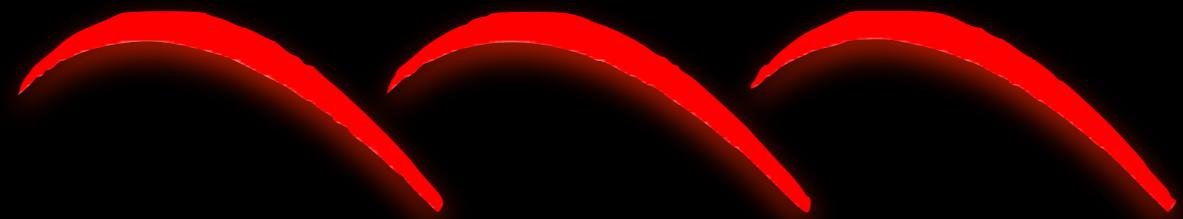


The Relationship between Search Based Software Engineering and Perceptive Model

CREST



Mark Harman

Centre for Research on Evolution Search and Testing

Est. 2006

I admin

4 faculty

8 post docs

II PhD students

I-4 resident visiting scholars



Research

Search Based Software Engineering

Quantitative Information Flow

Requirement Engineering

Service Oriented Computing

Dependence Analysis

Clone Detection

Testing

Digital Humanities

Thanks for slides

Yue Jia, CREST

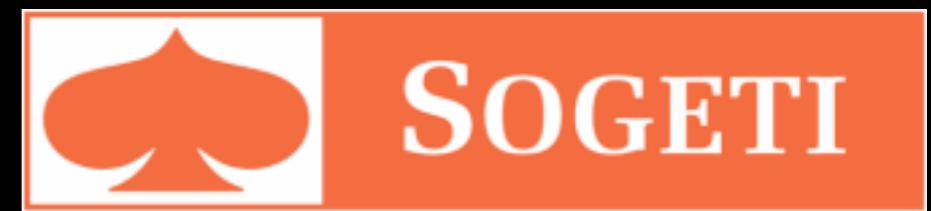
Spiros Mancoridis, Drexel

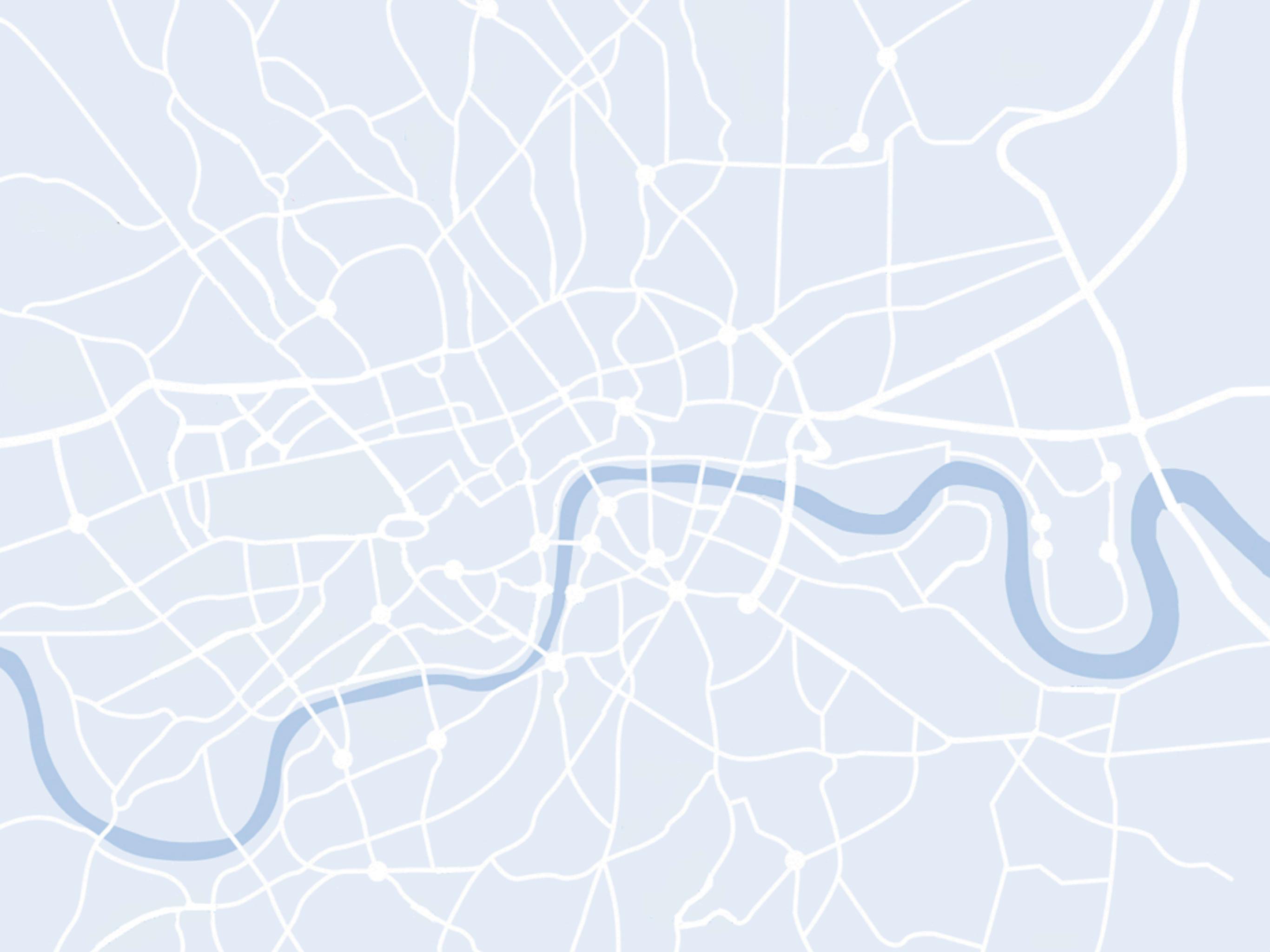
Shin Yoo, CREST

Joachim Wegener, B&M

Yuanyuan Zhang, CREST

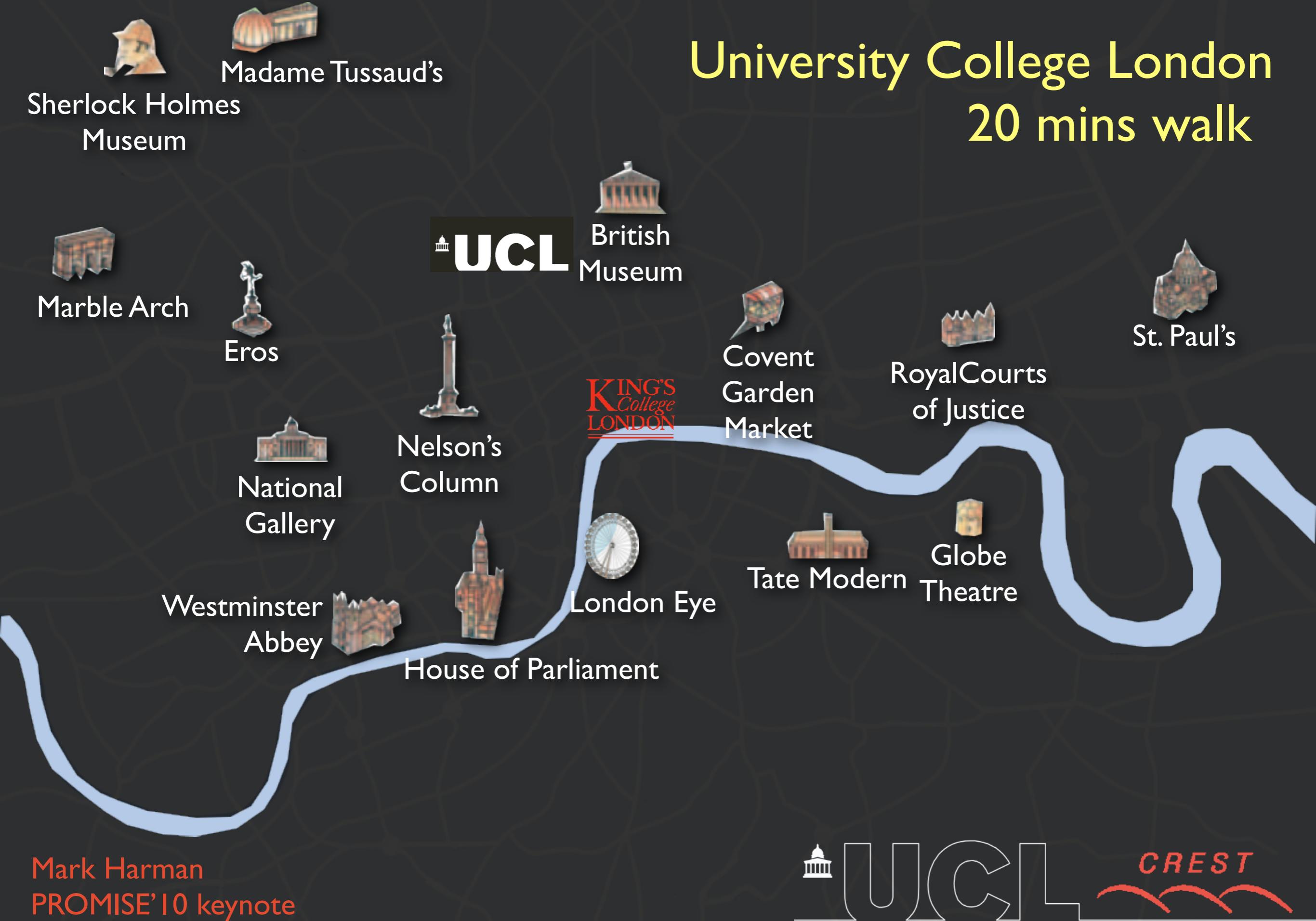
Funding





University College London

20 mins walk



Overview

Context

What is SBSE? Why should I care?

Examples

Pointers to literature

Relationships between SBSE and Predictive Models

SBSE for PM

How much do we need to know to build effective models?

How to adapt models to new data?

How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

Scientists' and Engineers' viewpoints

scientist:

what is true
correctness
model the world
to understand

engineer:

what is possible
within tolerance
model the world
to manipulate

Scientists' and Engineers' viewpoints

computer scientist:

what is true
about computation

prove correctness
make it perfect

software engineer:

what is possible
with software

test for imperfection
find where to improve

Combining Science and Engineering

where possible ...
... and where impossible ...
prove correctness test for imperfection
make it perfect find where to improve

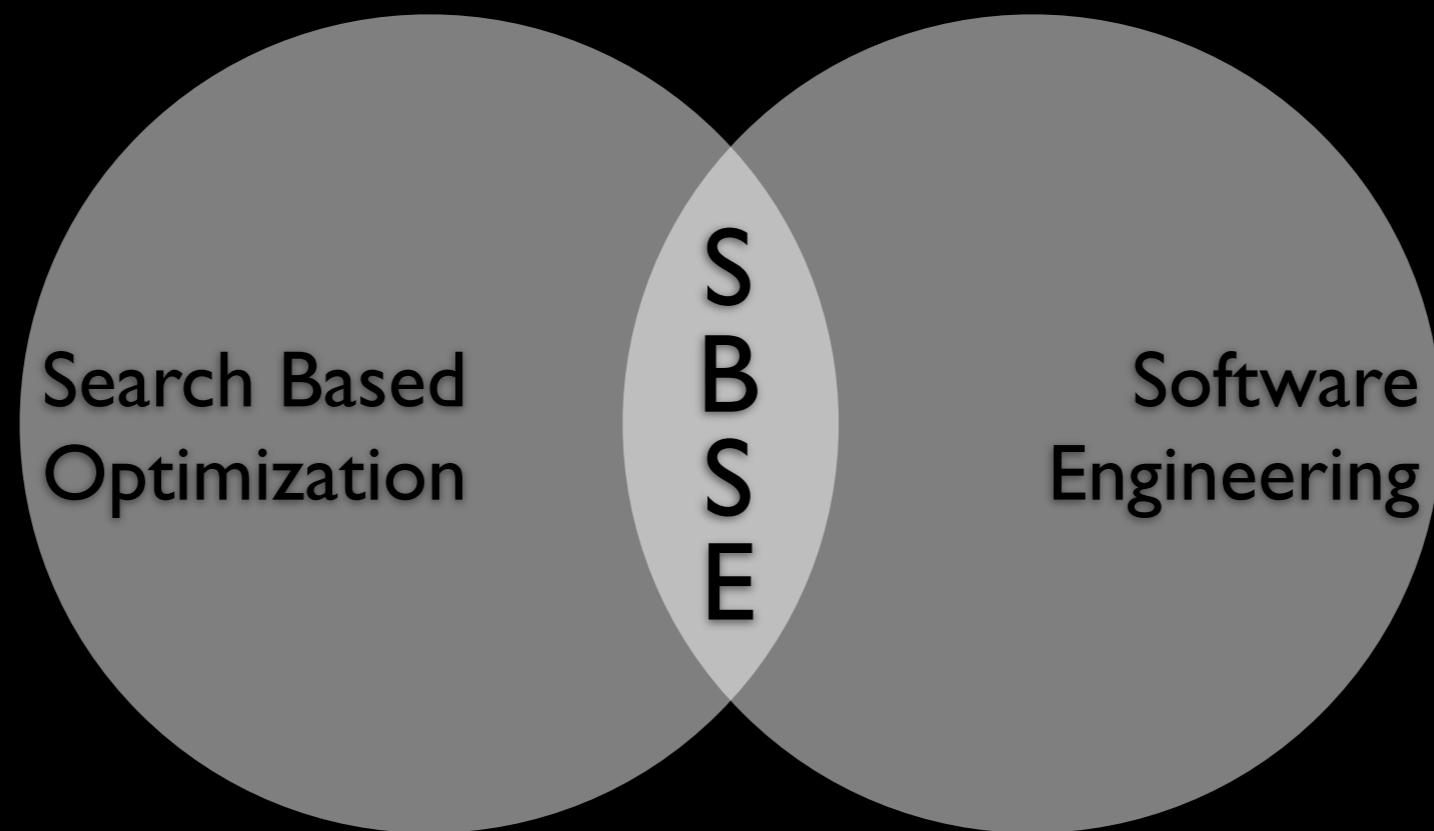
SBSE and PM

SBSE & Predictive modeling are *engineering*

We suffer for this in similar ways

- resistance to approximate methods
- non determinism is a non starter for formalists

What is SBSE



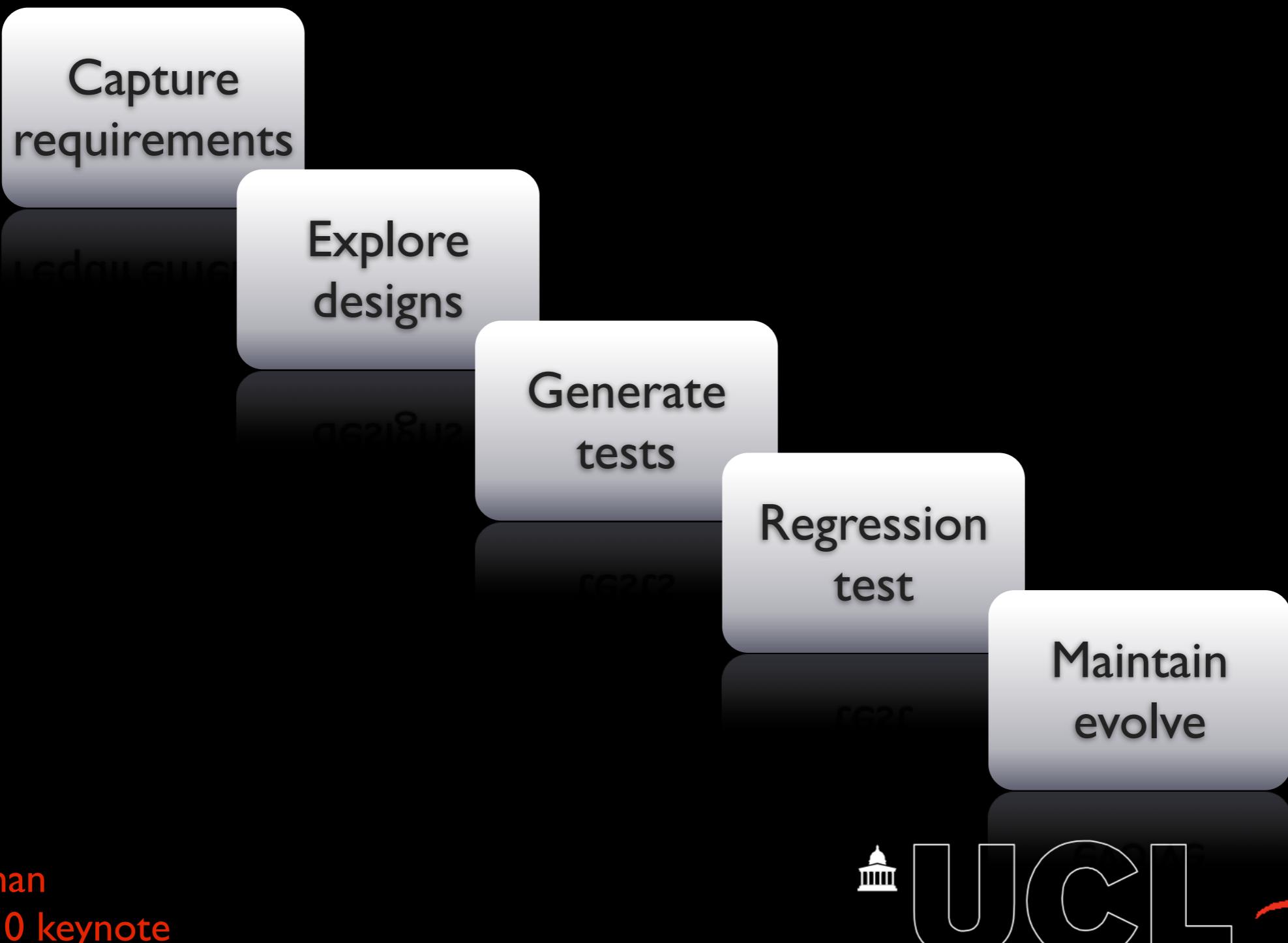
SBSE world view

SBSE takes an optimization approach

This is very generic

Engineering is all about optimization...

