The Exploratory Modeling Workbench

An open source toolkit for exploratory modeling, scenario discovery, and (many objective) robust decision making

Jan H. Kwakkel

TU Delft

Online resources

code: https://github.com/quaquel/EMAworkbench

documentation: http://emaworkbench.rtfd.io/en/latest/?badge=latest

demo: https://github.com/quaquel/lake_problem

Many of my recent papers are open access and have online resources, so check the repositories of my github account (https://github.com/quaquel)

Also check out https://waterprogramming.wordpress.com

TU Delft

OR FORUM

EXPLORATORY MODELING FOR POLICY ANALYSIS

STEVE BANKES
RAND, Santa Monica, California

(Received February 1992; revision received January 1993; accepted March 1993)

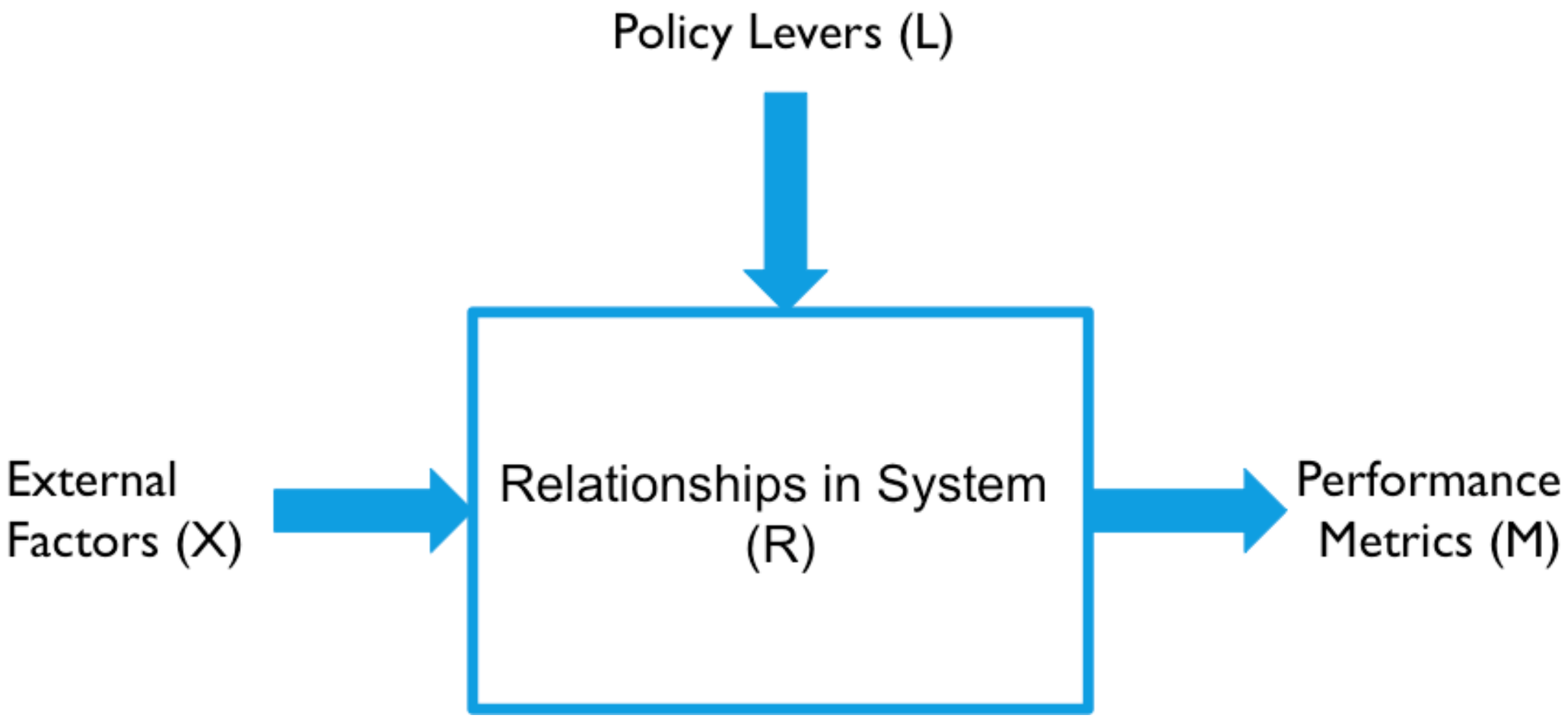
Exploratory modeling is using computational experiments to assist in reasonir
uncertainty. While frequently confused with the use of models to consolidate
predict system behavior, exploratory modeling is a very different kind of us
model development. This paper distinguishes these two broad classes of moc
used in exploratory modeling, and suggests some technological innovations ne

Building a model by consolidating known facts into a single package and then using it as a surrogate for the actual system, which I call *consolidative modeling*, is in many ways the standard approach to model development and use. Where successful, it is a powerful technique for understanding the behavior of complex systems. Unfortunately, the consolidative approach is not always possible.

