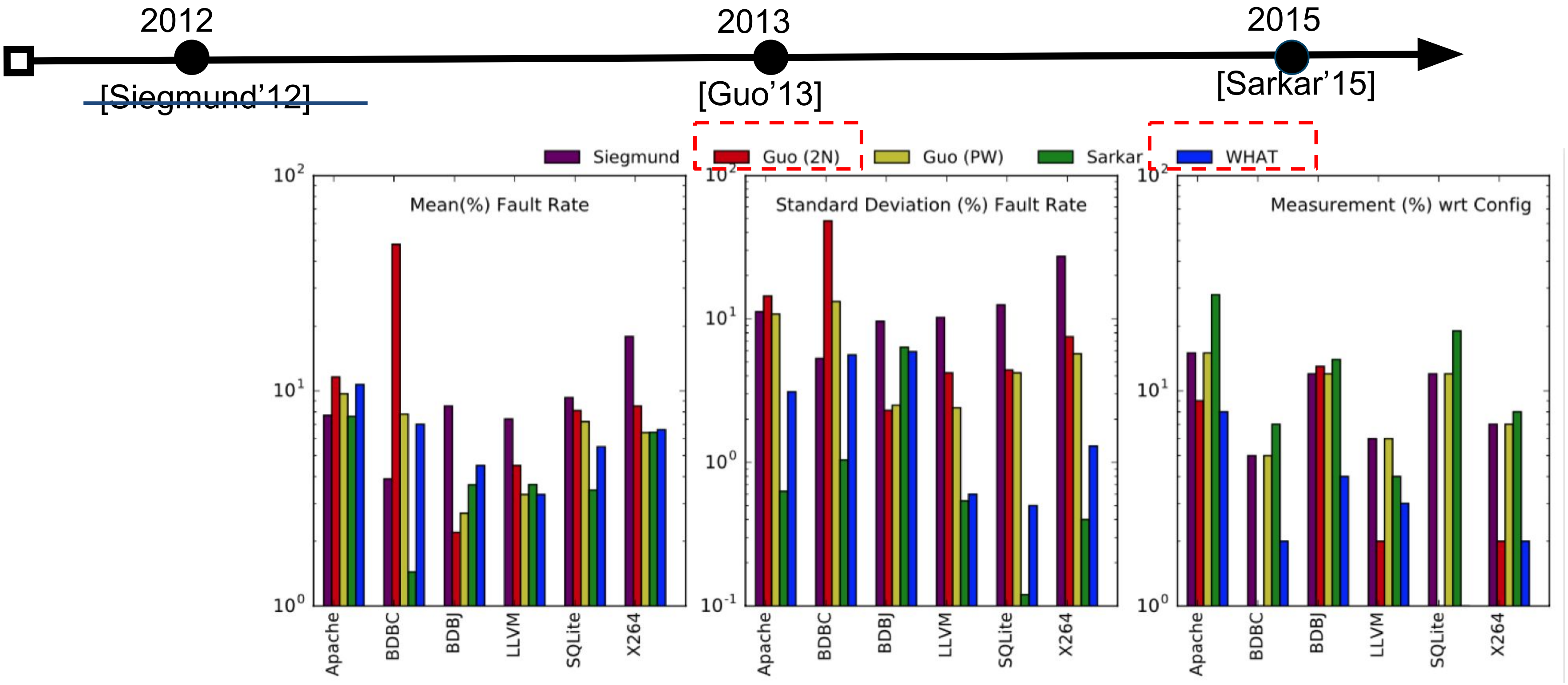
2012 — ~~[Siegmund'12]~~

2013 — [Guo'13]

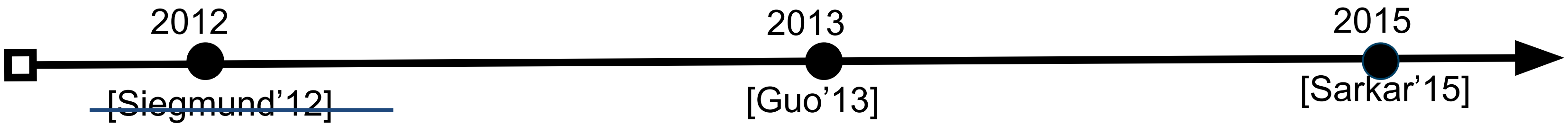
2015 — [Sarkar'15]