Software Engineering Problems



Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Minimize

Maximize

Maximize

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Cost
Development time

Satisfaction
Fairness

Maximize
Minimize

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Number of test
Execution time

Code coverage
Fault coverage

Maximize

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Coupling

Cohesion

Maximize
Minimize

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Coupling

Cohesion

Maximize

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Number of test
Execution time

Code coverage
Fault coverage

Emergent paradigms



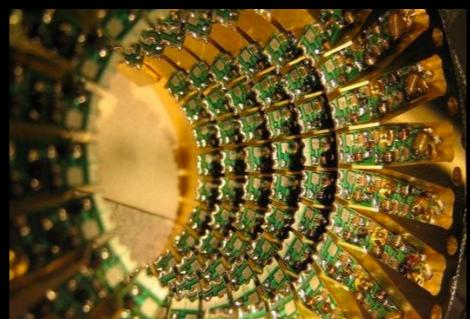
Cloud



Agile



Ambient



Quantum



Non functional
properties



Emergent paradigms



Cloud

Objectives:

leakage/sharing
fairness
response

Emergent paradigms



Agile

Objectives:

cost
value
refactoring split points

Emergent paradigms



Ambient

Objectives:

coverage
response
fit

Emergent paradigms

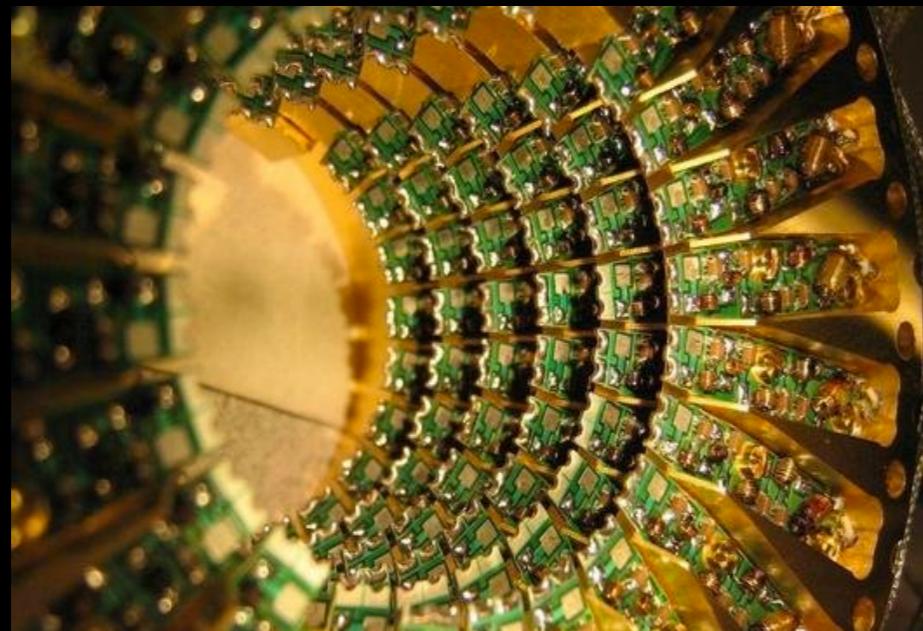


Non functional
properties

Objectives:

heat dissipation
WCET
resource use

Emergent paradigms



Quantum

Objectives:

functionality
entanglement
interference

What is SBSE

In SBSE we apply search techniques to search large search spaces, guided by a fitness function that captures properties of the acceptable software artefacts we seek.

What is SBSE

In SBSE we apply **search techniques** to search large search spaces, guided by a fitness function that captures properties of the acceptable software artefacts we seek.

Tabu Search

Ant Colonies

Particle Swarm Optimization

Hill Climbing

Genetic Algorithms

Simulated Annealing

Genetic Programming

Random

Greedy

LP

Estimation of Distribution Algorithms

What is SBSE

In SBSE we apply **search techniques** to search large search spaces, guided by a **fitness function** that captures properties of the acceptable software artefacts we seek.

Tabu Search

Ant Colonies

Particle Swarm Optimization

Hill Climbing

Genetic Algorithms

Genetic Programming

Simulated Annealing

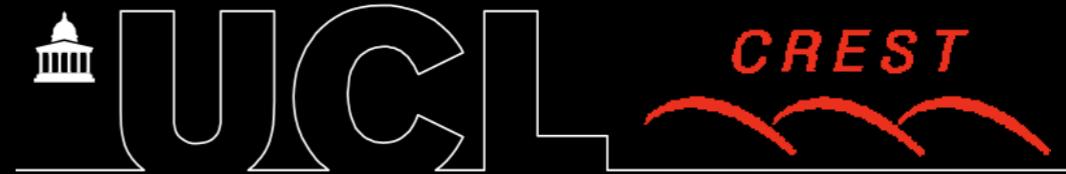
Greedy

Random

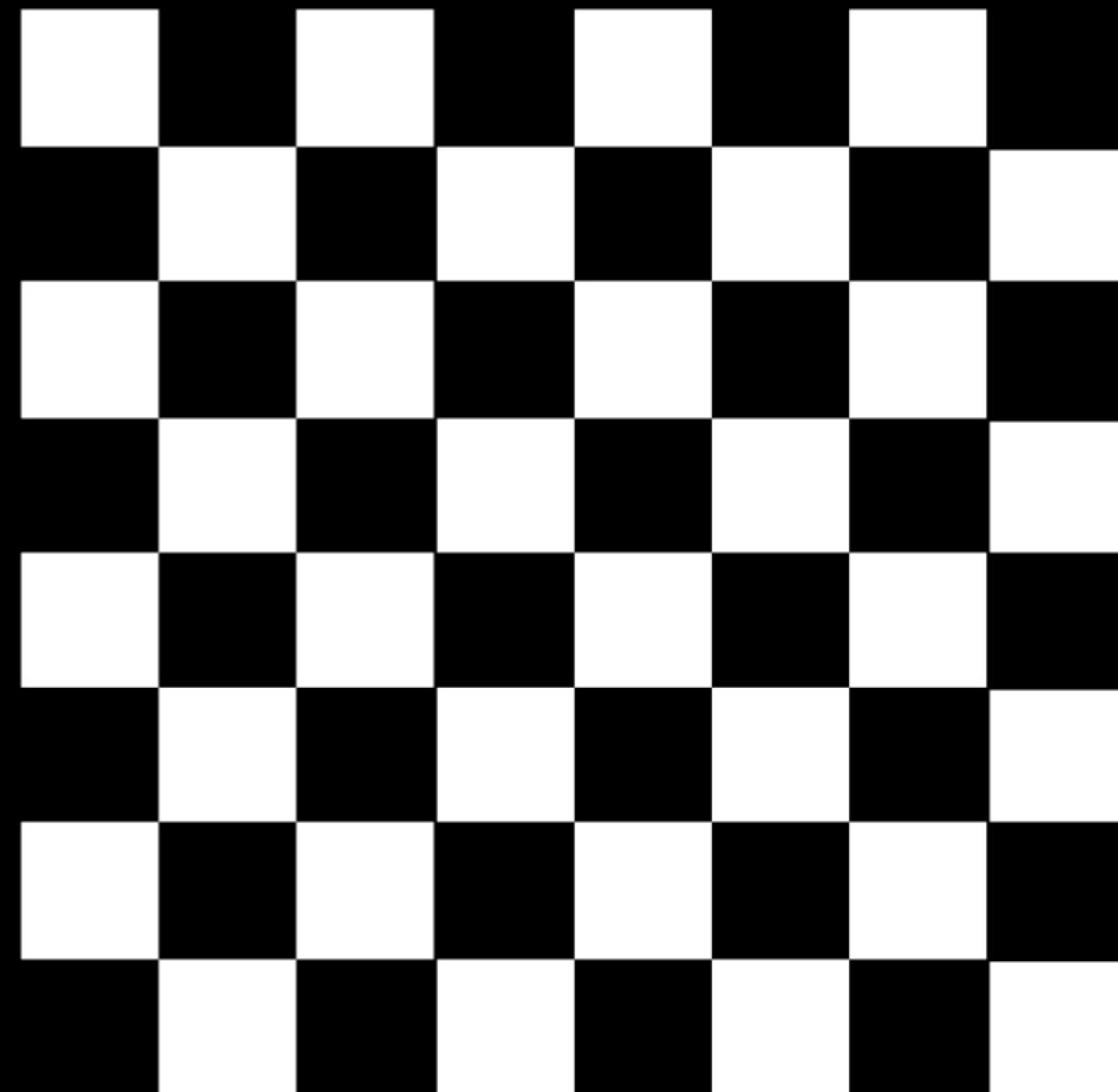
Estimation of Distribution Algorithms

Why?