When insufficient knowledge or unresolvable uncertainties preclude building a surrogate for the target system, modelers must make guesses at details and mechanisms. While the resulting model cannot be taken as a reliable image of the target system, it does provide a computational experiment that reveals how the world would behave if the various guesses were correct. *Exploratory modeling* is the use of series of such computational experiments to explore the implications of varying assumptions and hypotheses.

Enormous increases in the availability of computational power in the past few years have made aggressive exploratory use of complex computer models possible for the first time. We now live in an era in which computational experiments are commonplace in many of the sciences (Strauss 1974, Campbell et al. 1985, Rose and Dobson 1985, Anderson 1988, Lipton, Marr and Welsh 1989). Exploratory use

XLRM framework



TU Delft

Running a model as a function

$$!\,(!\,,!\,) = !$$

X : uncertainties

- Model structure uncertainties as well as exogenous forces
- Continuous or categorical

L : policies

- Policies to be tested over the uncertainties

M : outcomes

- Outcomes of interest, scalar or time series

TU Delft

Generation of Policies

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Iterative
stress test and refine

Generation of Scenarios

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Relationships within the system

Robustness Measures

Regret
comparing alternatives

Satisficing
performance threshold

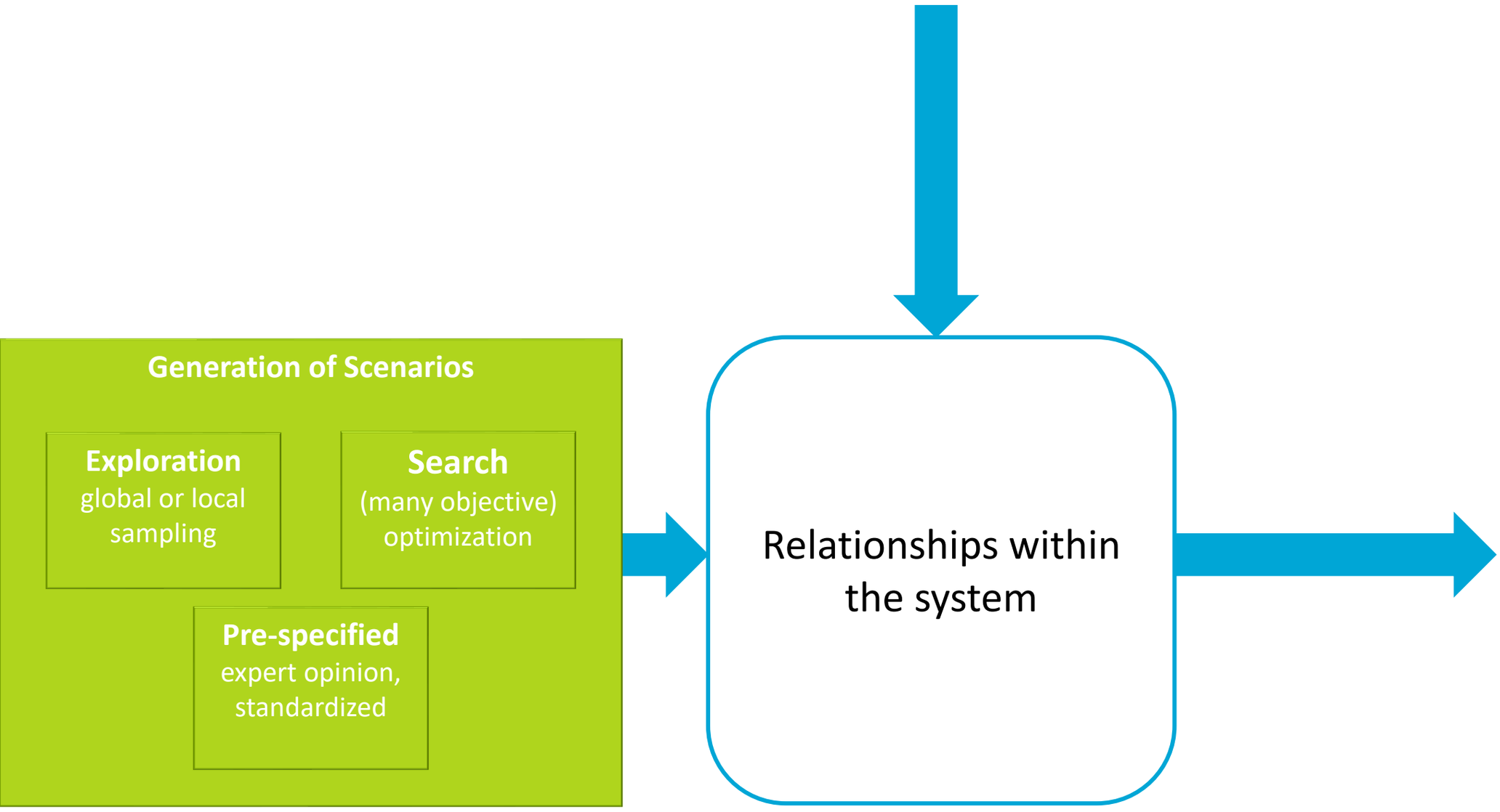
Outcomes
Descriptive statics

Vulnerability Analysis

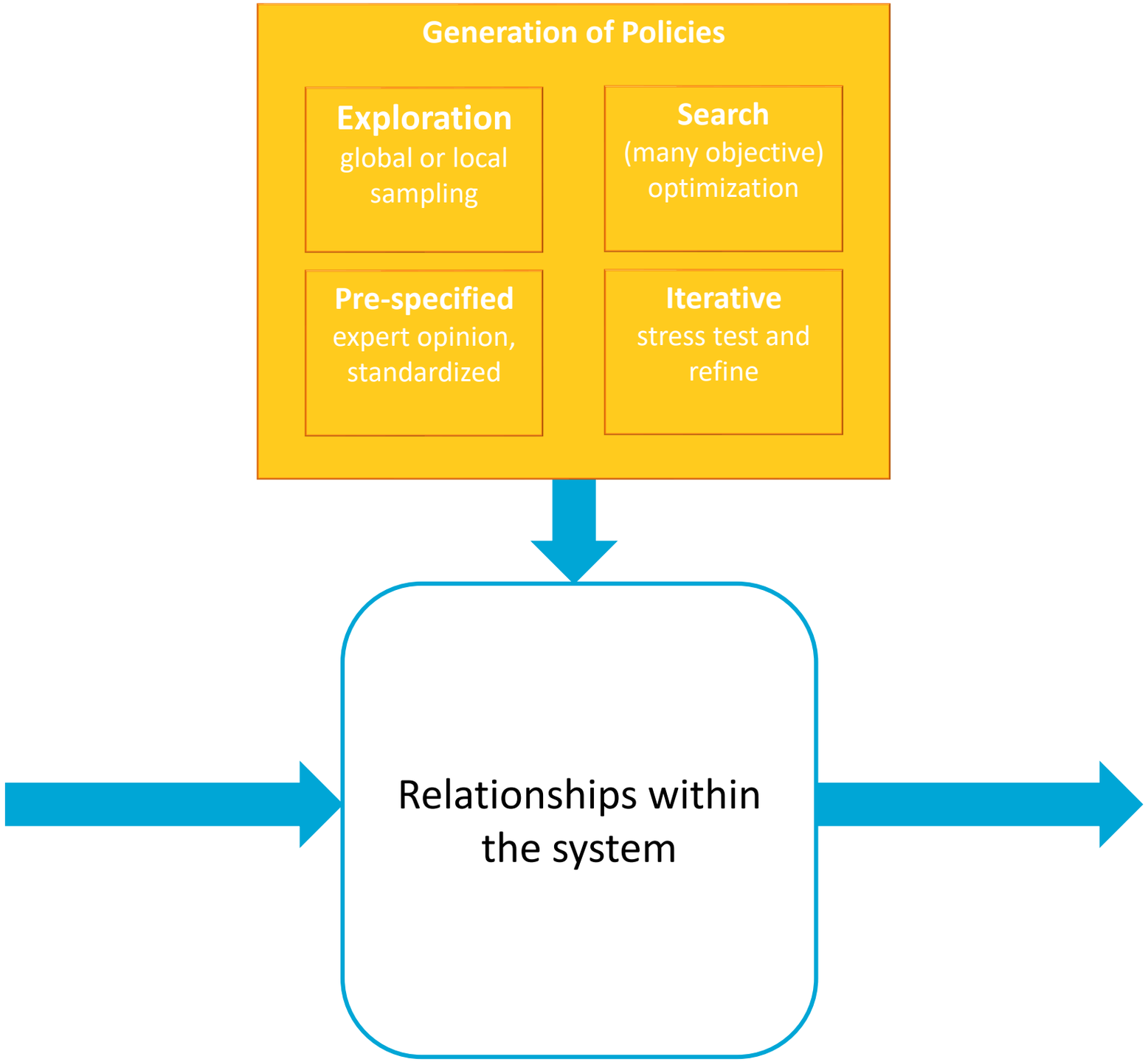
Subspace partitioning
scenario discovery and adaptation tipping points