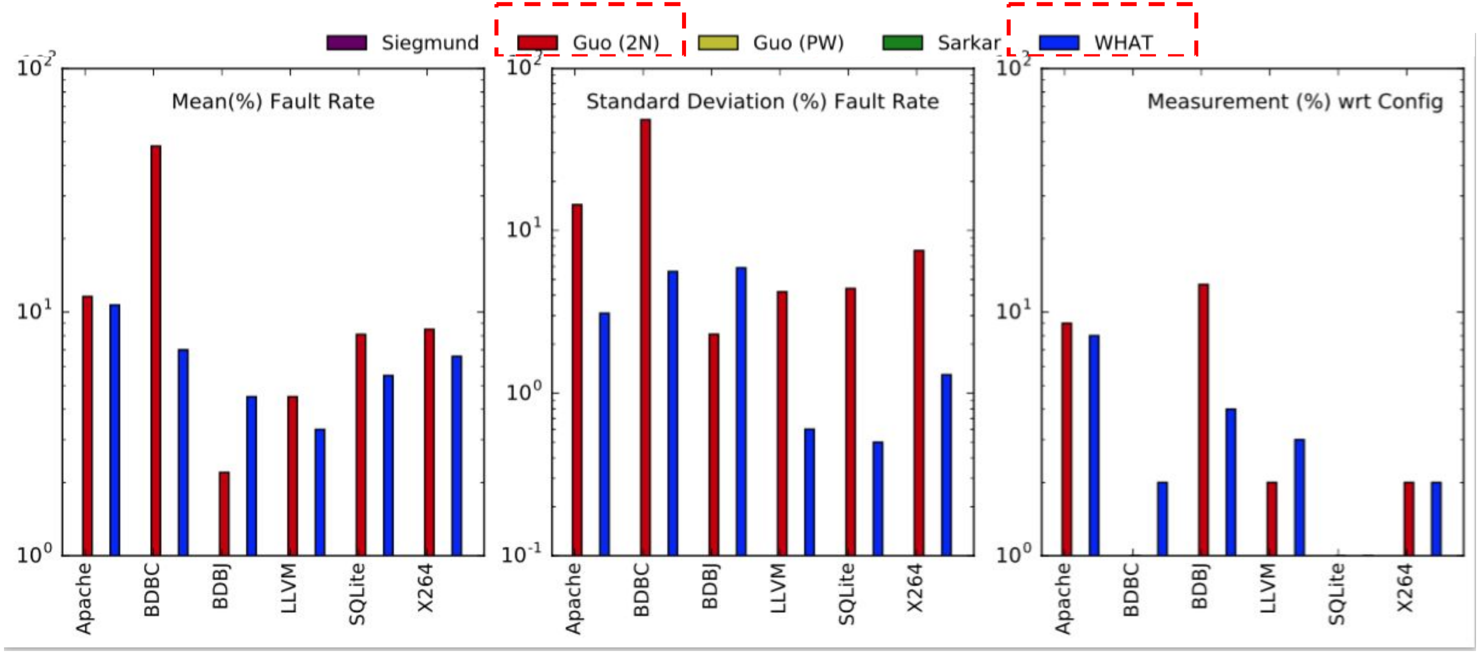


RQ 4: How good is WHAT compared to the state of the art predictors?



RQ 4: How good is WHAT compared to the state of the art predictors?