Mark Harman
PROMISE'10 keynote

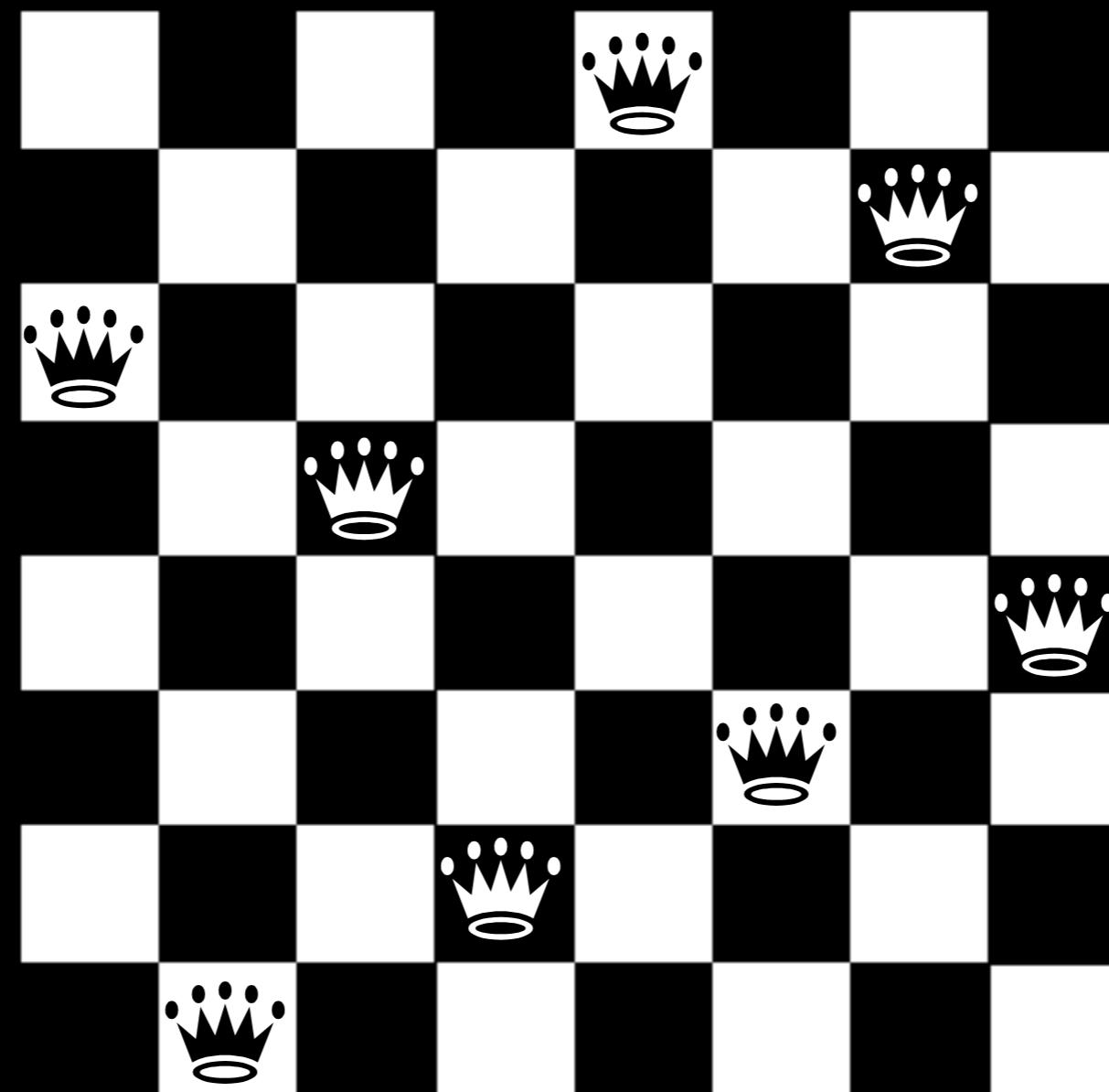


Eight Queens Problem



Eight Queens Problem

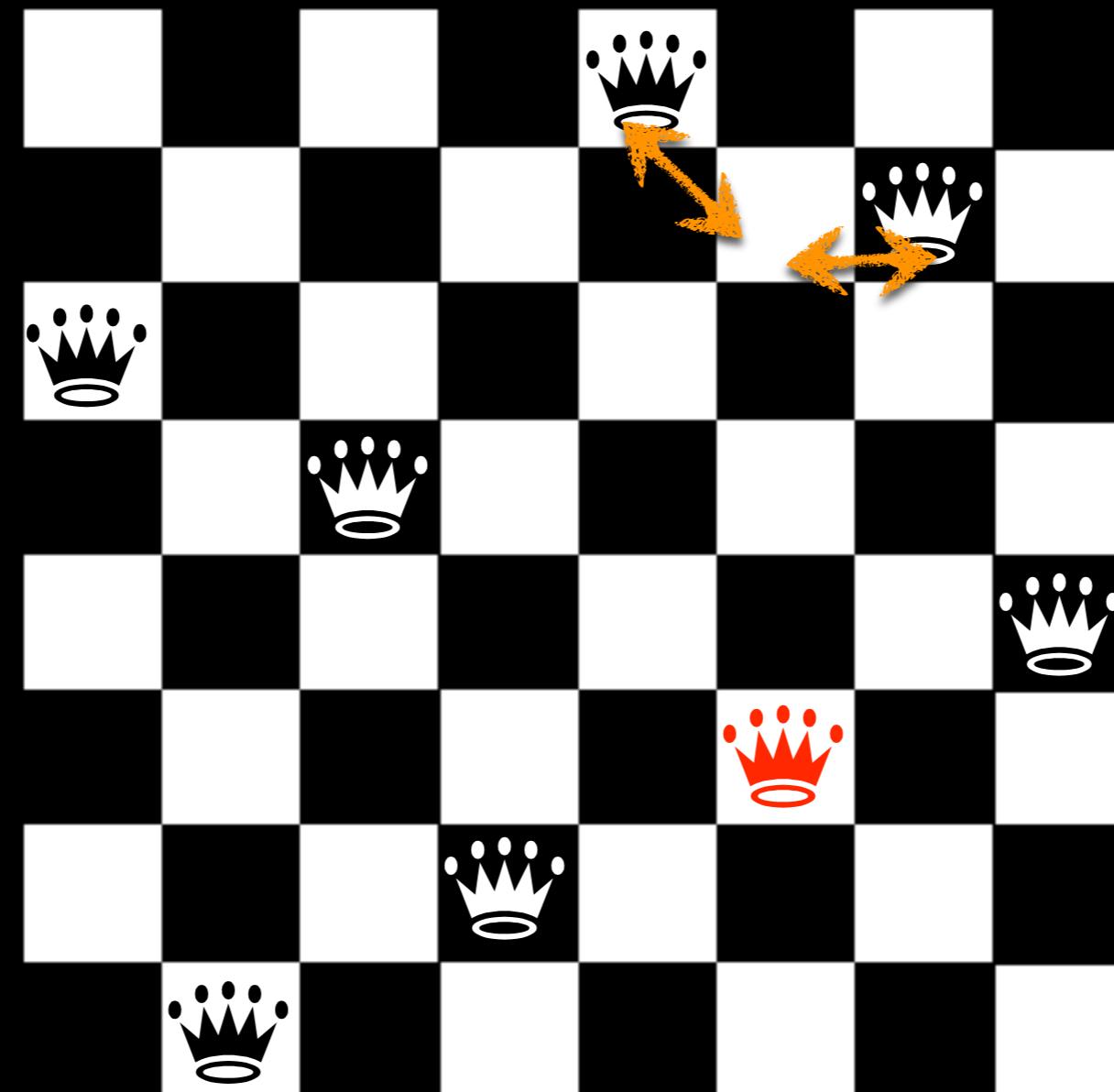
Perfect



Score 0

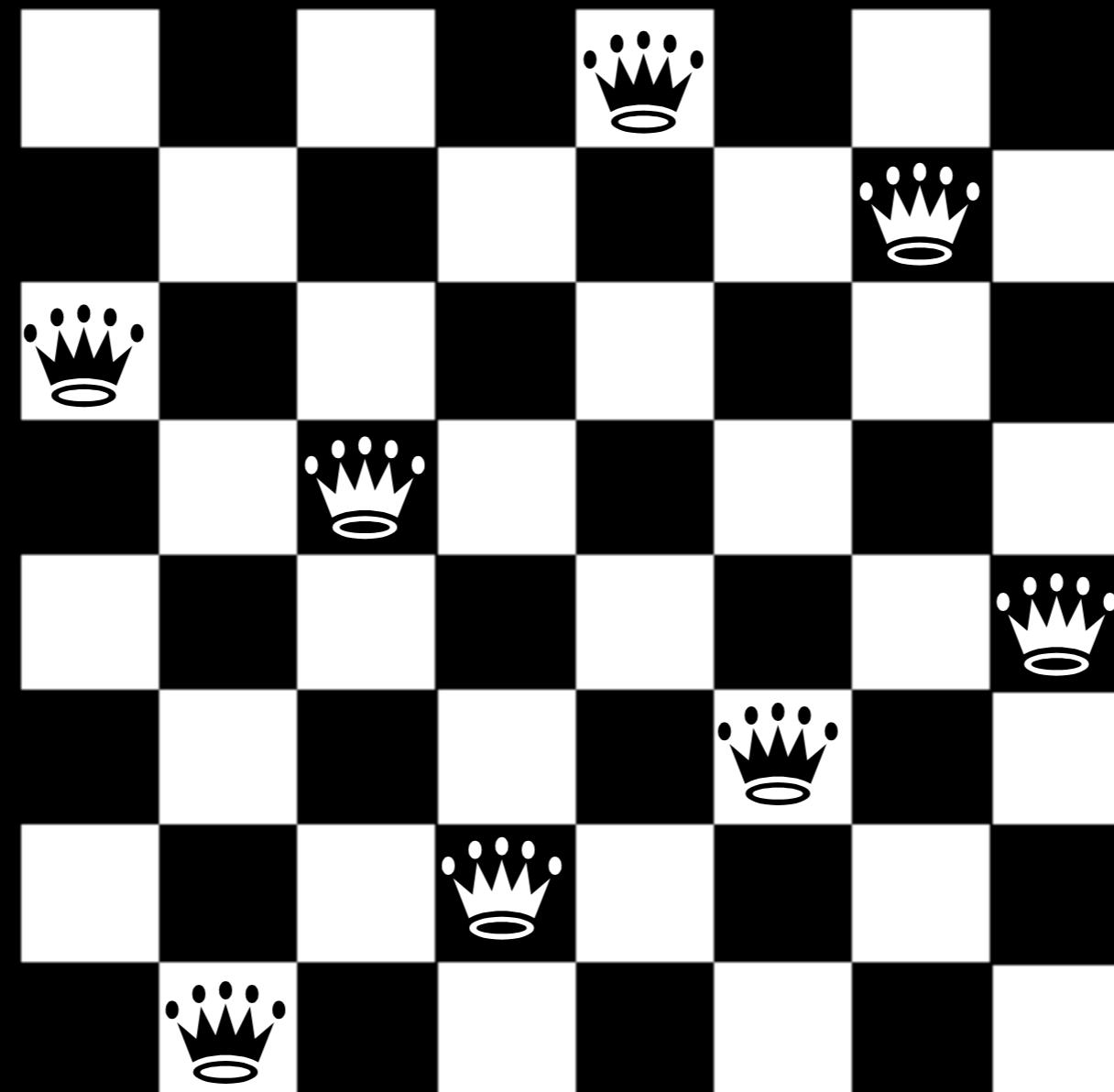
Eight Queens Problem

Two
attacks



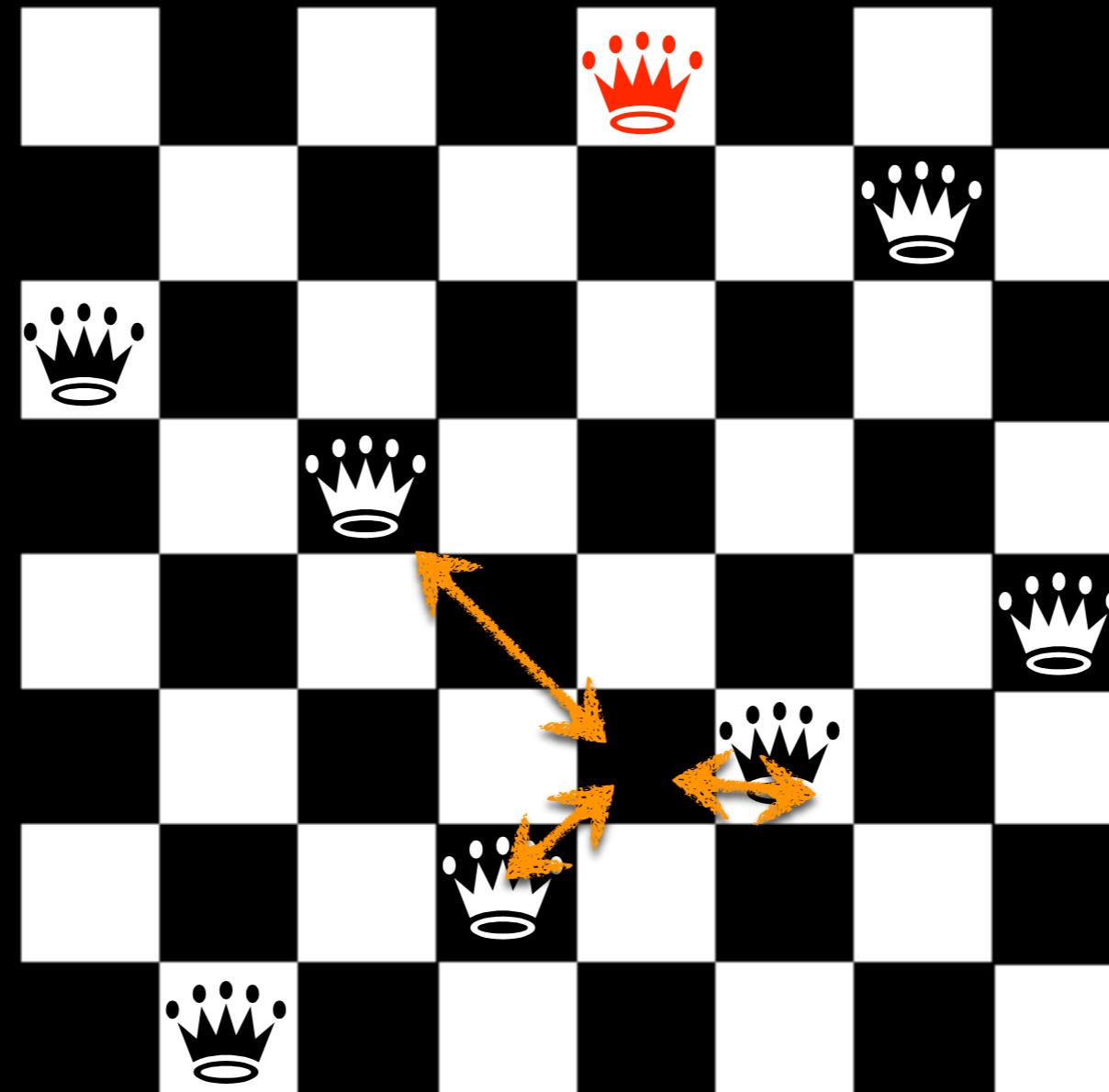
Score -2

Eight Queens Problem



Eight Queens Problem

Three attacks

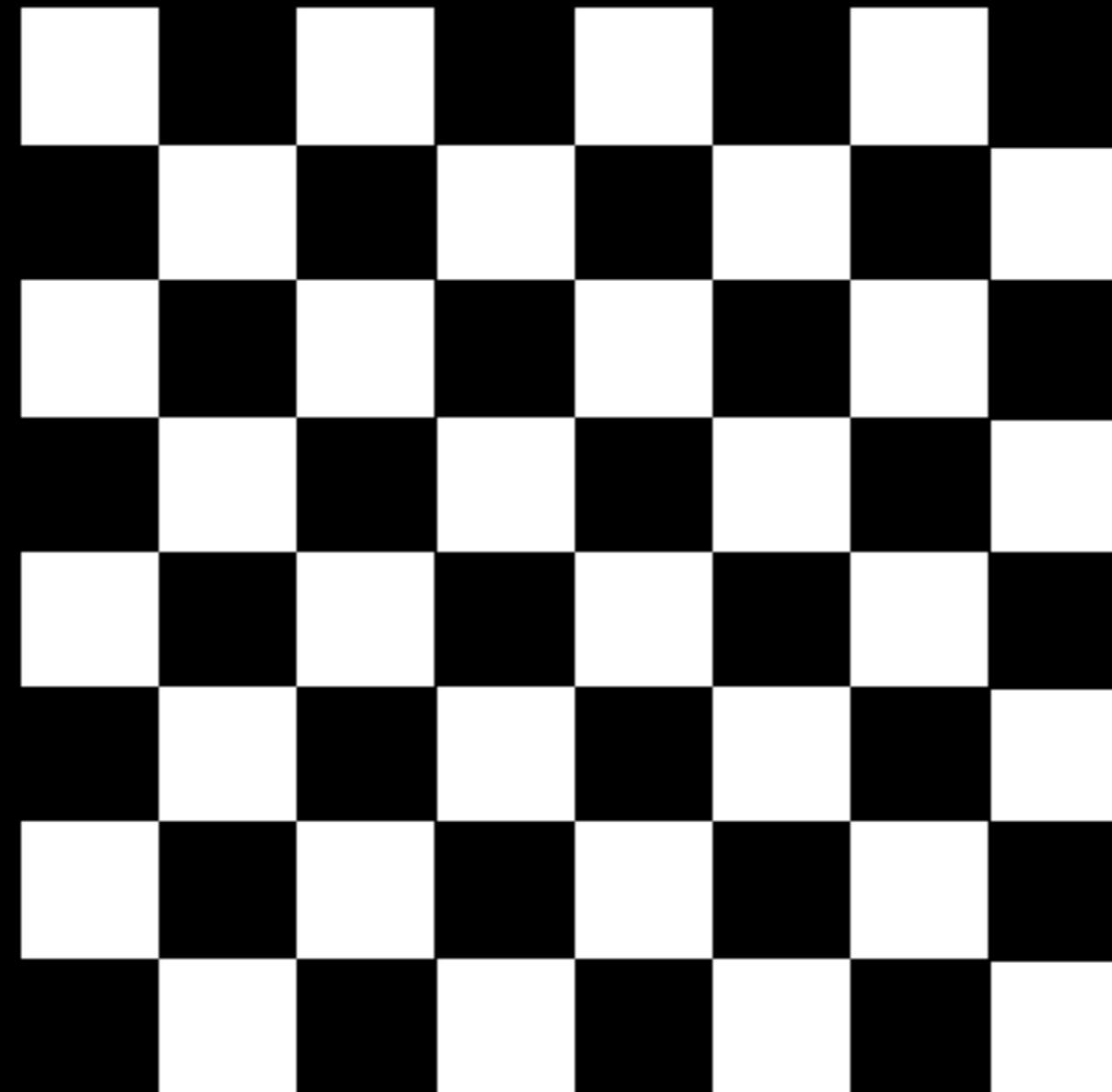


Score -3

That was easy

Eight Queens Problem

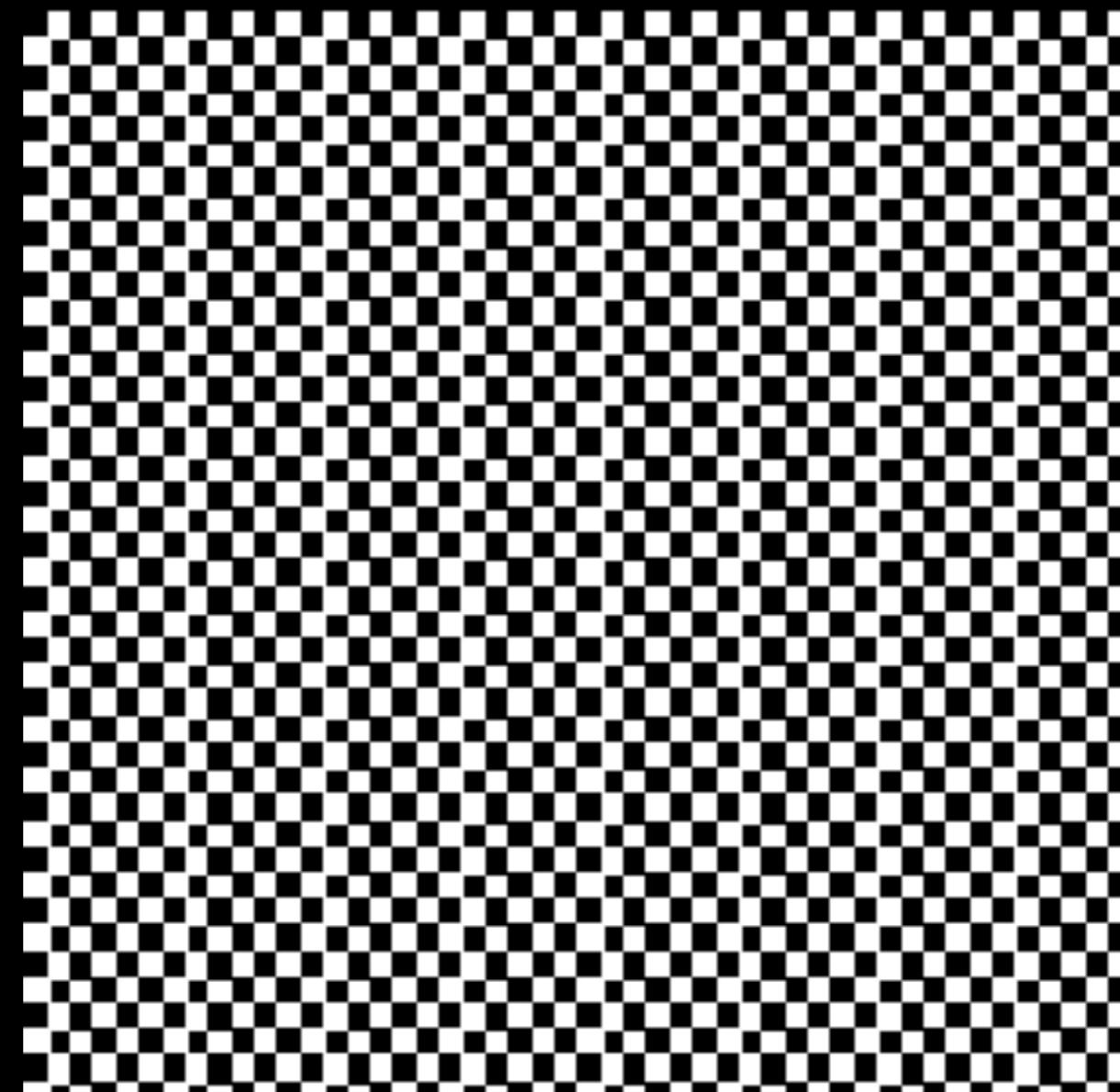
Place 8
queens
on the
board



So that
there are
no
attacks

Eight Queens Problem

Place 44
queens
on the
board



So that
there are
no
attacks

Eight Queens Problem

Place 10^{12}
queens
on the
board



So that
there are
no
attacks

Checking vs Generating

Task One:

Write a method to determine which is the better of two placements of N queens

Task Two:

Write a method to construct a board placement with N non attacking queens

Checking vs Generating

Task One:

Write a method to determine which is the better of two placements of N queens

Task Two:

Write a method to construct a board placement with N non attacking queens

Checking vs Generating

Search Based Software Engineering

Write a method to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a method to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a **method** to determine which is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a **fitness function** to determine which
is the better of two solutions

Conventional Software Engineering

Write a method to construct a perfect
solution

Checking vs Generating

Search Based Software Engineering

Write a **fitness function** to guide a search

Conventional Software Engineering

Write a method to construct a perfect solution

Checking vs Generating

Search Based Software Engineering

Write a **fitness function** to guide **automated search**

Conventional Software Engineering

Write a method to construct a perfect solution

SBSE for PM

How much do we need to know to build effective models?

How to adapt models to new data?

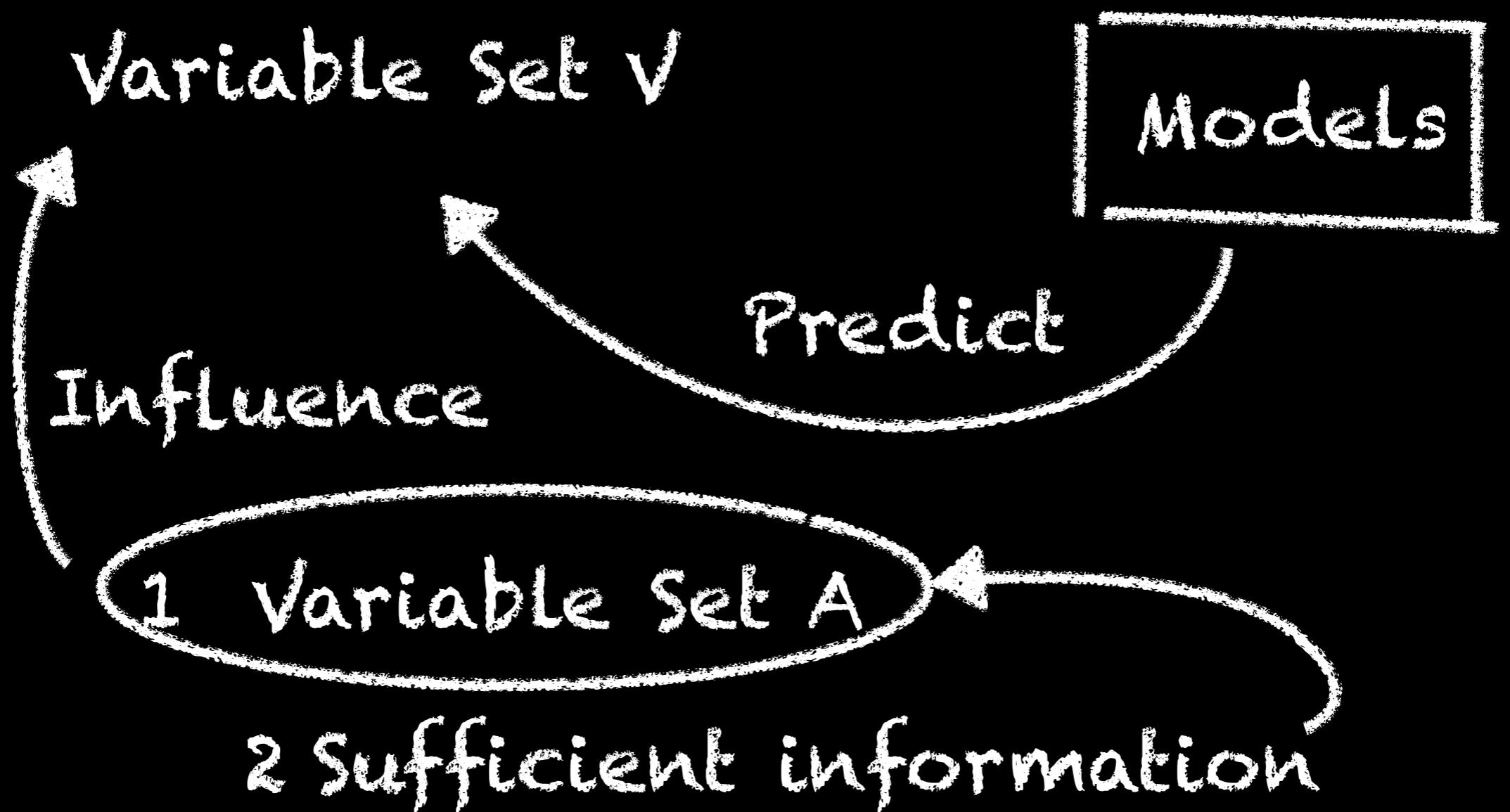
How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

Previous work on SBSE for PM



Build Effective Models



Build Effective Models

Variable Set V

“Which of the candidate variables’ values can affect the values of the variable we seek to predict?”