Sensitivity analysis
Factor prioritization and factor fixing

TU Delft


Generation of Scenarios
Exploration
global or local sampling
Search
(many objective) optimization
Pre-specified
expert opinion, standardized
Relationships within the system

TU Delft


Generation of Policies
Exploration
global or local sampling
Search
(many objective) optimization
Pre-specified
expert opinion, standardized
Iterative
stress test and refine
Relationships within the system

TU Delft

Generation of Policies

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Iterative
stress test and refine

Generation of Scenarios

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Relationships within the system

Robustness Measures

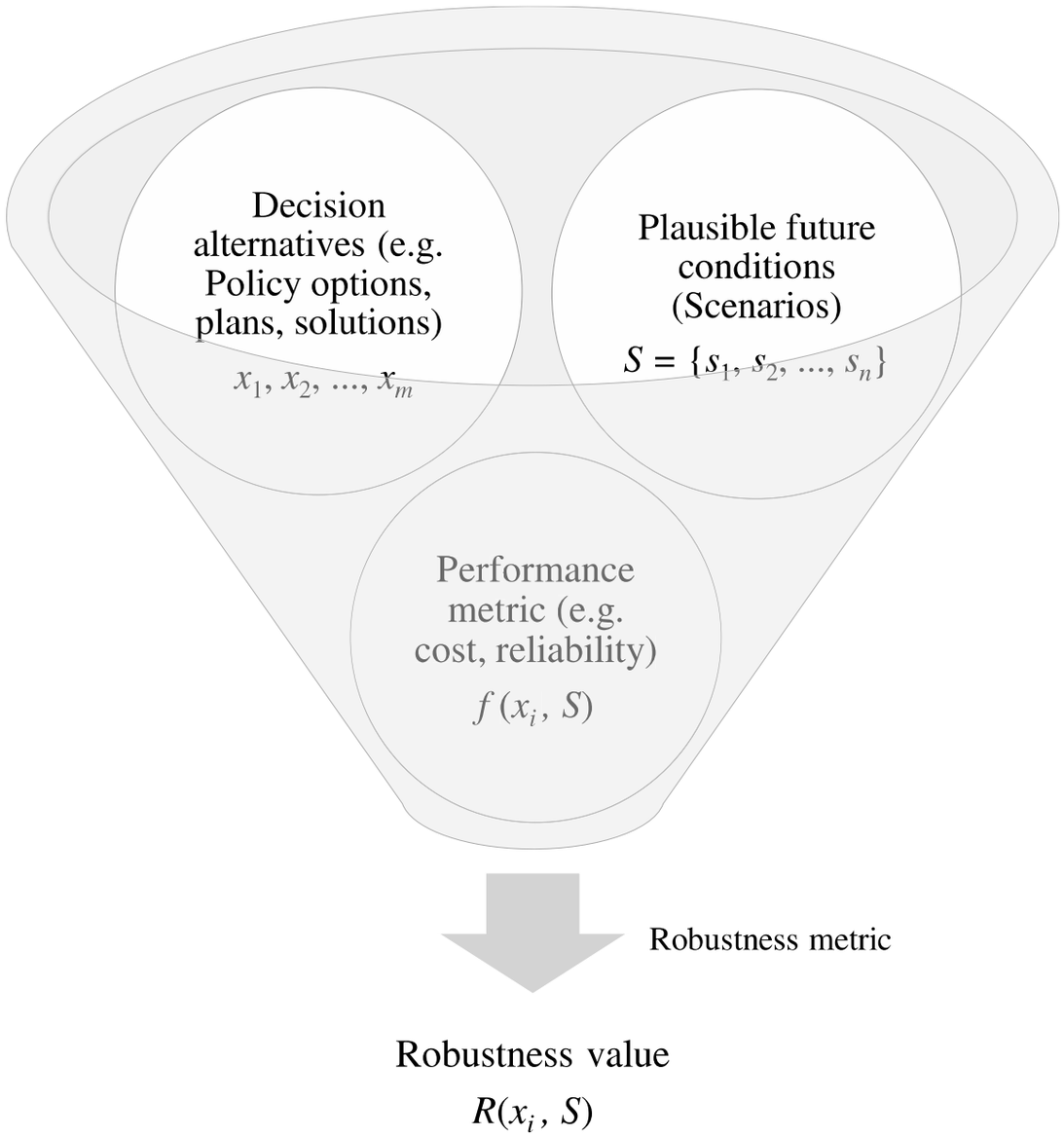
Regret
comparing alternatives

Satisficing
performance threshold

Outcomes
Descriptive statics

TU Delft

Robustness



Regret

Comparison amongst policy options

Satisficing

Performance threshold on outcomes

Descriptive statistics

Moments of the distribution of outcomes over the scenarios

McPhail et al (2018) 10.1002/2017EF000649

Generation of Policies

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Iterative
stress test and refine