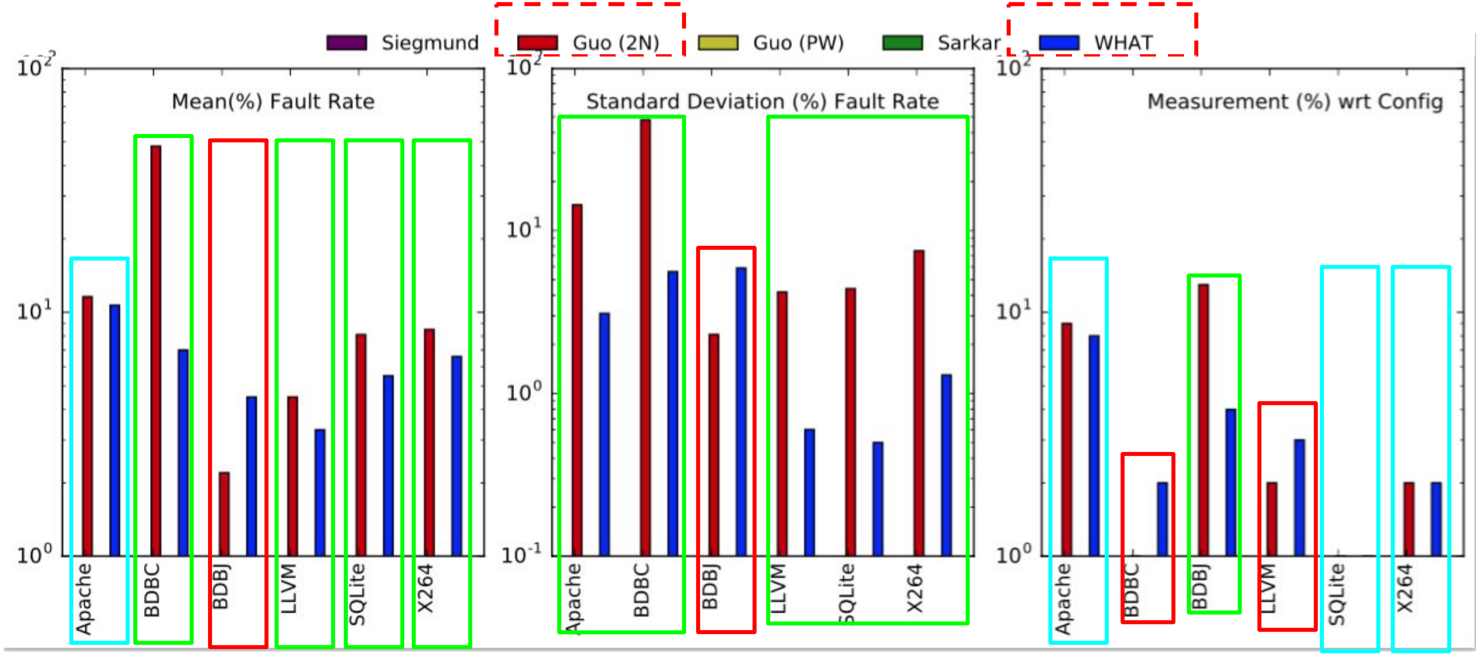


RQ 4: How good is WHAT compared to the state of the art predictors?