Influence

1 Variable Set A

Multi Objective Search

Many problems have multiple objectives

Often we have many metrics

Recent SBSE work has been multi objective

Pareto optimality can yield insight

Multi Objective Search

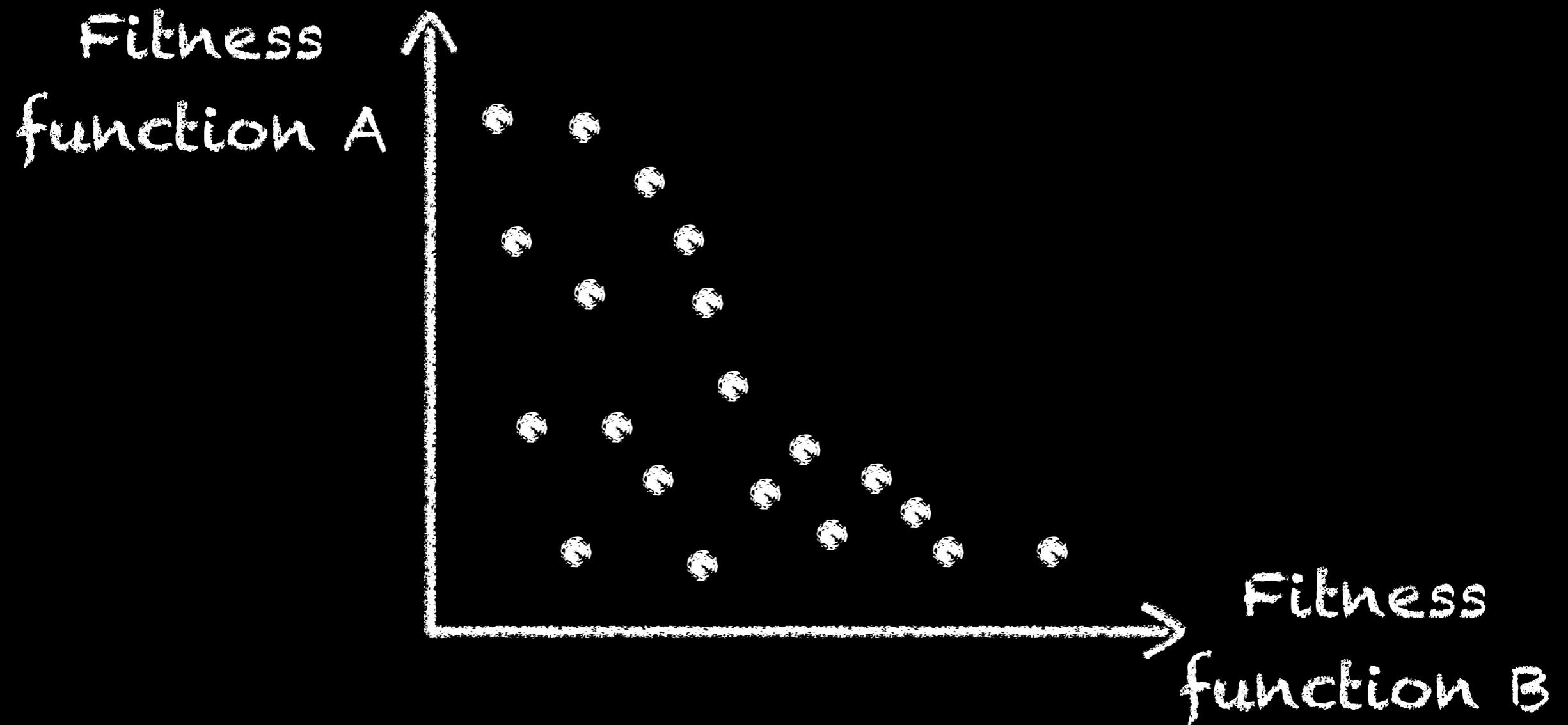
Many problems have multiple objectives

Often we have many metrics

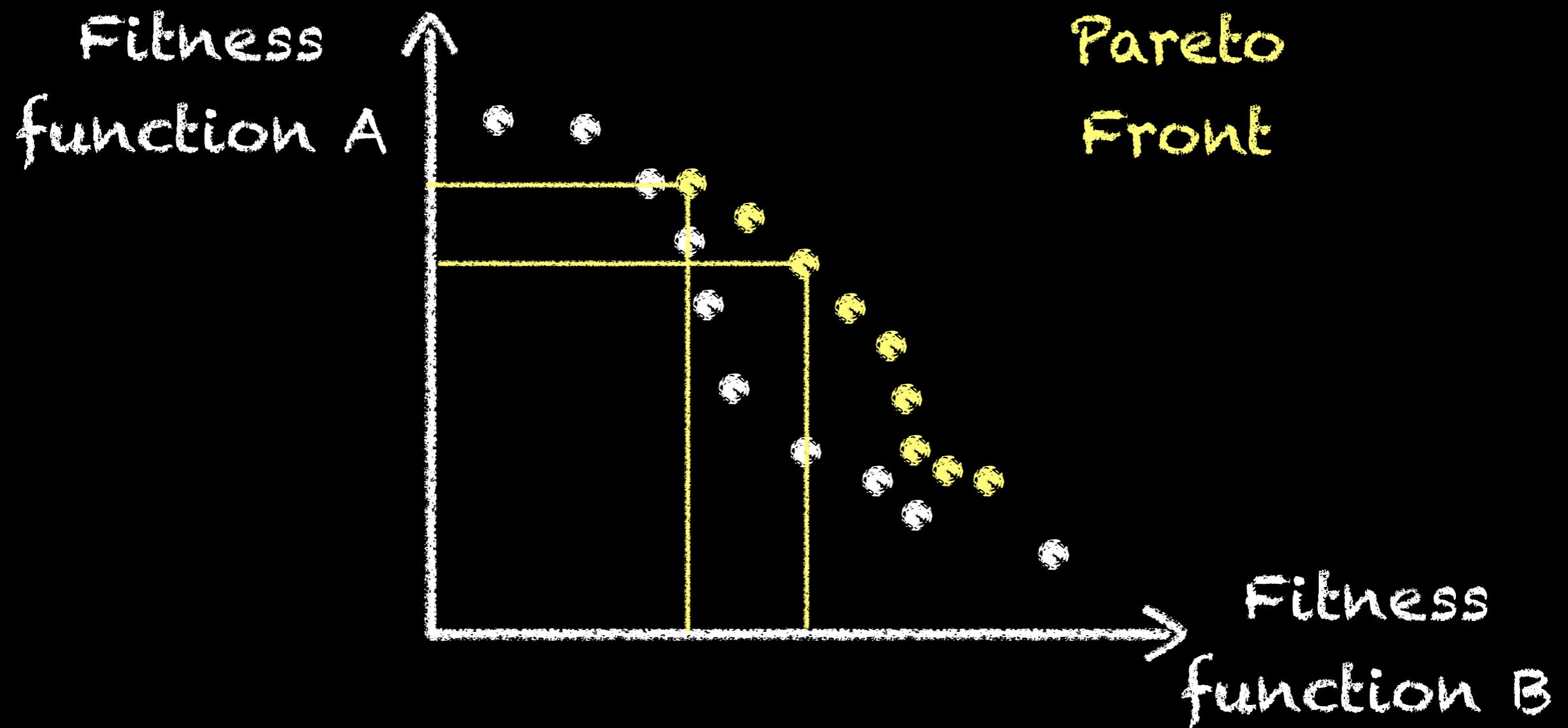
Recent SBSE work has been multi objective