Generation of Scenarios

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Relationships within the system

Robustness Measures

Regret
comparing alternatives

Satisficing
performance threshold

Outcomes
Descriptive statics

Vulnerability Analysis

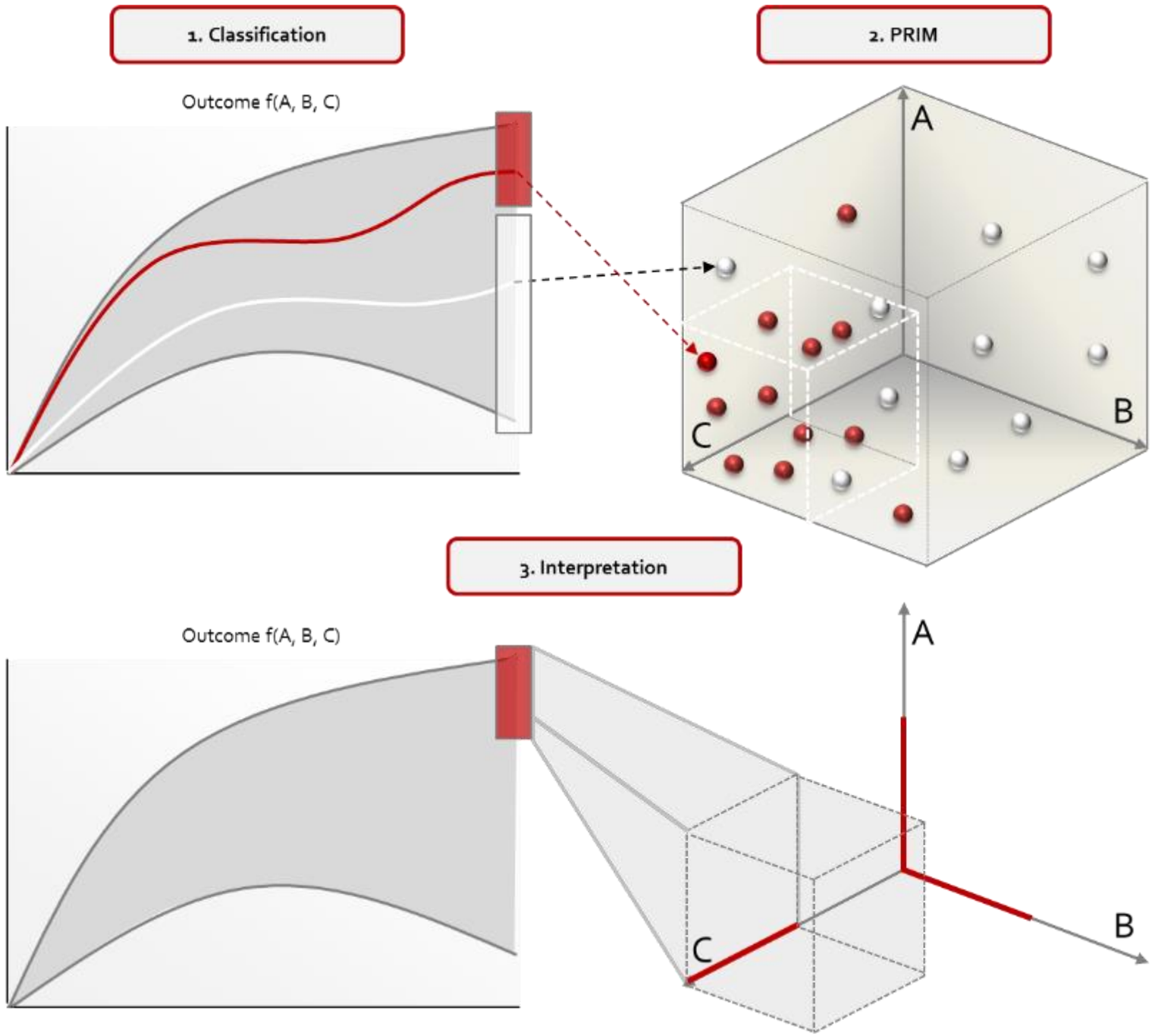
Subspace partitioning
scenario discovery and adaptation tipping points