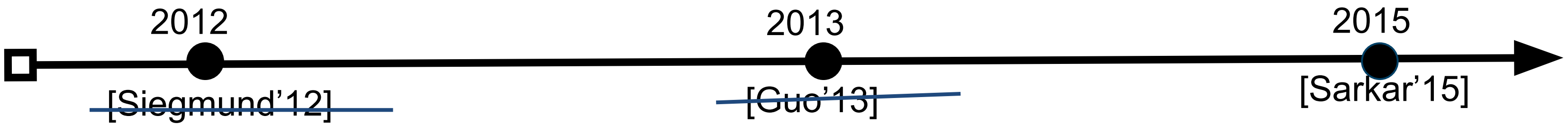
2012 — ~~[Siegmund'12]~~

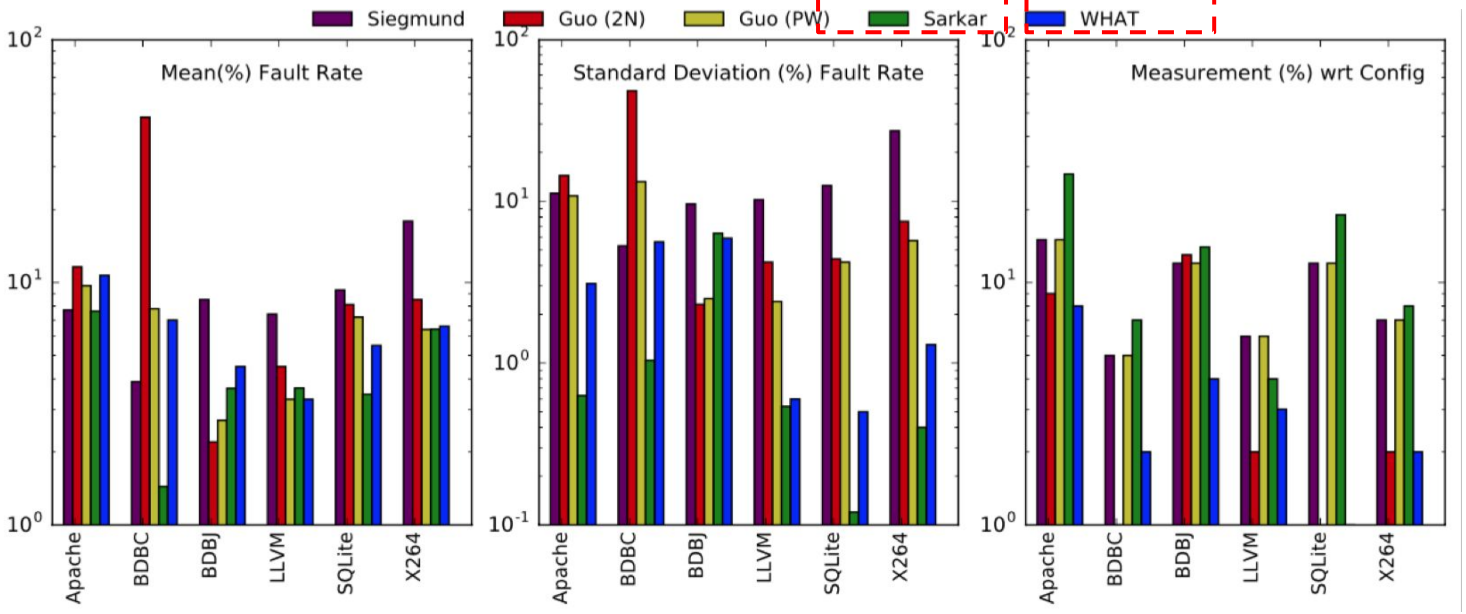
2013 — [Guo'13]

2015 — [Sarkar'15]



RQ 4: How good is WHAT compared to the state of the art predictors?