Pareto optimality can yield insight

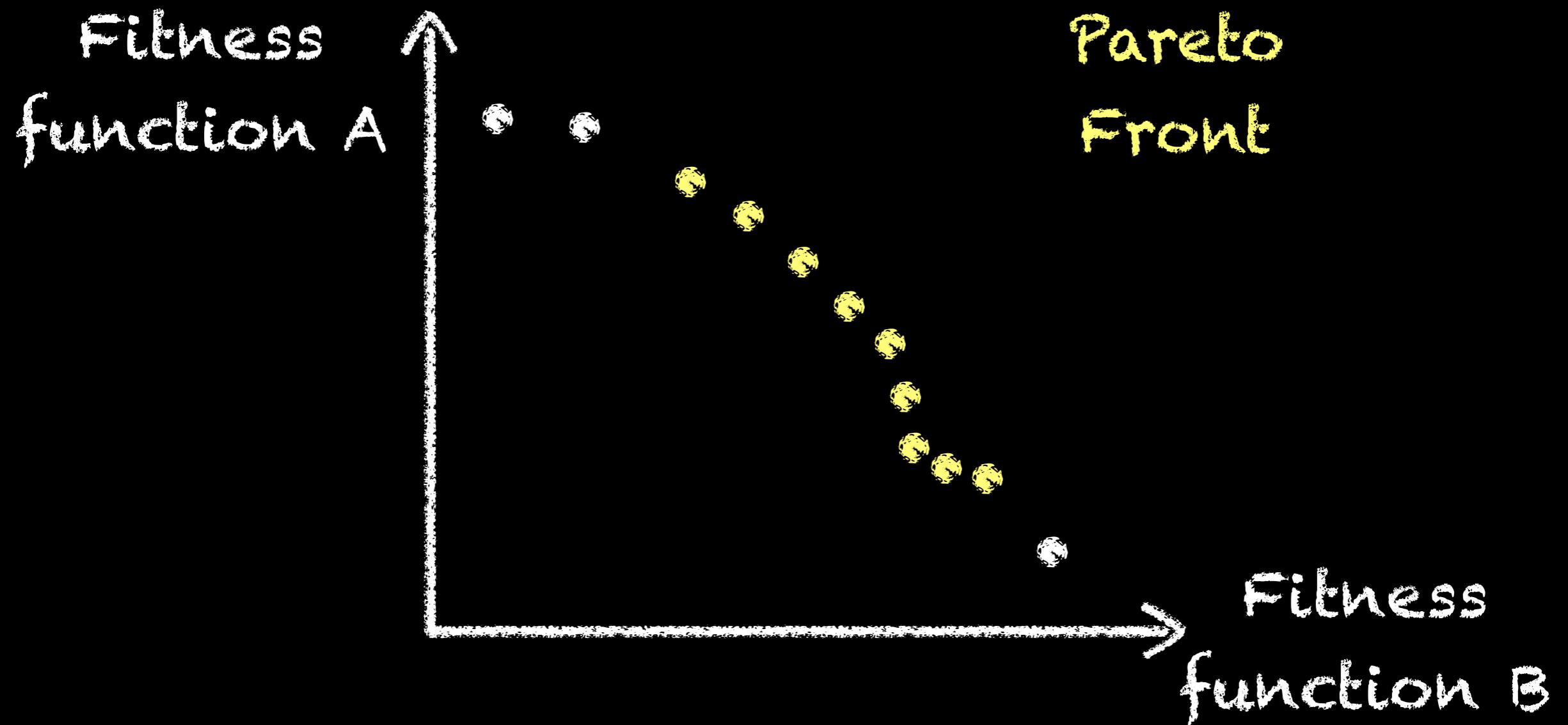
Multi Objective Search



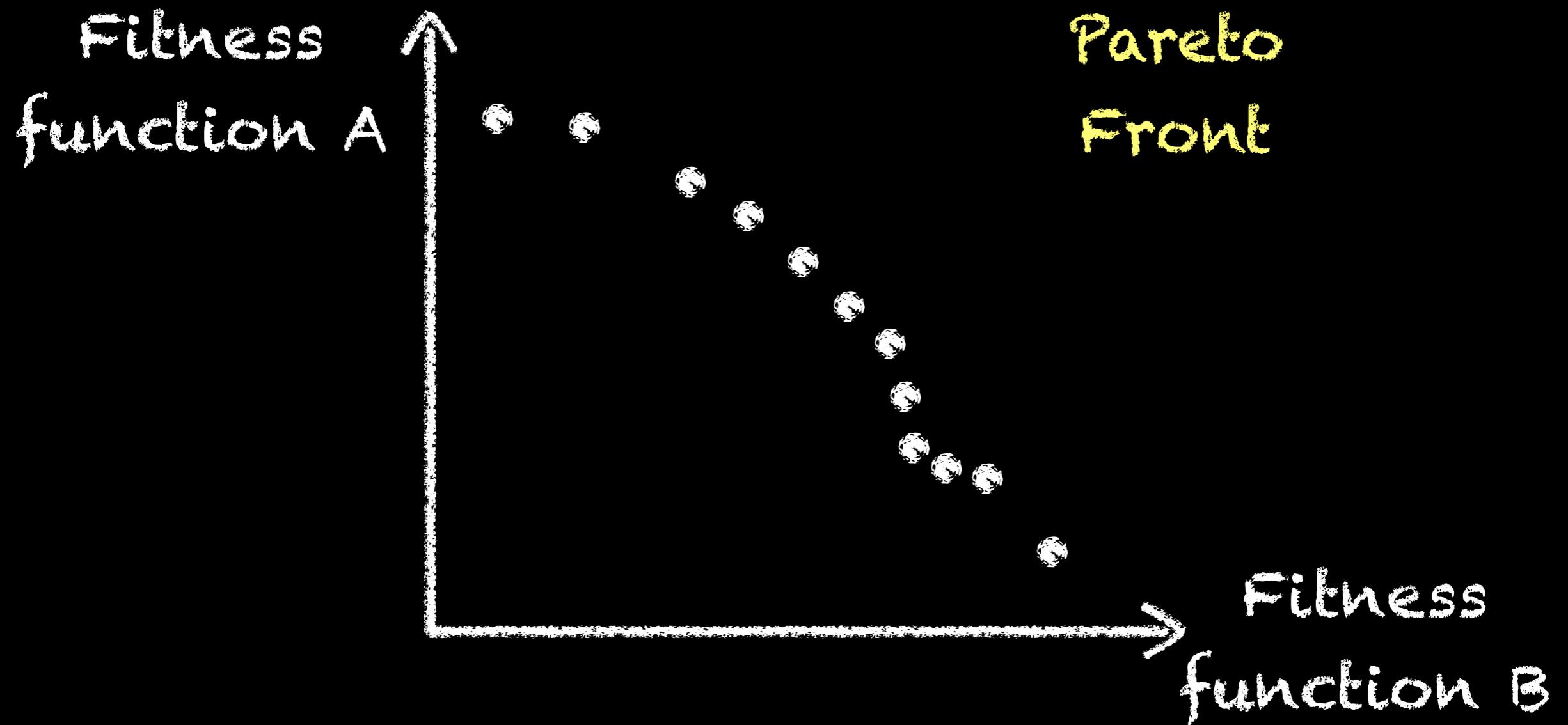
Multi Objective Search



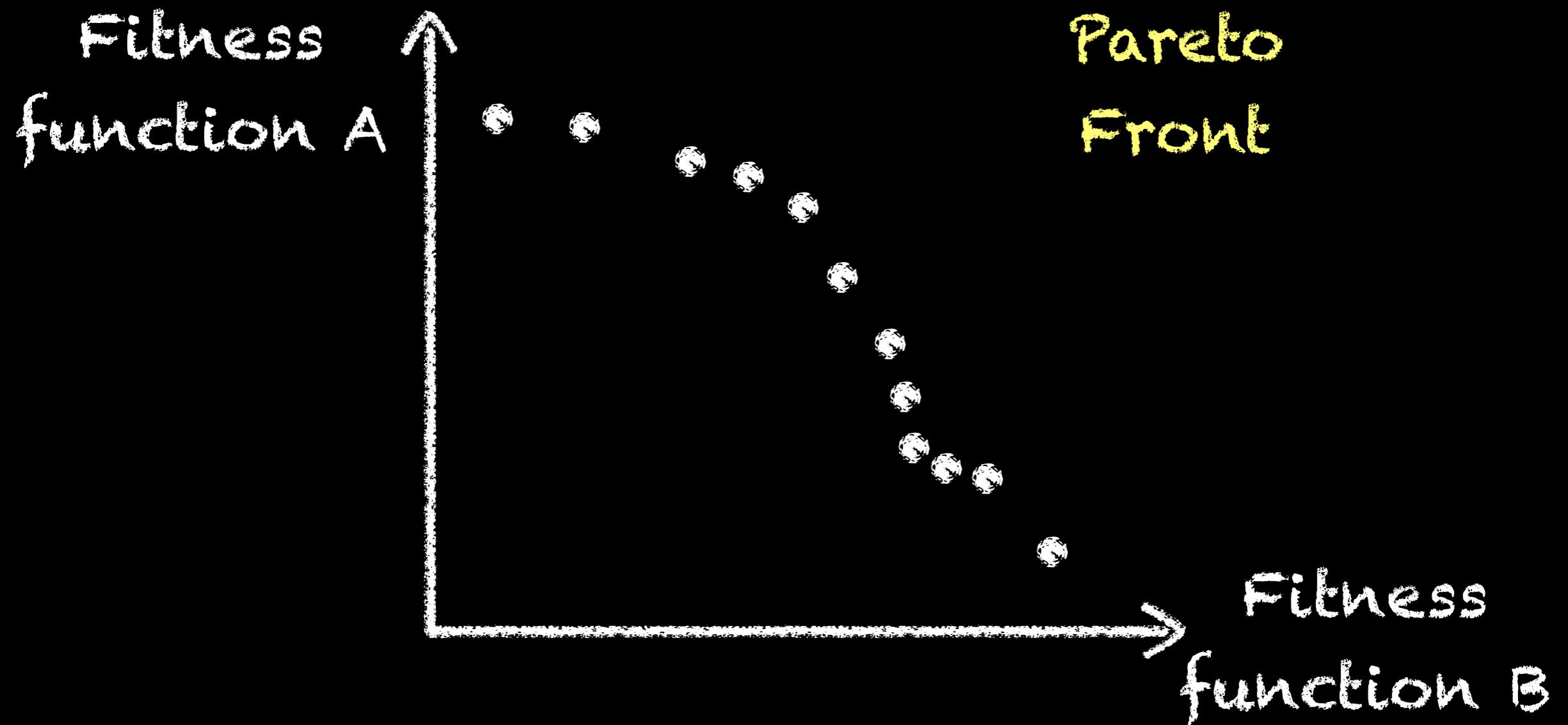
Multi Objective Search



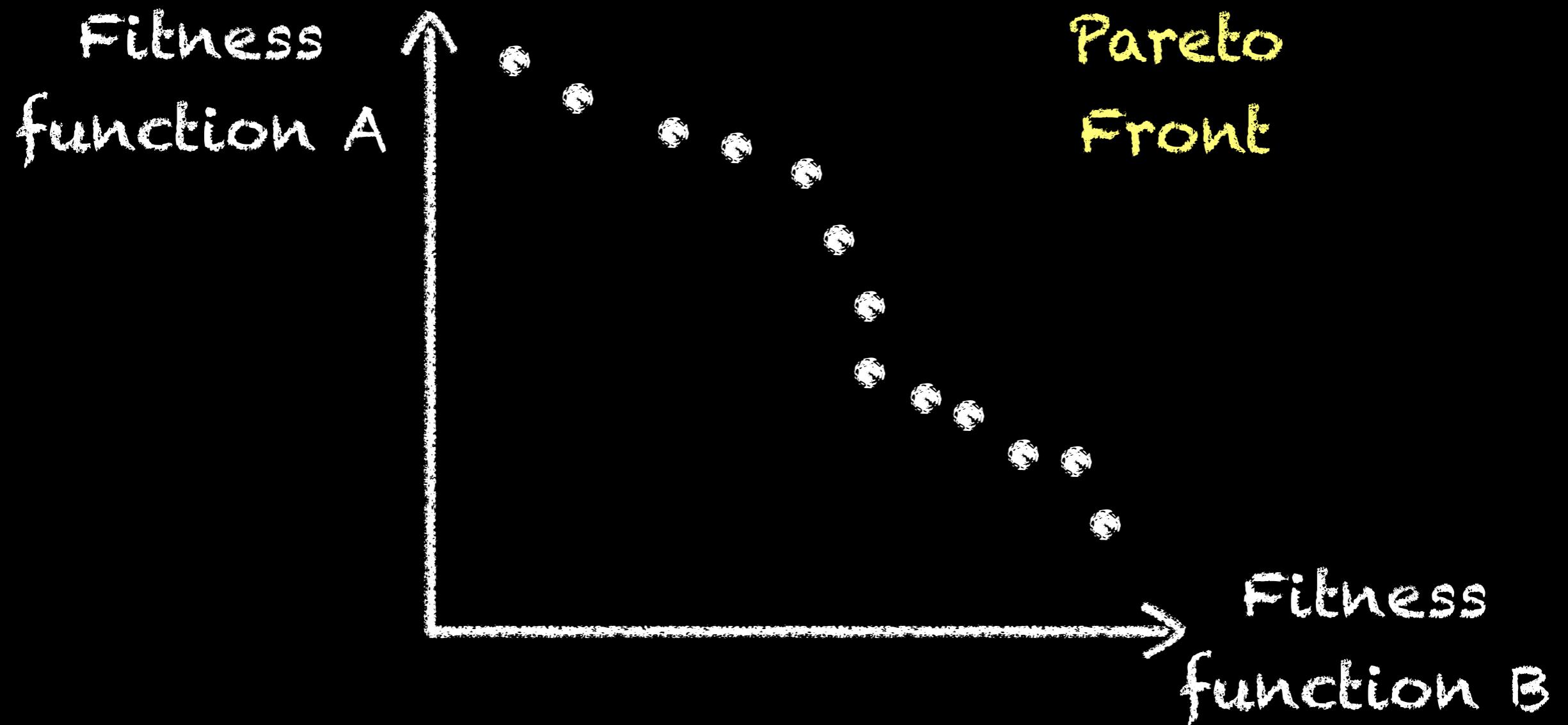
Multi Objective Search



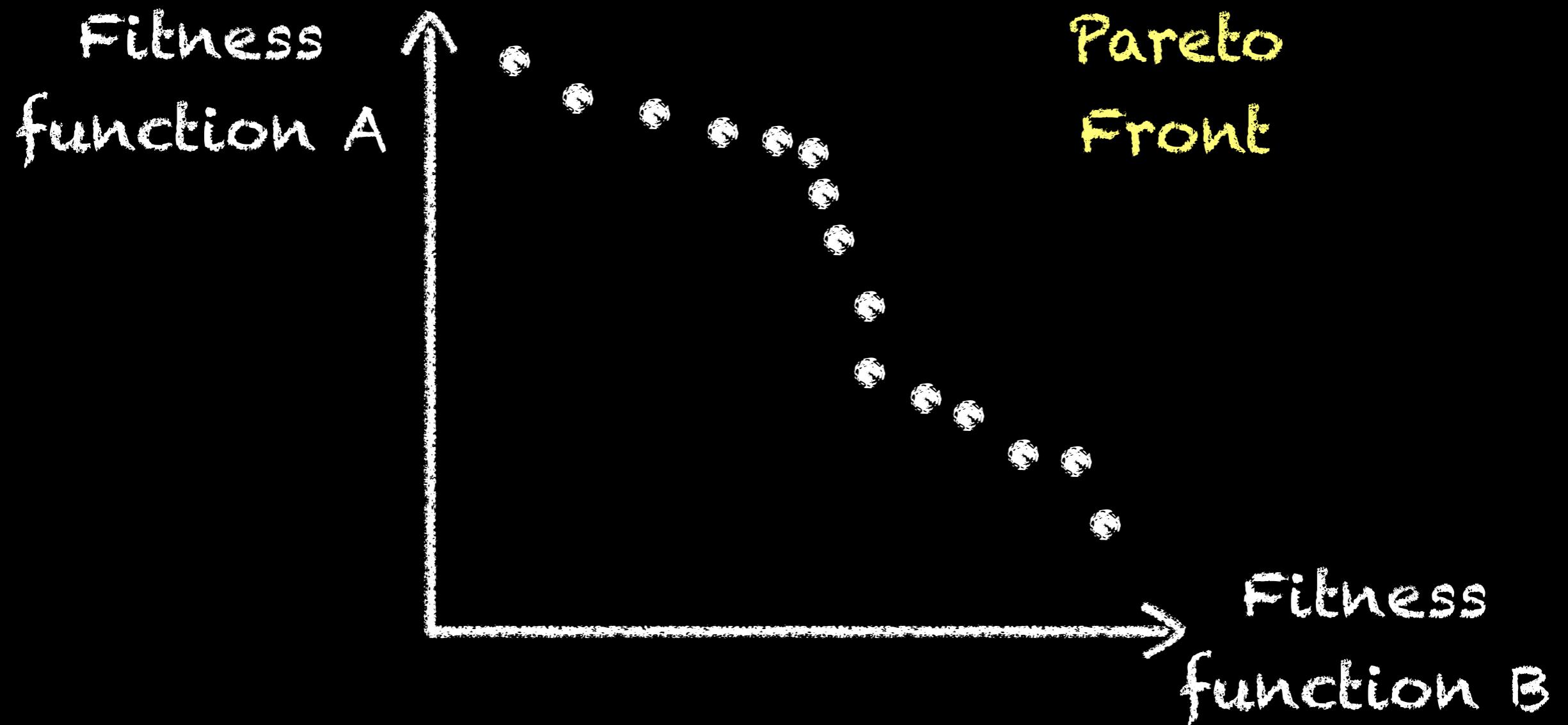
Multi Objective Search



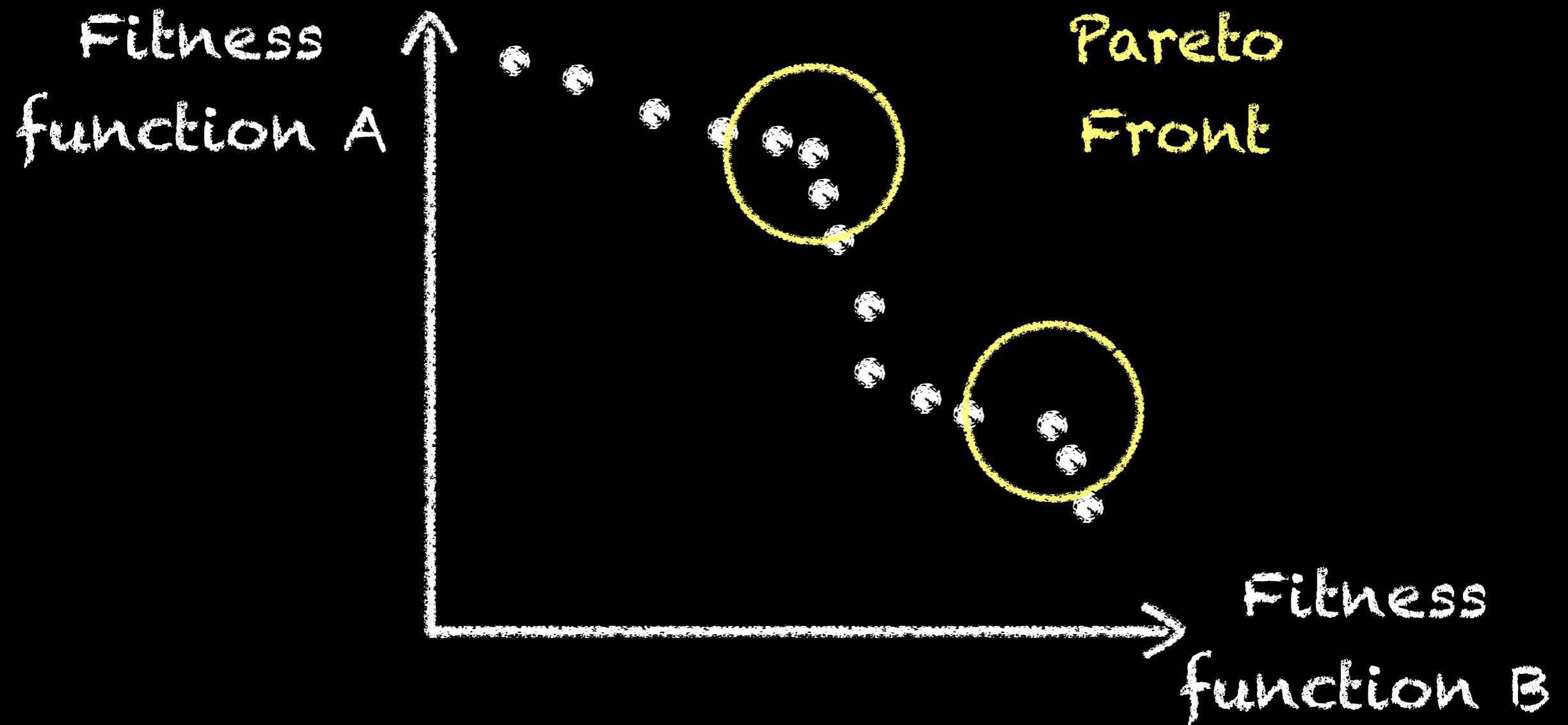
Multi Objective Search



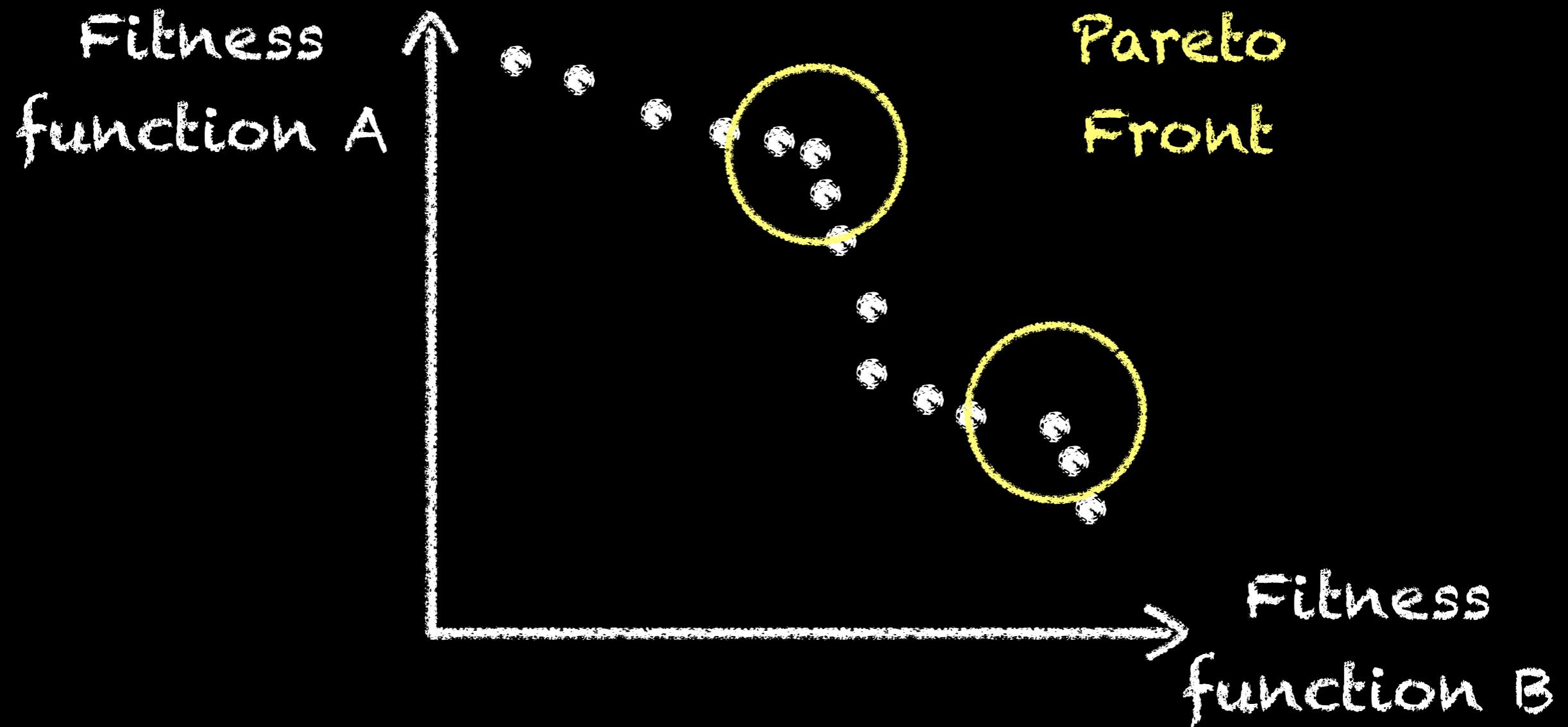
Multi Objective Search



Multi Objective Search



Direction of front growth



Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Regression Testing

Requirements Analysis

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Regression Testing

Requirements Analysis

Capture
requirements

Generate
tests

Explore
designs

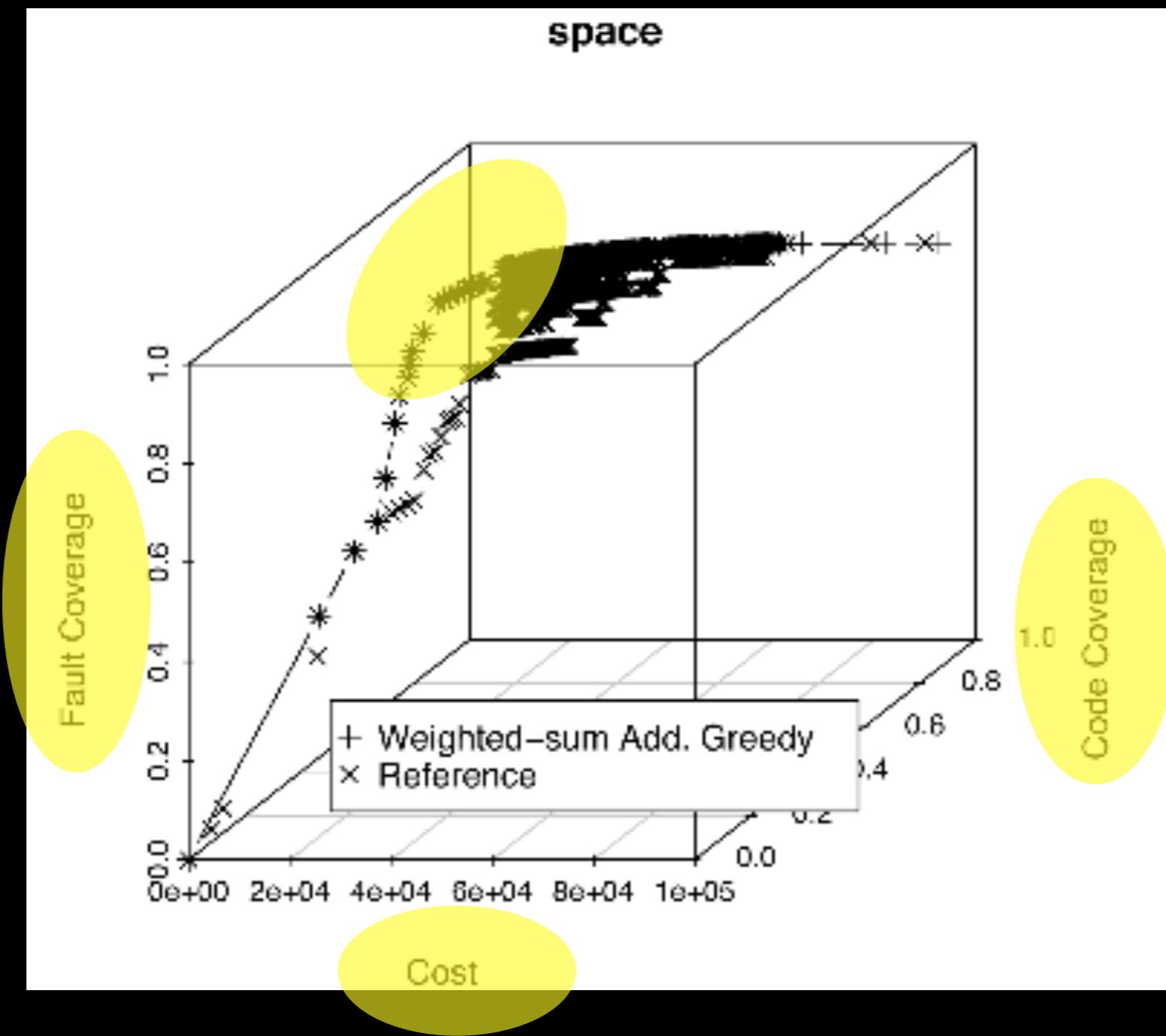
Maintain
evolve

Regression
test

Regression Testing

Requirements Analysis

Regression Test Selection



Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

Regression
test

Regression Testing

Requirements Analysis

Capture
requirements

Generate
tests

Explore
designs

Maintain
evolve

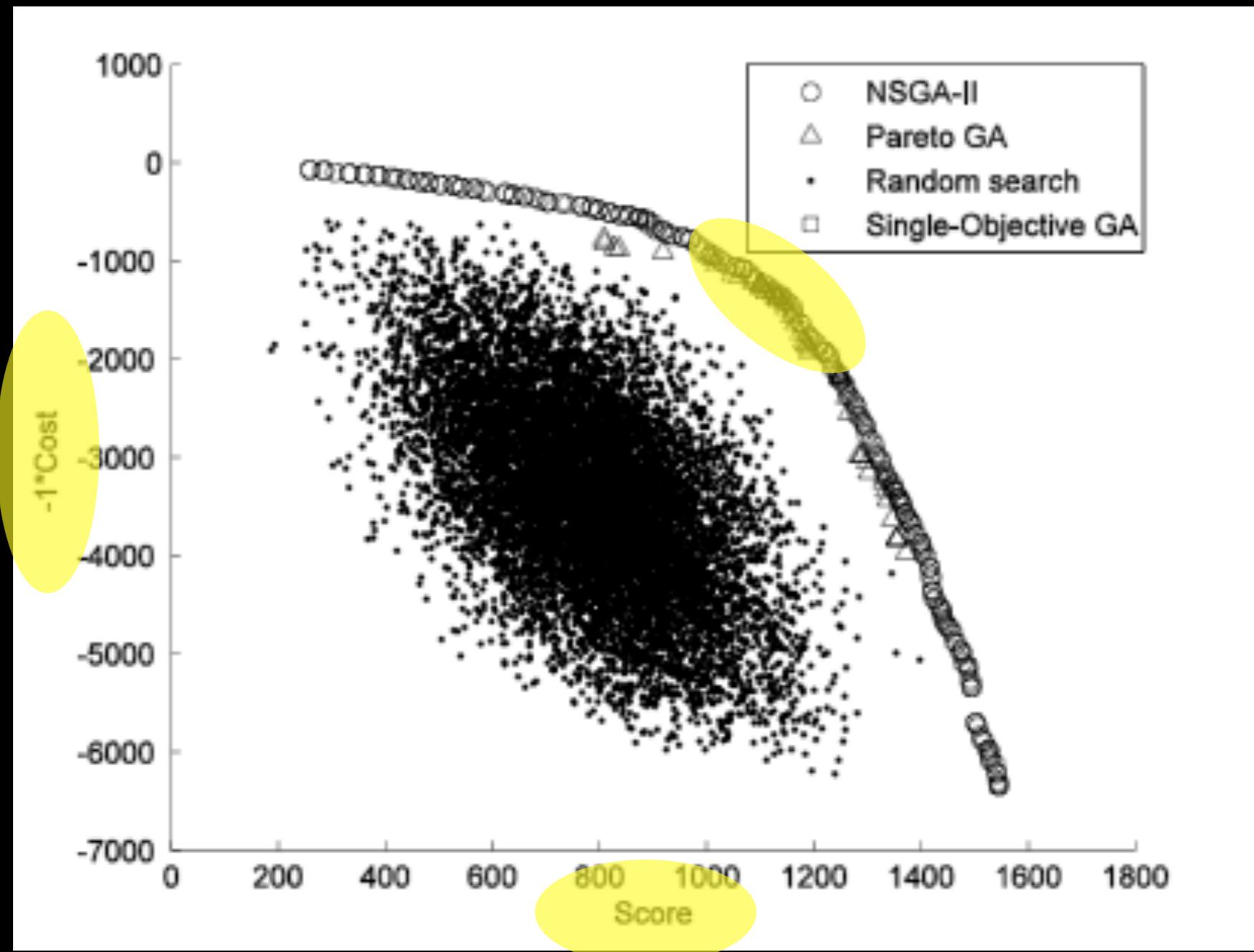
Regression
test

Regression Testing

Requirements Analysis



Motorola Cell Phone Requirements



SBSE is so generic

Requirements

Fitness function: cost, value ...