Sensitivity analysis
Factor prioritization and factor fixing

TU Delft

Subspace Partitioning

Scenario Discovery



Greeven et al (2016) 10.18564/jasss.3134

ROBUSTNESS FRAMEWORKS

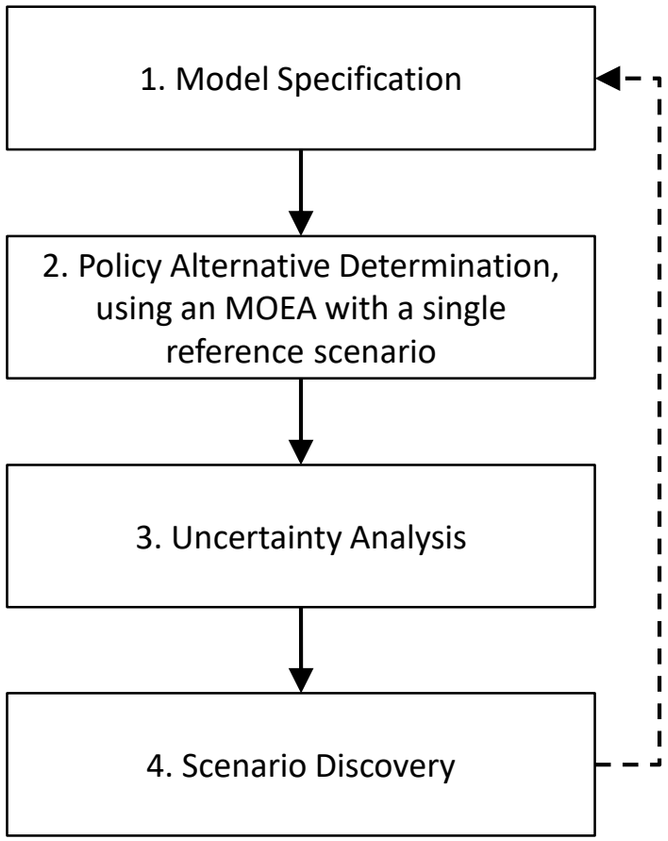
TU Delft

Multi-Objective Robust Decision Making (MORDM)

Introduced by Kasprzyk (2013) as an extension of RDM

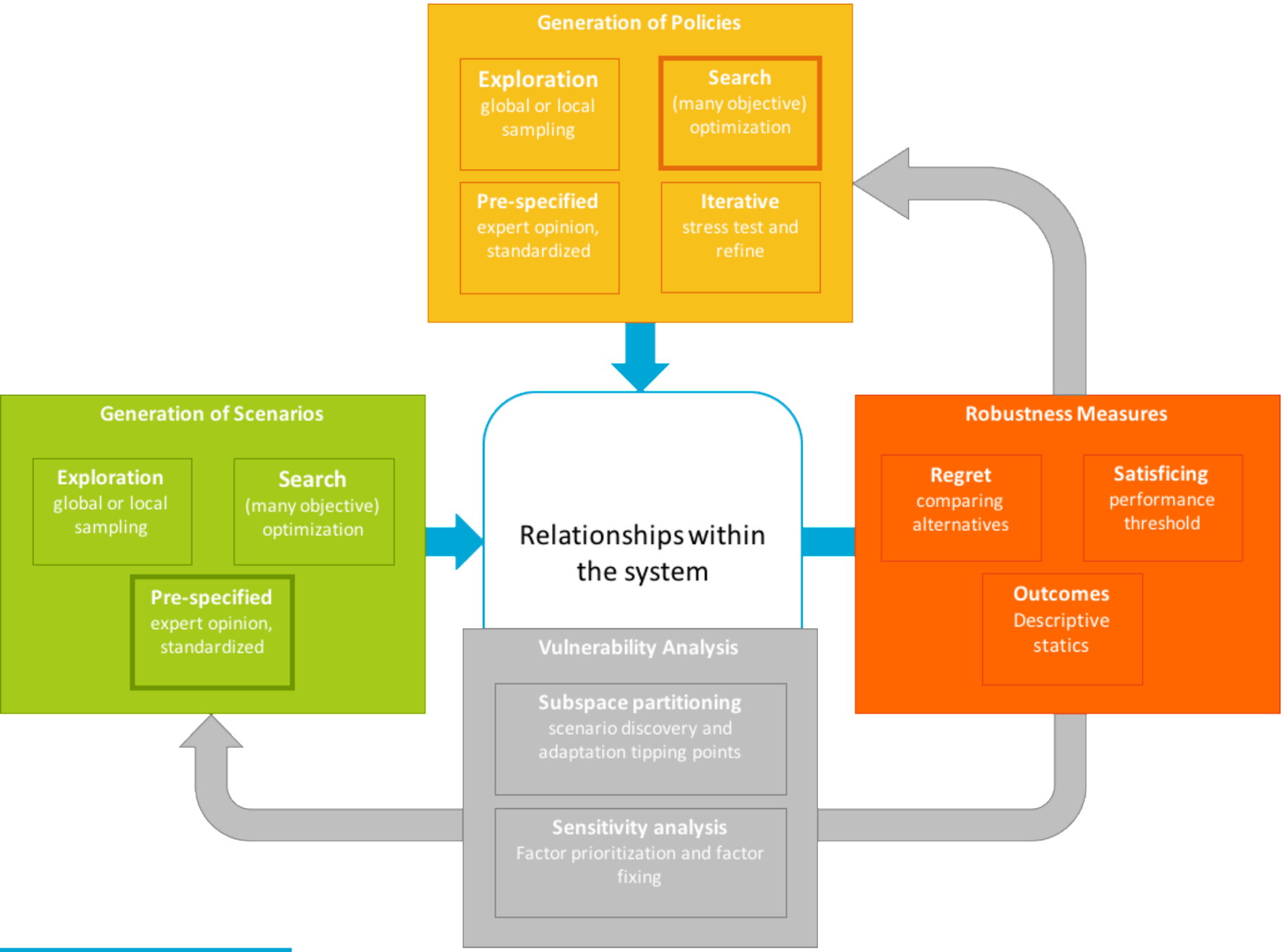
Uses an MOEA to determine potentially robust policy alternatives