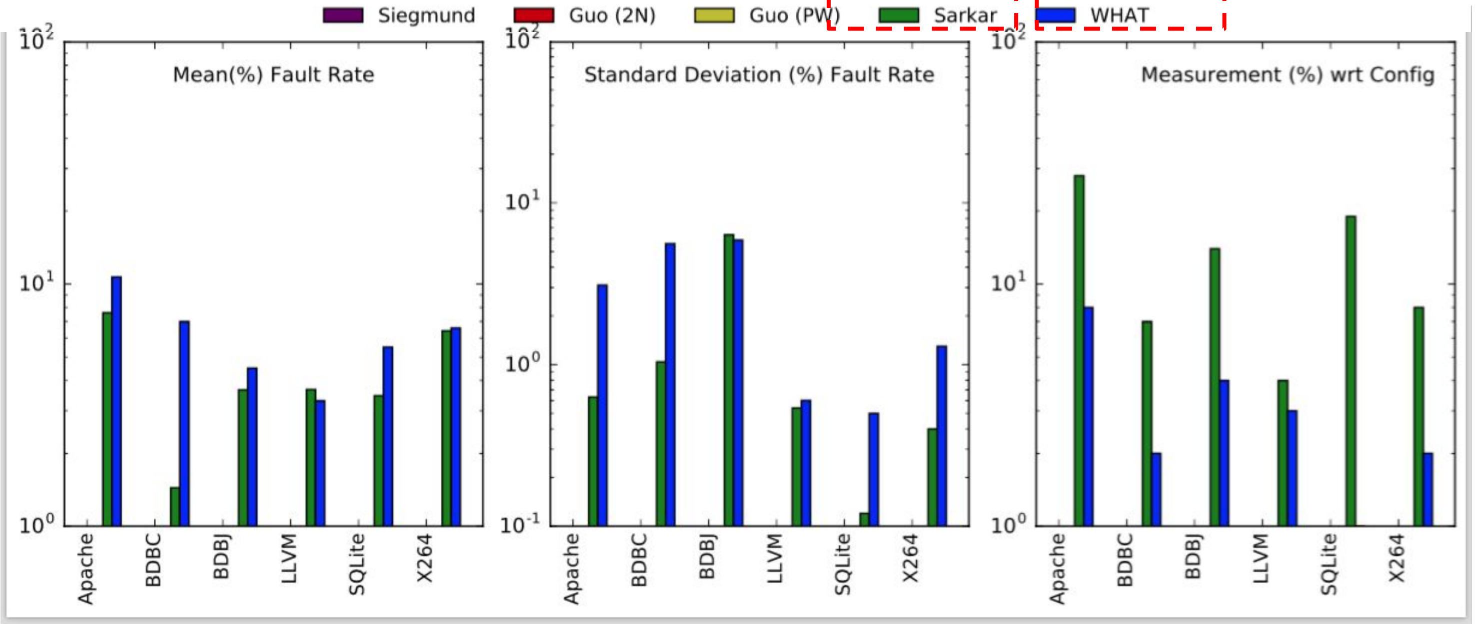


RQ 4: How good is WHAT compared to the state of the art predictors?

2012 [~~Siegmund'12~~]

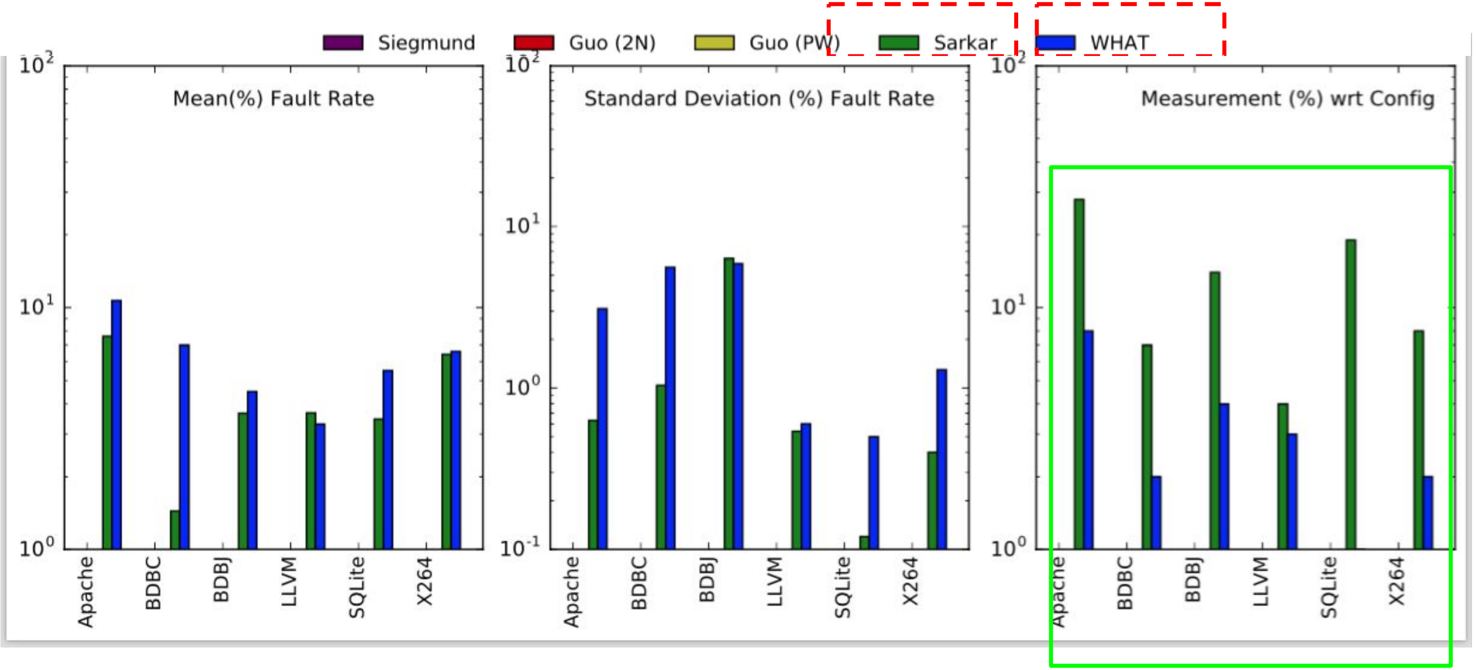
2013 [~~Guo'13~~]