Representation: bitset of requirements

Regression

Fitness function: coverage, time, faults

Representation: bitset of test cases

SBSE is so generic

Requirements

Fitness function: cost, value ...

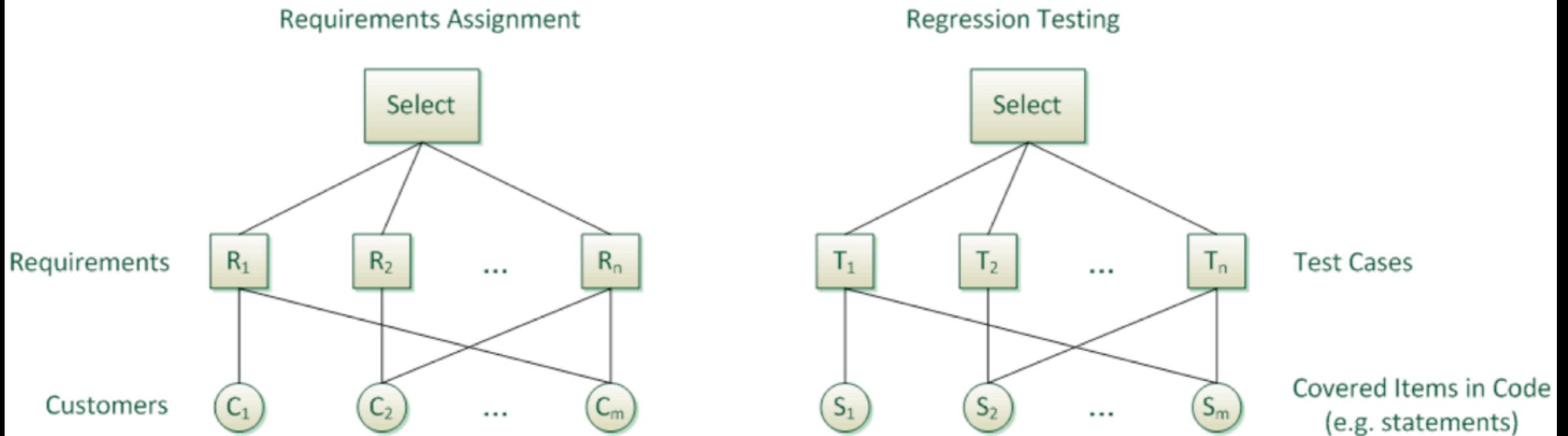
Representation: bitset of requirements

Regression

Fitness function: coverage, time, faults

Representation: bitset of test cases

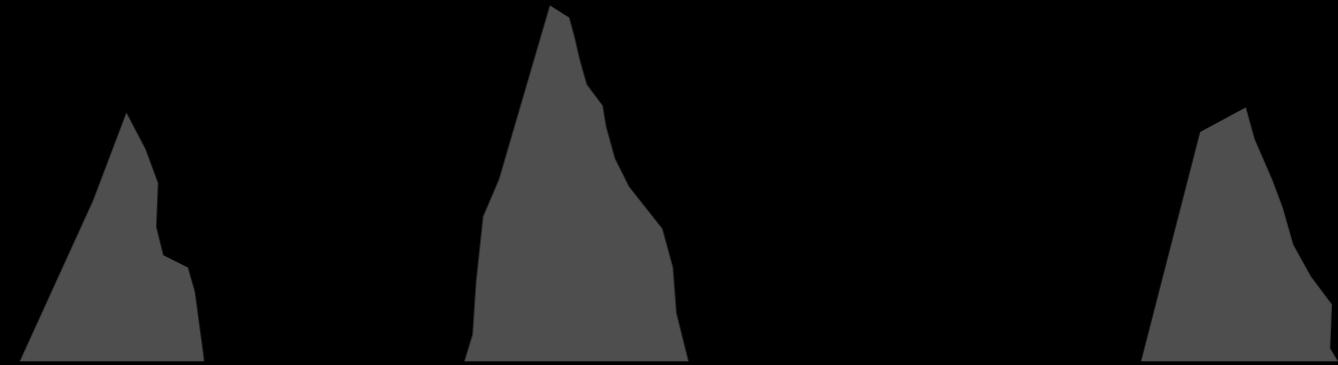
Selection Problems



Prioritization Problems



Requirements and regression testing: really different ?



Al o n e

SBSE for PM

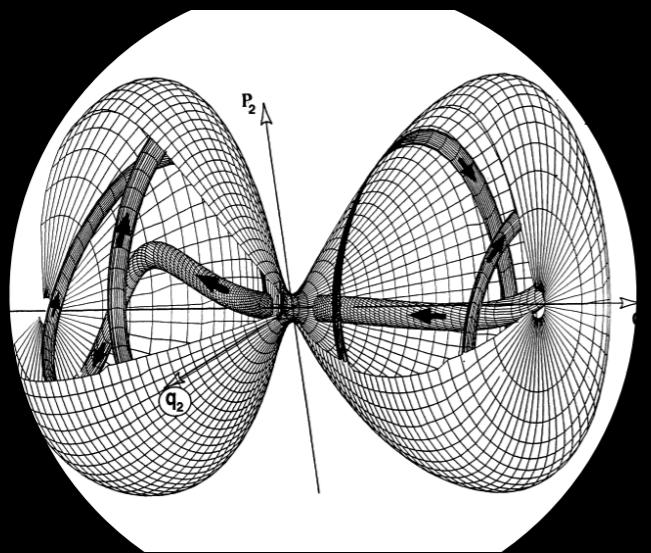
How much do we need to know to build effective models?

How to adapt models to new data?

How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

Multi Objective Search



Quality



Cost



Privacy

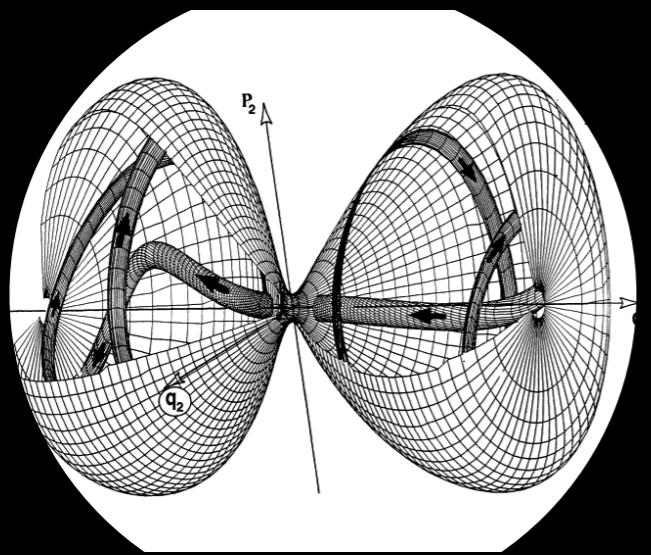


Readability



Coverage and Weighting

Multi Objective Search



Quality



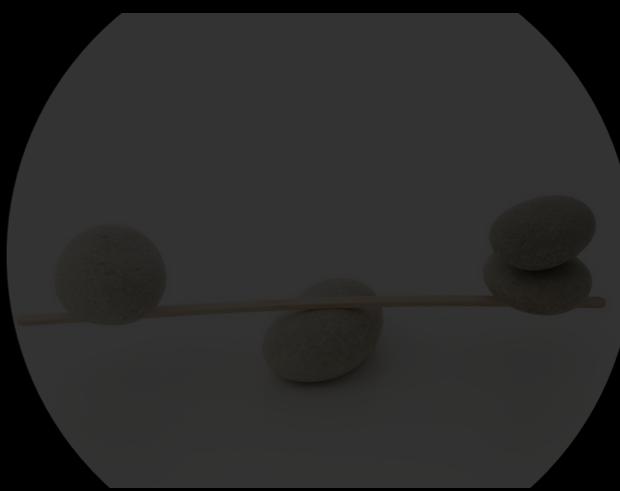
Cost



Privacy

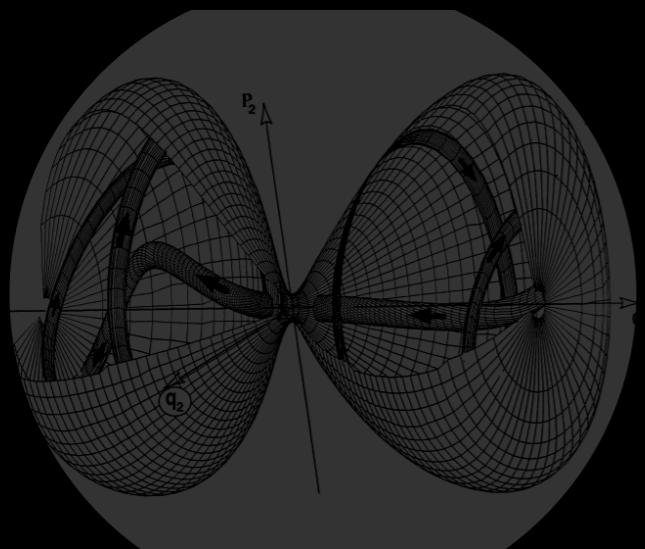


Readability



Coverage and Weighting

Multi Objective Search



Quality



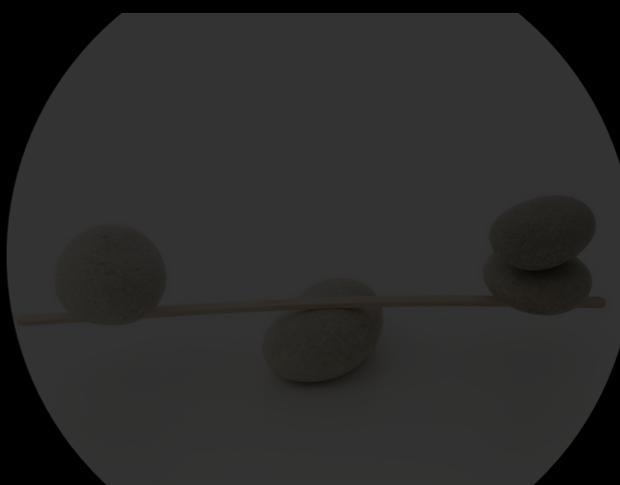
Cost



Privacy

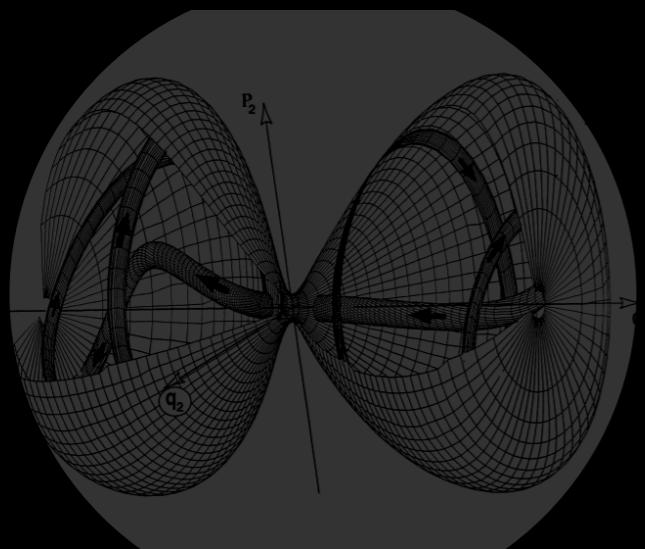


Readability



Coverage and Weighting

Multi Objective Search



Quality



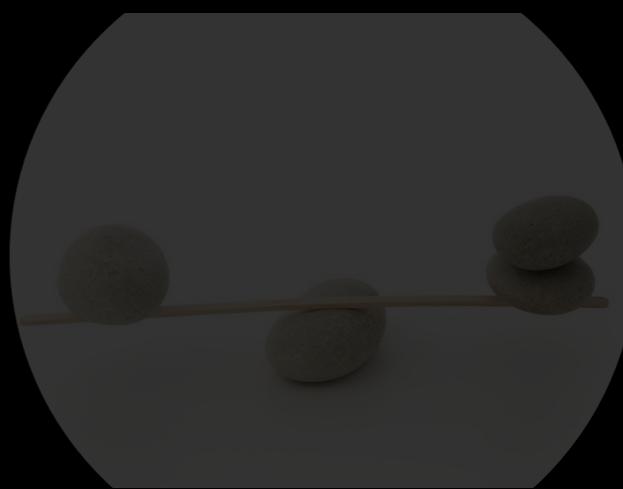
Cost



Privacy

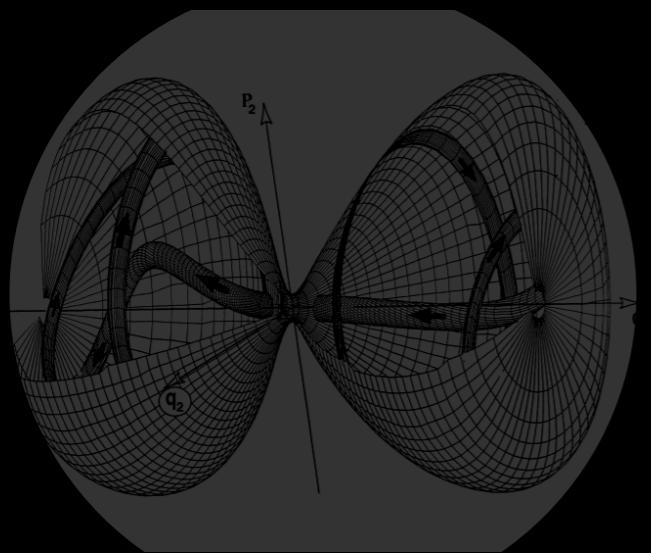


Readability



Coverage and Weighting

Multi Objective Search



Quality



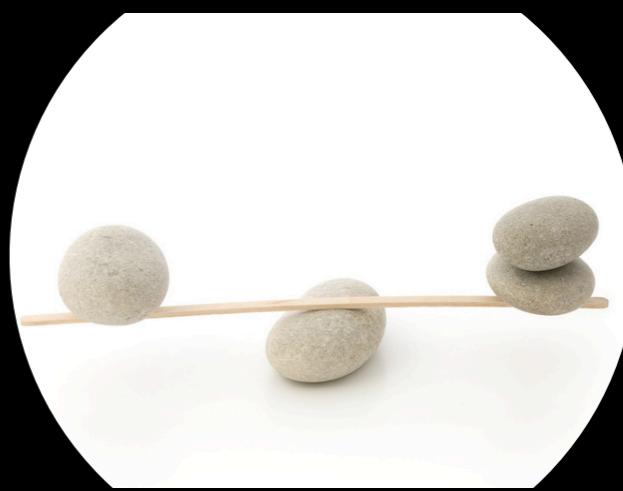
Cost



Privacy

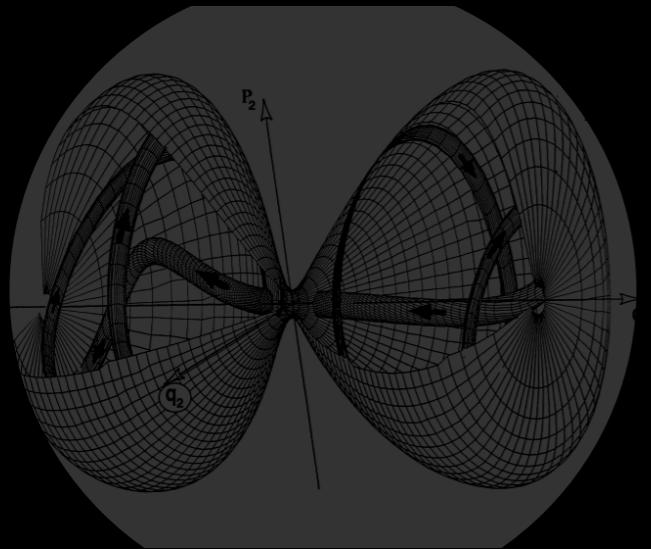


Readability



Coverage and Weighting

Multi Objective Search



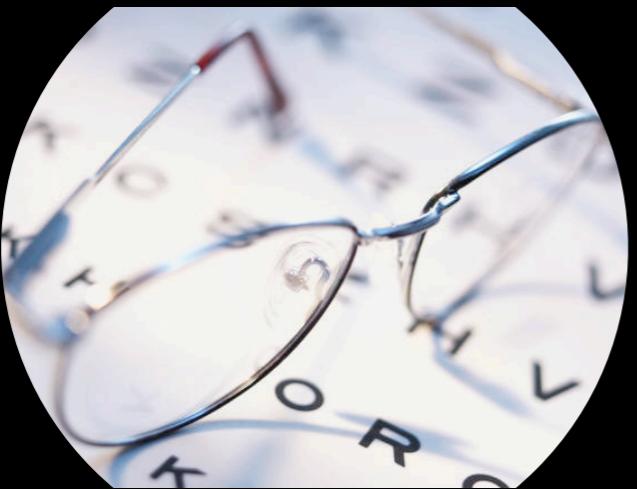
Quality



Cost



Privacy



Readability
suggested by Tim Menzies

Mark Harman

PROMISE'10 keynote



Coverage and Weighting

SBSE for PM

How much do we need to know to build effective models?

How to adapt models to new data?

How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

Adapt Models to New Data

Existing Data

VS

New data

Which is better ?

Adaptive Evolution

Bayesian Model

Search for good Bayesian nets and allow priors to play a role in optimization

Estimation of Distribution (EDA)

Bayesian Evolutionary Algorithms (BEA)

Bayesian Optimisation Algorithms (BOA)

SBSE for PM

How much do we need to know to build effective models?

How to adapt models to new data?