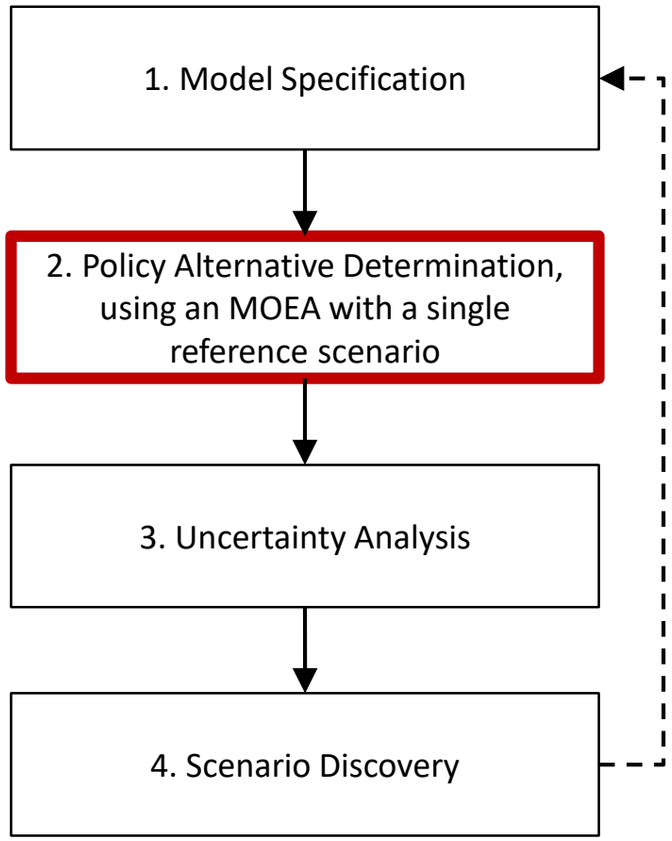
Policy selection is based on performance in a single reference scenario