2015 [Sarkar'15]

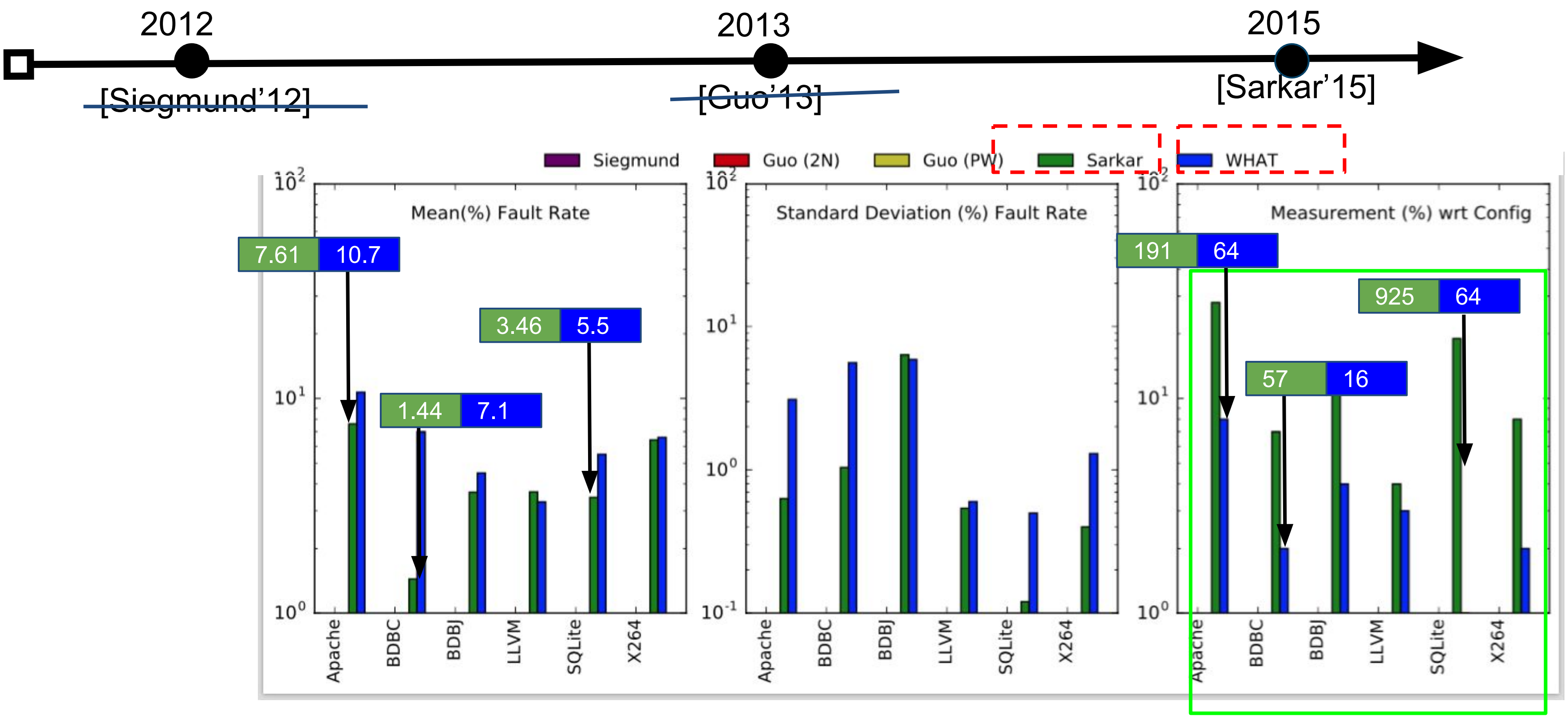


RQ 4: How good is WHAT compared to the state of the art predictors?