How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

Streamline the Data or the Process

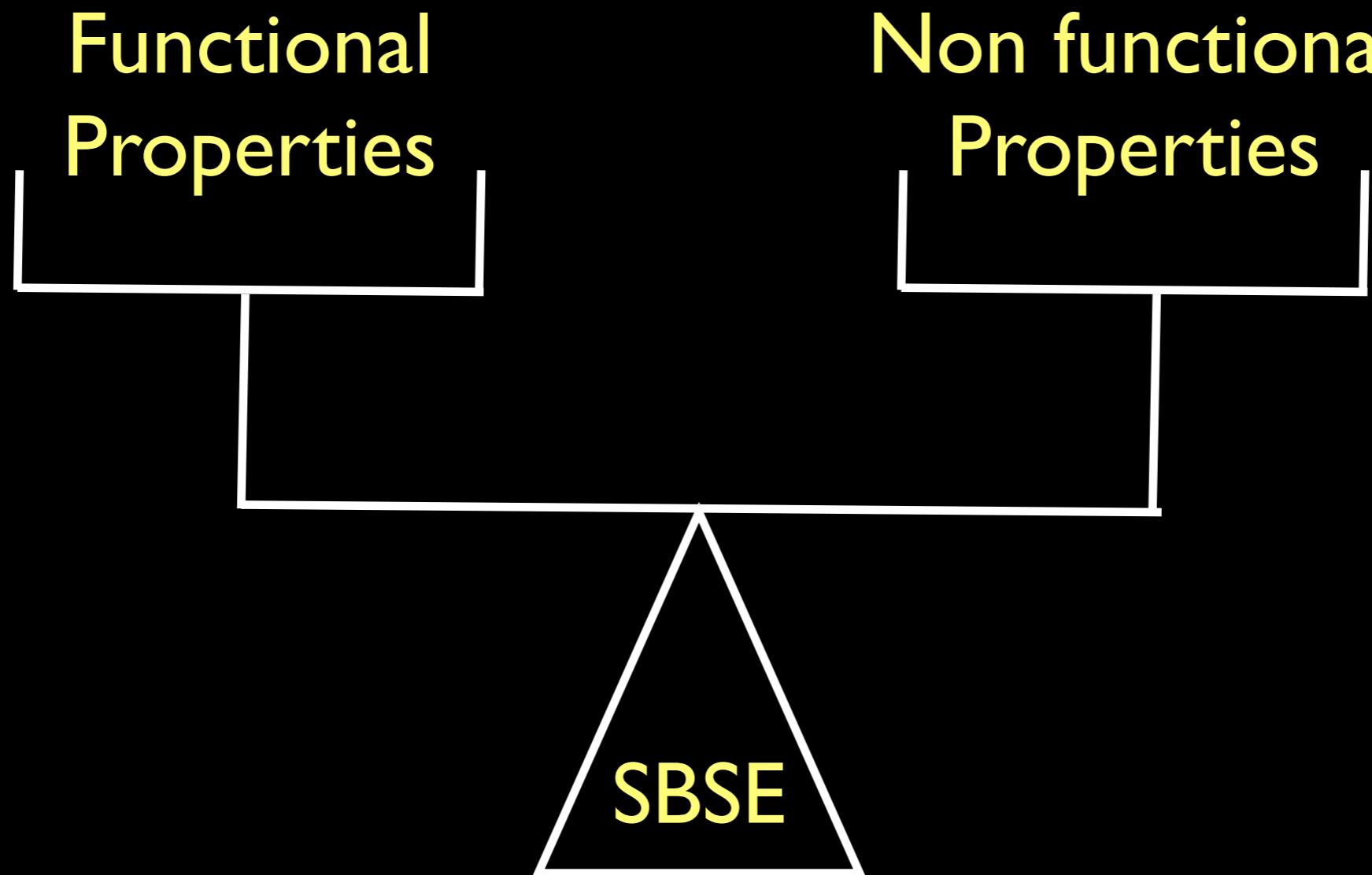
Functional Properties

Predictive quality
Cost
Characteristics

Non functional Properties

Temporal aspects in
Software testing and
Stress testing

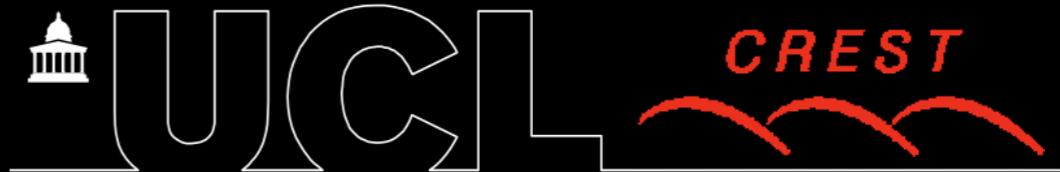
Streamline the Data or the Process



Work by David White

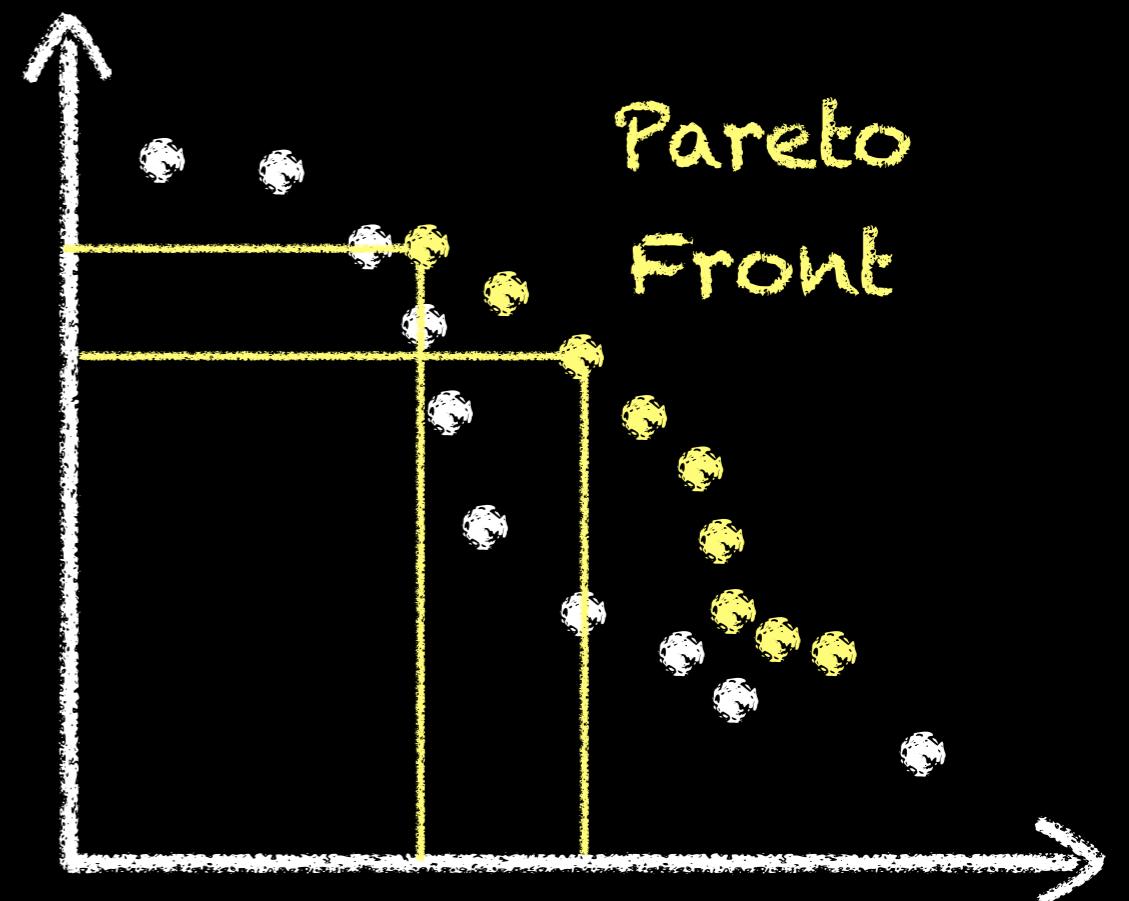
University of York ...

Mark Harman
PROMISE'10 keynote



Streamline the Data or the Process

Randomness
VS
Power consumption

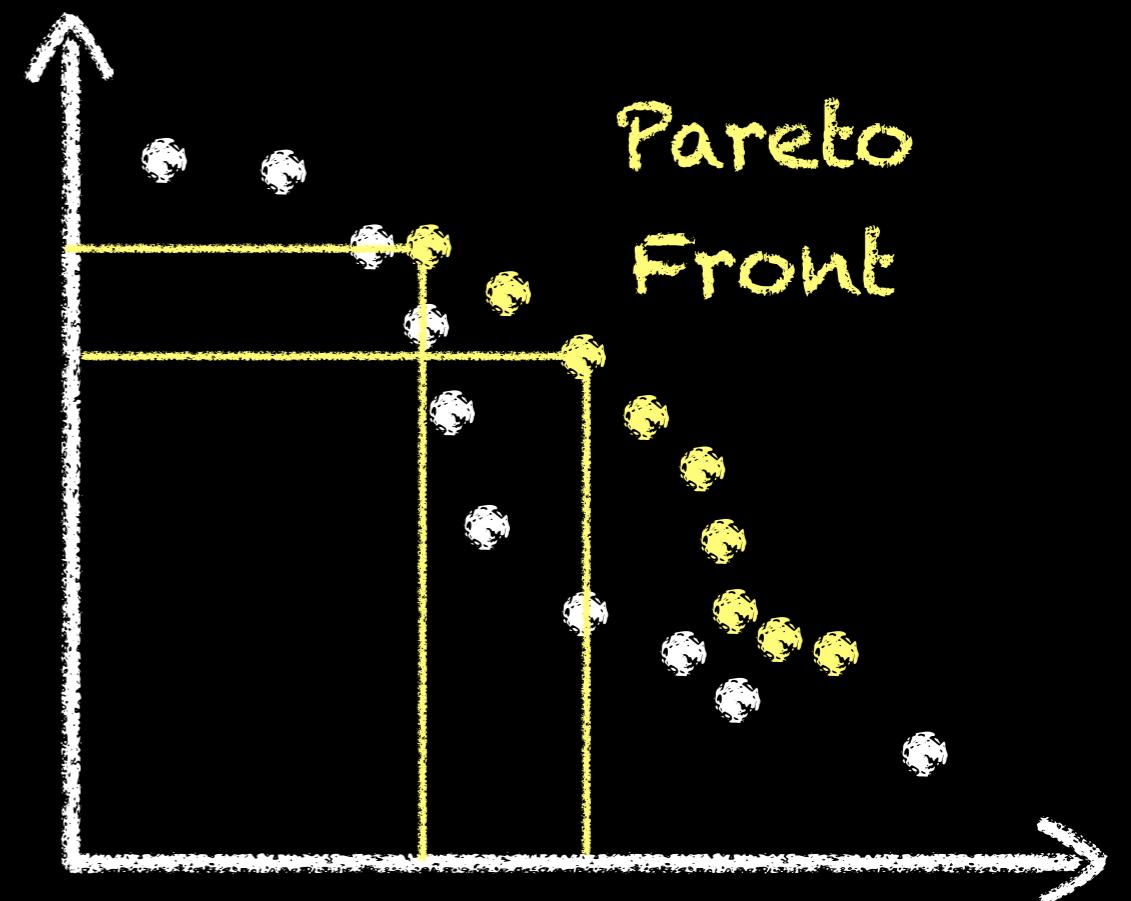


Streamline the Data or the Process

Model quality

VS

Execution Time

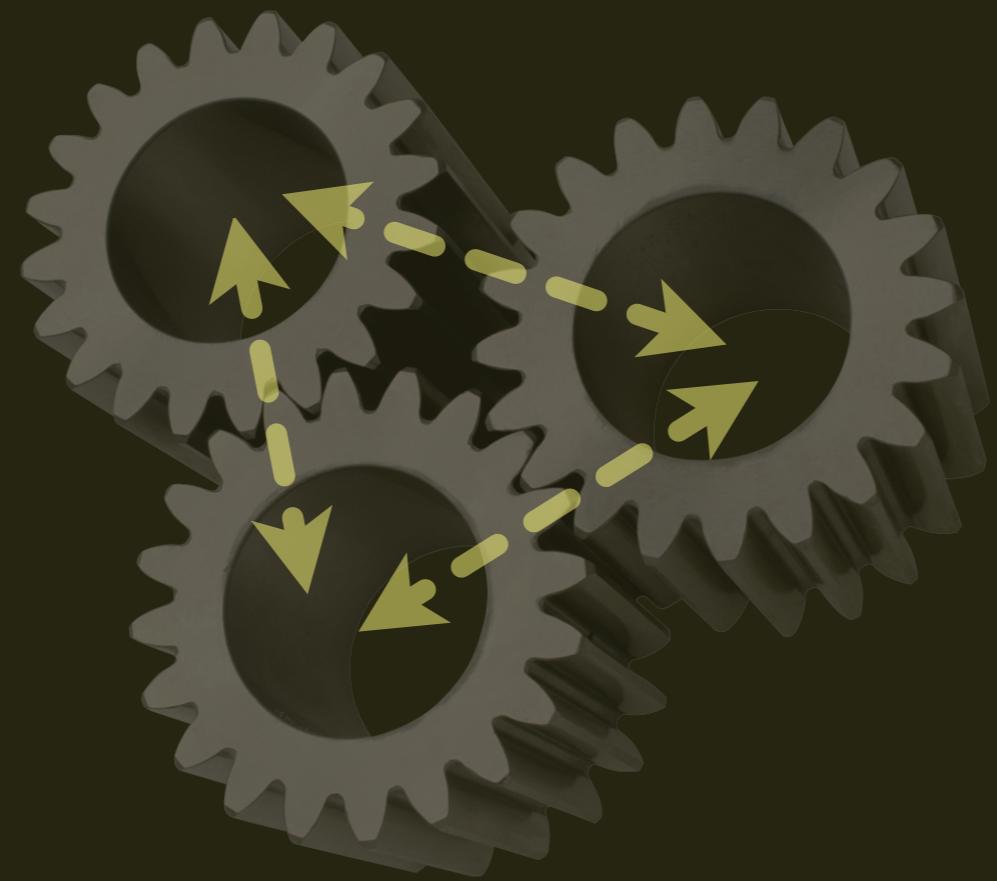


Streamline the Data or the Process



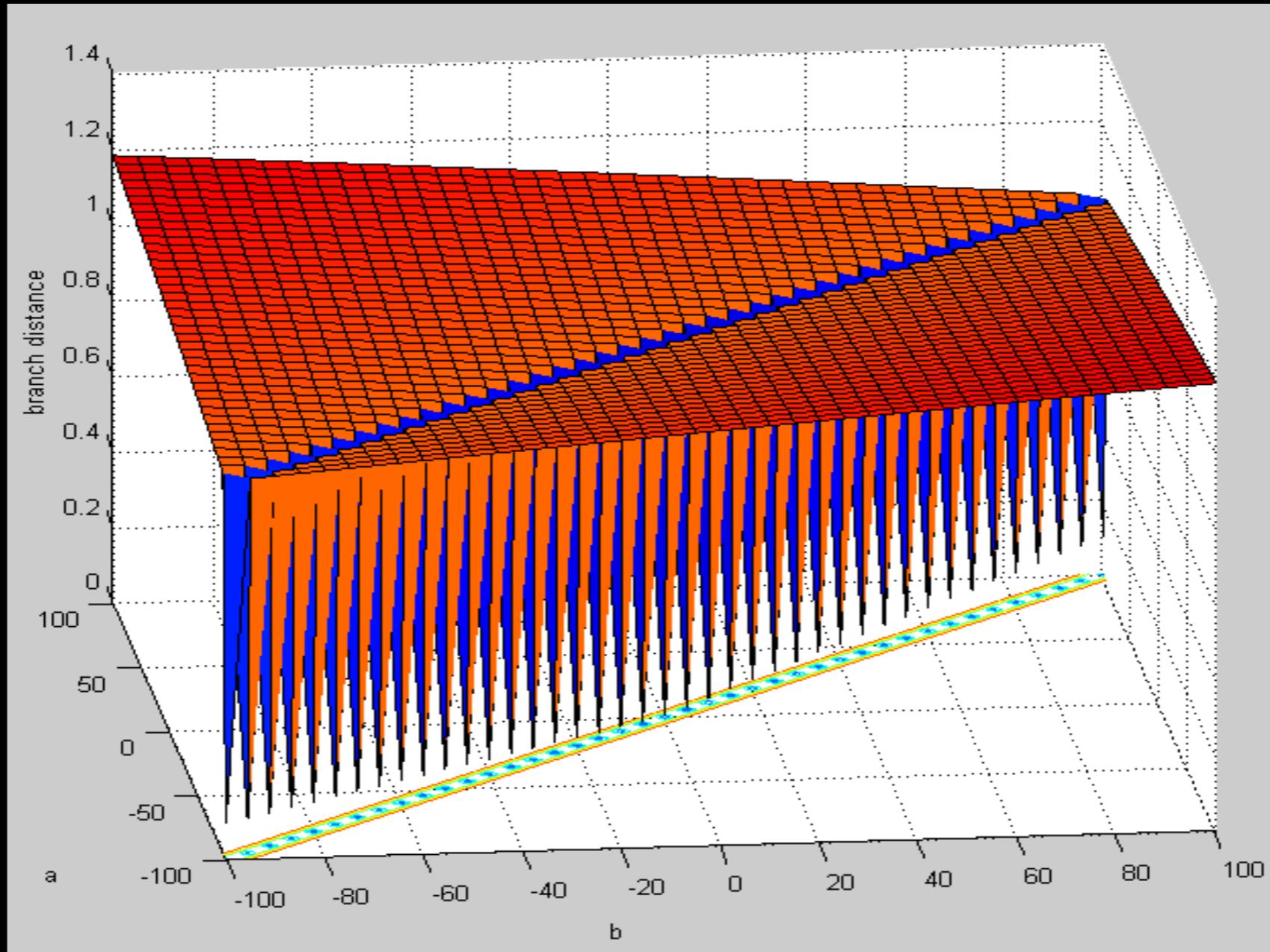
Solution

Streamline the Data or the Process



Insight

Streamline the Data or the Process



SBSE for PM

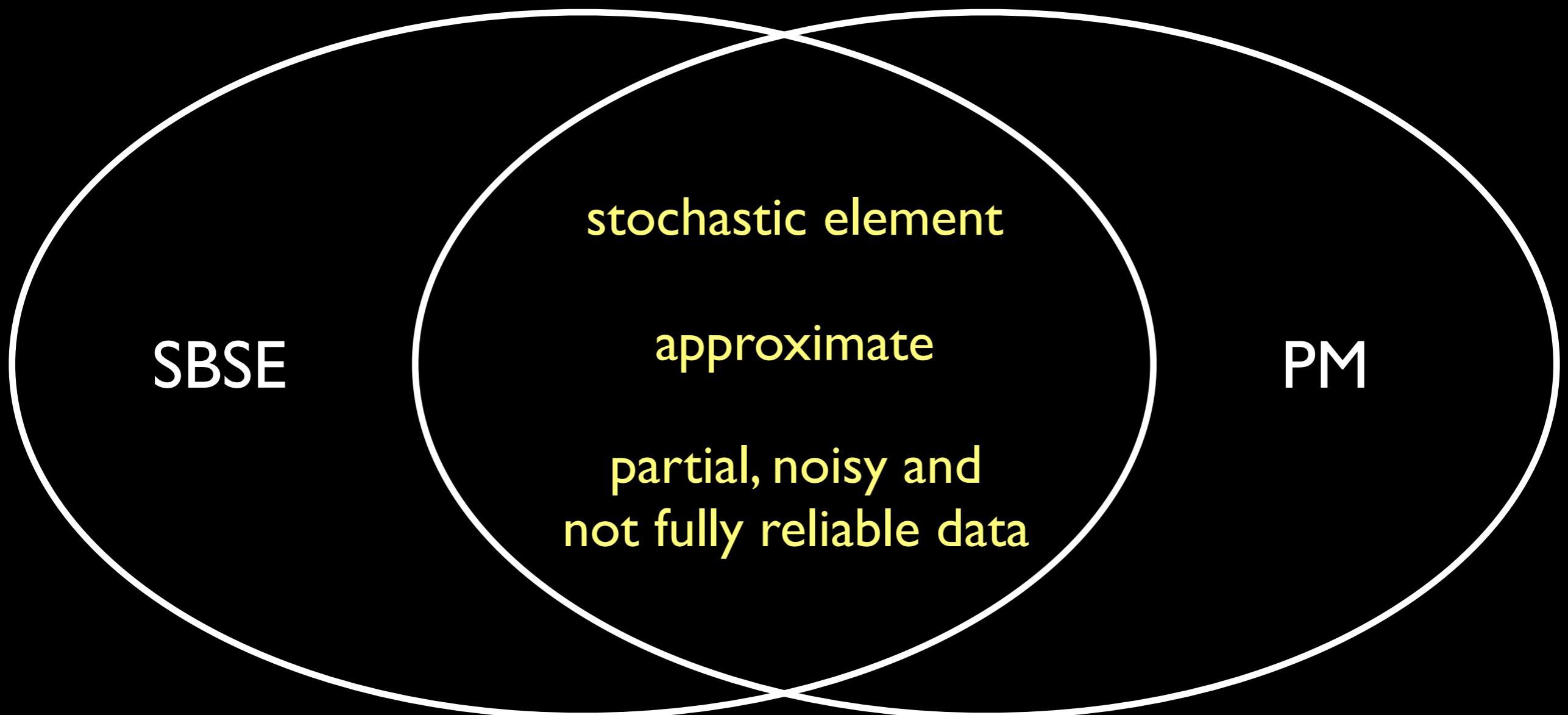
How much do we need to know to build effective models?

How to adapt models to new data?

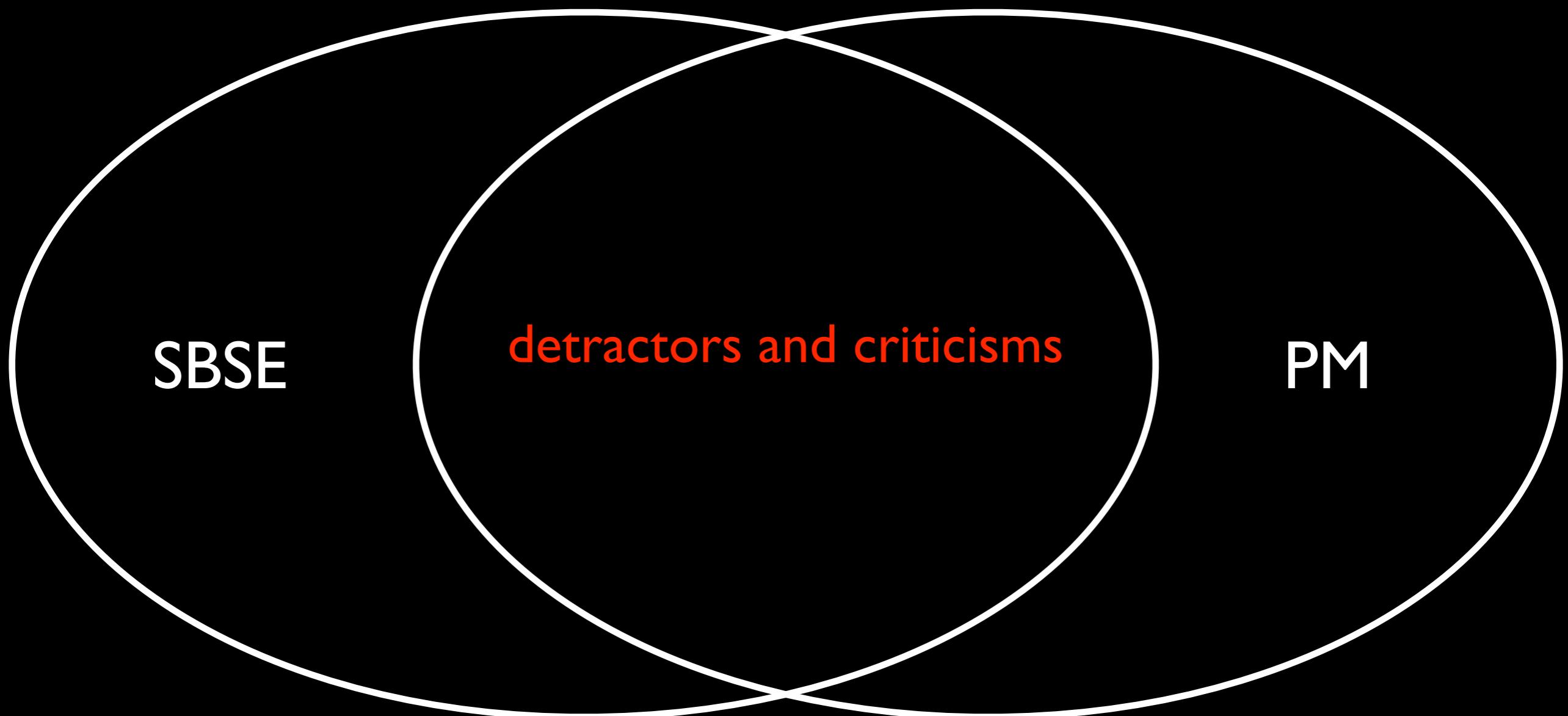
How to streamline the data or the process?

How can predictive modelling gain greater acceptance?

The sad outcasts



The sad outcasts



The advantages of SBSE



Scalable



Insight-rich



Robust



Generic



Realistic

The advantages of SBSE



Scalable



Insight-rich



Robust



Generic



Realistic

The advantages of SBSE



Insight-rich



Scalable



Robust



Generic



Realistic

The advantages of SBSE



Scalable



Insight-rich



Robust



Generic



Realistic

The advantages of SBSE



Scalable



Insight-rich



Robust



Generic



Realistic

The advantages of SBSE



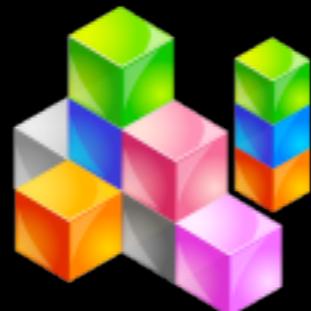
Scalable



Insight-rich



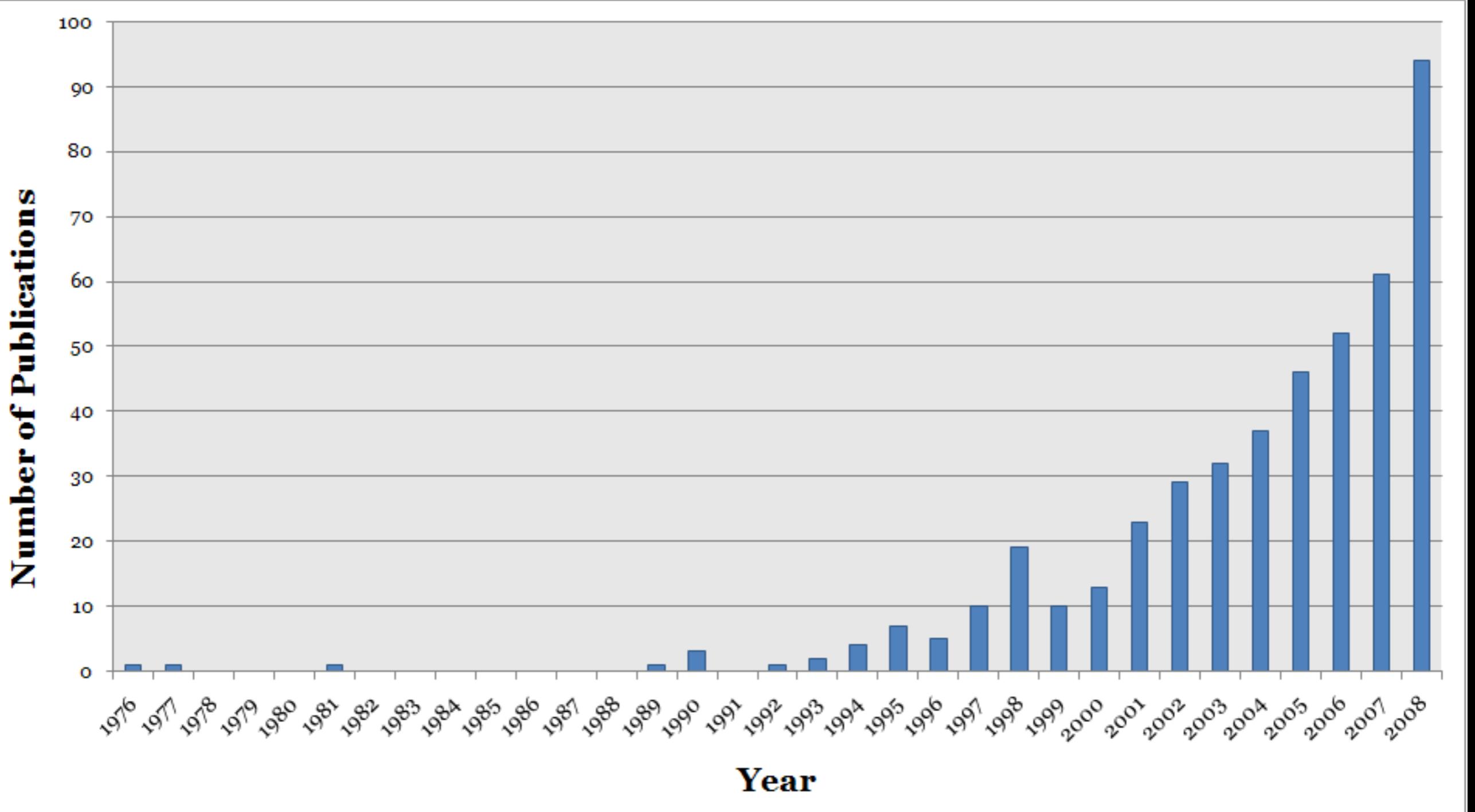
Robust



Generic



Realistic

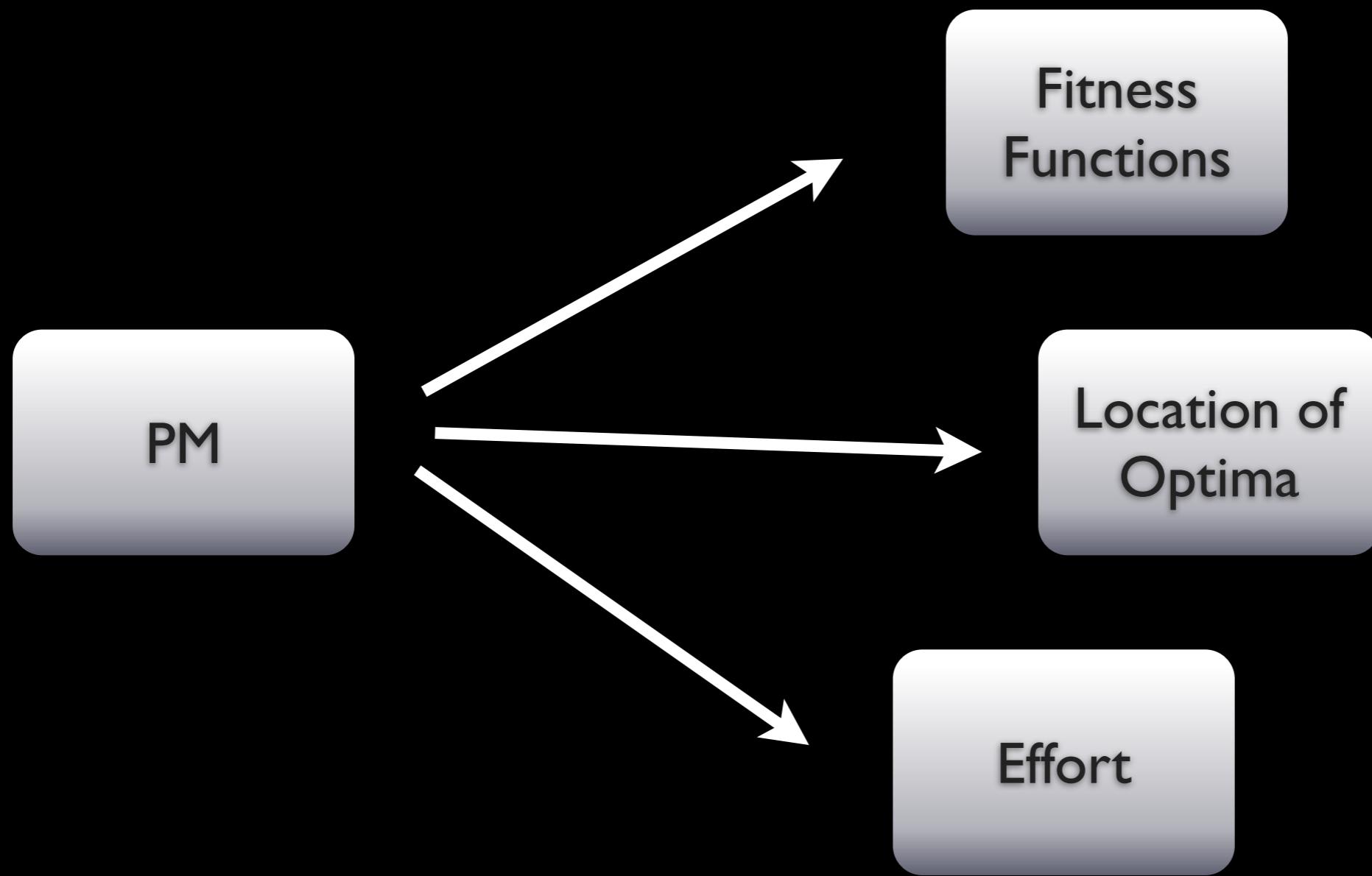


World Wide Activity

24 countries every continent > 600 papers

Next session at PROMISE ... and two others

PM for SBSE



Why Search Based Software Engineering



***La trahison des images* 1929**

René Magritte, 1898–1967

*“The famous pipe. How people reproached me for it!
And yet, could you stuff my pipe?
No, it's just a representation, is it not?
So if I had written on my picture 'This is a pipe,'
I'd have been lying!”*

Quoted in: Harry Torczyner. *Magritte: Ideas and Images*. Alcuin Books.

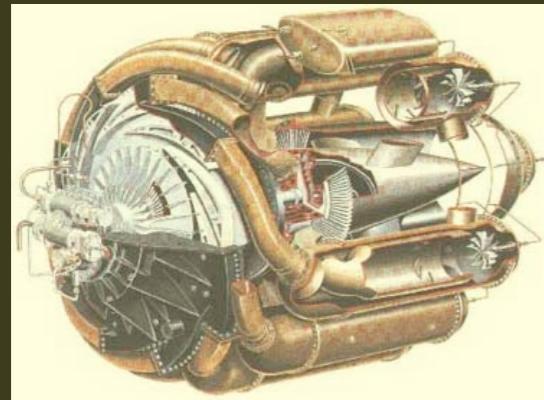
*“The famous pipe. How people reproached me for it!
And yet, could you stuff my pipe?
No, it's just a representation, is it not?
So if I had written on my picture 'This is a pipe,'
I'd have been lying!”*

Quoted in: Harry Torczyner. *Magritte: Ideas and Images*. Alcuin Books.

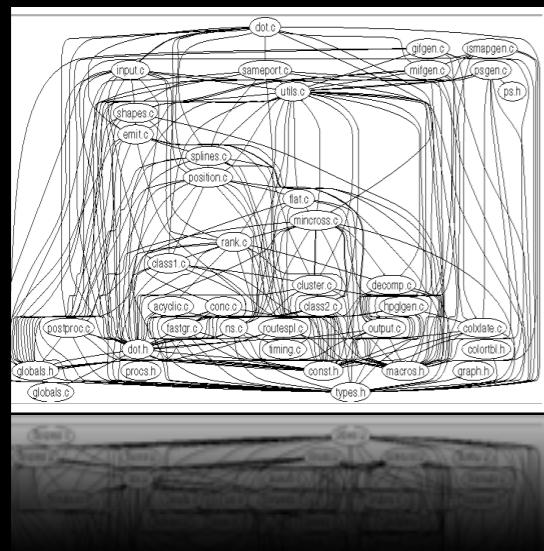
*“The famous pipe. How people reproached me for it!
And yet, could you stuff my pipe?
No, it's just a representation, is it not?
So if I had written on my picture 'This is a pipe,'
I'd have been lying!”*

Quoted in: Harry Torczyner. *Magritte: Ideas and Images*. Alcuin Books.

Traditional Engineering Artifact



Software Engineering Artifact



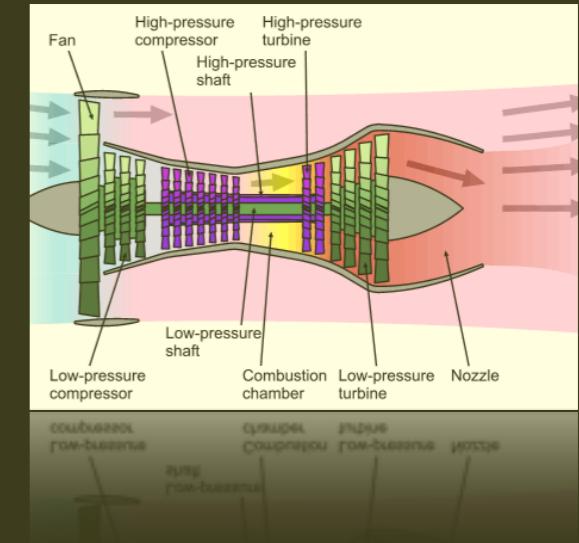
Mark Harman
PROMISE'10 keynote

Optimization goal

Maximize compression

Minimize fuel consumption

Fitness computed on a representation



Optimization goal

Fitness computed Directly

Maximize cohesion
Minimize coupling

www.sebase.org

What do you predict
for
predictive modelling
?



Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: **Search-based software test data generation: a survey.** *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering:* 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: **The Current State and Future of Search Based Software Engineering.** *ICSE Future of Software Engineering:* 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. *Technical Report D-2009-1, University of Tampere, Finland*, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, *Technical Report TR-09-03, King's College London*, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: The Current State and Future of Search Based Software Engineering. *ICSE Future of Software Engineering*: 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010

Overviews and Survey Paper

1. Mark Harman, Bryan Jones: Search-based software engineering. *Information & Software Technology* 43(14): 833-839, 2001
2. Phil McMinn: Search-based software test data generation: a survey. *Software Testing, Verification and Reliability* 14(2): 105-156, 2004
3. Mark Harman: **The Current State and Future of Search Based Software Engineering.** *ICSE Future of Software Engineering:* 342-357, 2007
4. Outi Räihä: A Survey on Search-Based Software Design. Technical Report D-2009-1, University of Tampere, Finland, 2009
5. Wasif Afzal, Richard Torkar and Robert Feldt: A Systematic Review of Search-based Testing for Non-Functional System Properties. *Information and Software Technology*, 51(6):957-976, 2009
6. Mark Harman, Afshin Mansouri and Yuanyuan Zhang, Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Technical Report TR-09-03, King's College London, 2009
7. Shaukat Ali, Lionel Briand, Hadi Hemmati and Rajwinder Panesar-Walawege: A Systematic Review of the Application and Empirical Investigation of Search-Based Test-Case Generation. *IEEE Transactions on Software Engineering*, To appear, 2010