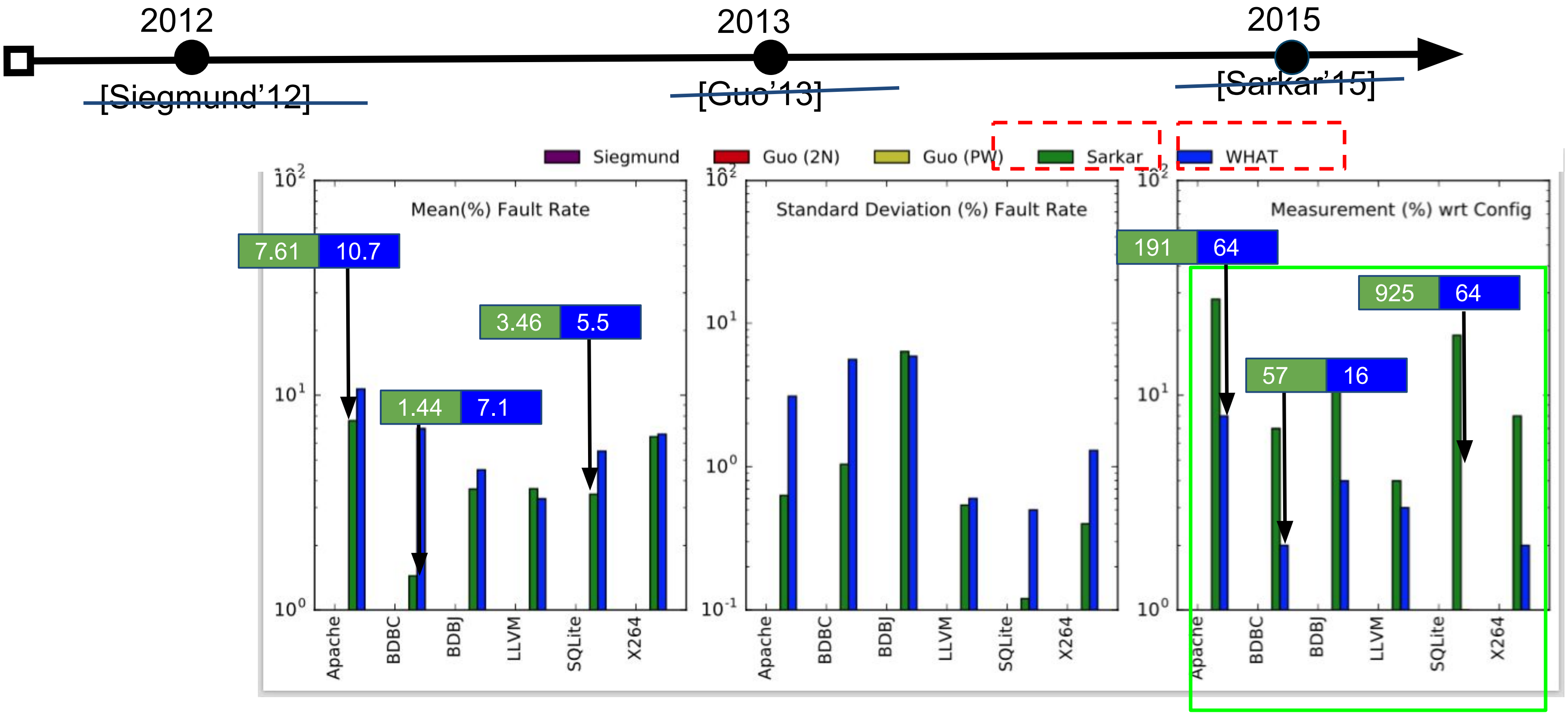
2012 ~~[Siegmund'12]~~ 2013 ~~[Guo'13]~~ 2015 [Sarkar'15]



RQ 4: How good is WHAT compared to the state of the art predictors?



RQ 4: How good is WHAT compared to the state of the art predictors?



Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations?

RQ 2: Do less data cause larger variances in predicted values?

RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

RQ 4: How good is WHAT compared to the state of the art predictors?

Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations? **YES**

RQ 2: Do less data cause larger variances in predicted values?

RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

RQ 4: How good is WHAT compared to the state of the art predictors?

Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations? **YES**

RQ 2: Do less data cause larger variances in predicted values? **NO**

RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT?

RQ 4: How good is WHAT compared to the state of the art predictors?

Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations? **YES**

RQ 2: Do less data cause larger variances in predicted values? **NO**

RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT? **YES**

RQ 4: How good is WHAT compared to the state of the art predictors?

Research Questions

RQ 1: Can WHAT generate good predictions using only a small number of configurations? **YES**

RQ 2: Do less data cause larger variances in predicted values? **NO**

RQ 3: Can "good" surrogate models (to be used in optimizers) be built using WHAT? **YES**

RQ 4: How good is WHAT compared to the state of the art predictors? **Comparable**

Future Work

Future Work

- Progressive WHAT
  - WHAT is rigid
  - Has no options of budget
  - Progressive Sampling using WHAT

- Multi-objective Problems
  - Problem are multi-objective
  - New surrogates required
  - New surrogate model update techniques

- Sampling Way
  - Sampling is preferable if evaluation is expensive
  - Initial results are competitive with other algorithms

- Spectral Grid Search
  - Exploit the underlying dimension while generating Grids

RQ 1: Can WHAT generate good predictions using only a small number of configurations? **YES**

RQ 2: Do less data cause larger variances in predicted values? **NO**

RQ 3: Can “good” surrogate models (to be used in optimizers) be built using WHAT? **YES**

RQ 4: How good is WHAT compared to the state of the art predictors? **Comparable**

Question and Comments

References

- [Samad'08] A. Samad and K.-Y. Kim, "Multiple surrogate modeling for axial compressor blade shape optimization," J. Propulsion Power, vol. 24, no. 2, pp. 302–310, Mar. 2008.
- [Zerpa'05] L. E. Zerpa, N. V. Queipo, S. Pintos, and J.-L. Salager, "An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates," J. Petroleum Sci. Eng., vol. 47, no. 3–4, pp. 197–208, Jun. 2005.
- [Marjavaara'07] D. Marjavaara, S. Lundstr¨om, and W. Shyy, "Hydraulic turbine diffuser shape optimization by multiple surrogate model approximations of pareto fronts," ASME J. Fluids Eng., vol. 129, no. 9, pp. 1228–1240, 2007.
- [Sanchez'06] E. Sanchez, S. Pintos, and N. V. Queipo, "Toward an optimal ensemble of kernel-based approximations with engineering applications," in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2006, pp. 2152–2158.
- [Sakata'08] Sakata, S., F. Ashida, and M. Zako. "Microstructural design of composite materials using fixed-grid modeling and noise-resistant smoothed Kriging-based approximate optimization." *Structural and Multidisciplinary Optimization* 36.3 (2008): 273-287.
- [Guo'13] Jianmei Guo, Krzysztof Czarnecki, Sven Apel, Norbert Siegmund, and Andrzej Wasowski. Variability-aware performance prediction: A statistical learning approach. In IEEE/ACM 28th International Conference on Automated Software Engineering, pages 301–311. IEEE, 2013.
- [Bergstra'12] Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." The Journal of Machine Learning Research 13.1 (2012): 281-305.
- [Loshchilov'10] Loshchilov, Ilya, Marc Schoenauer, and Michèle Sebag. "A mono surrogate for multiobjective optimization." Proceedings of the 12th annual conference on Genetic and evolutionary computation. ACM, 2010.
- [Abboud'01] Abboud, Schoenauer. Surrogate deterministic mutation: Preliminary results, Artificial Evolution '01
- [Zhou'07] Zhou, Zongzhao, et al. "Combining global and local surrogate models to accelerate evolutionary optimization." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37.1 (2007): 66-76.
- [Storn'95] Storn, Rainer, and Kenneth Price. *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces*. Vol. 3. Berkeley: ICSI, 1995.
- [Deb'02] Deb, Kalyanmoy, et al. "A fast and elitist multiobjective genetic algorithm: NSGA-II."*Evolutionary Computation, IEEE Transactions on*6.2 (2002): 182-197.