TU Delft

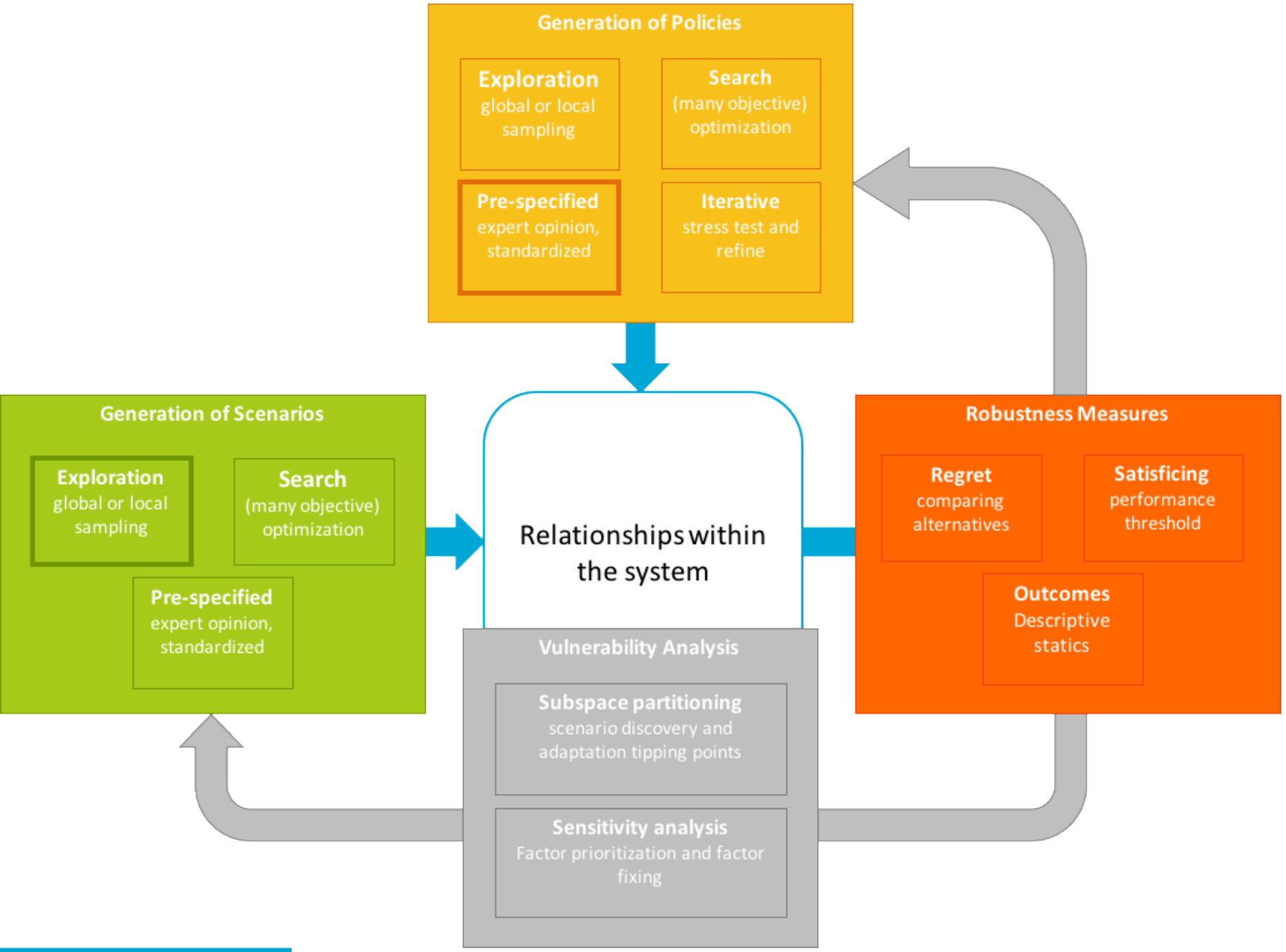
Multi-Objective Robust Decision Making (MORDM)

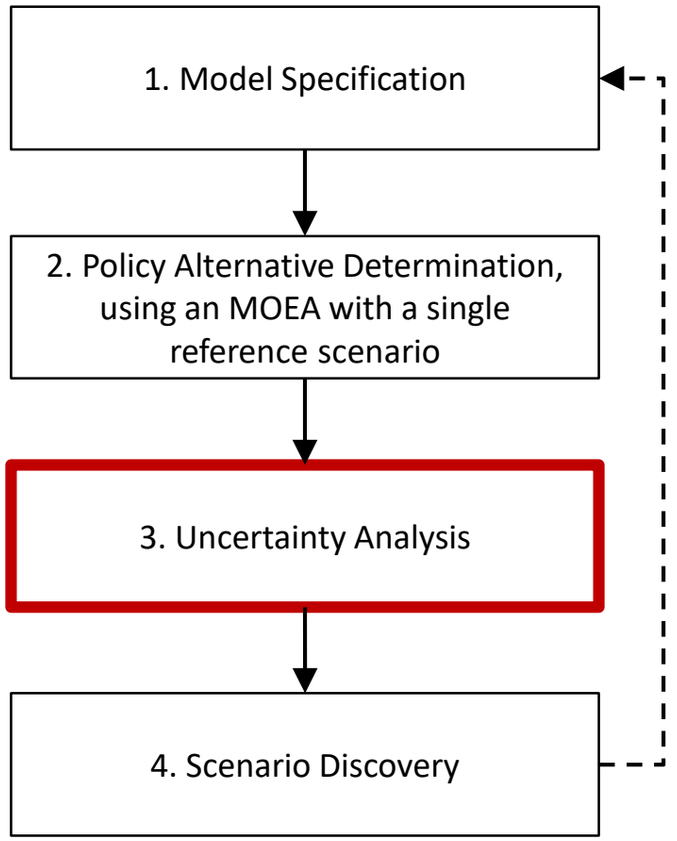




TU Delft

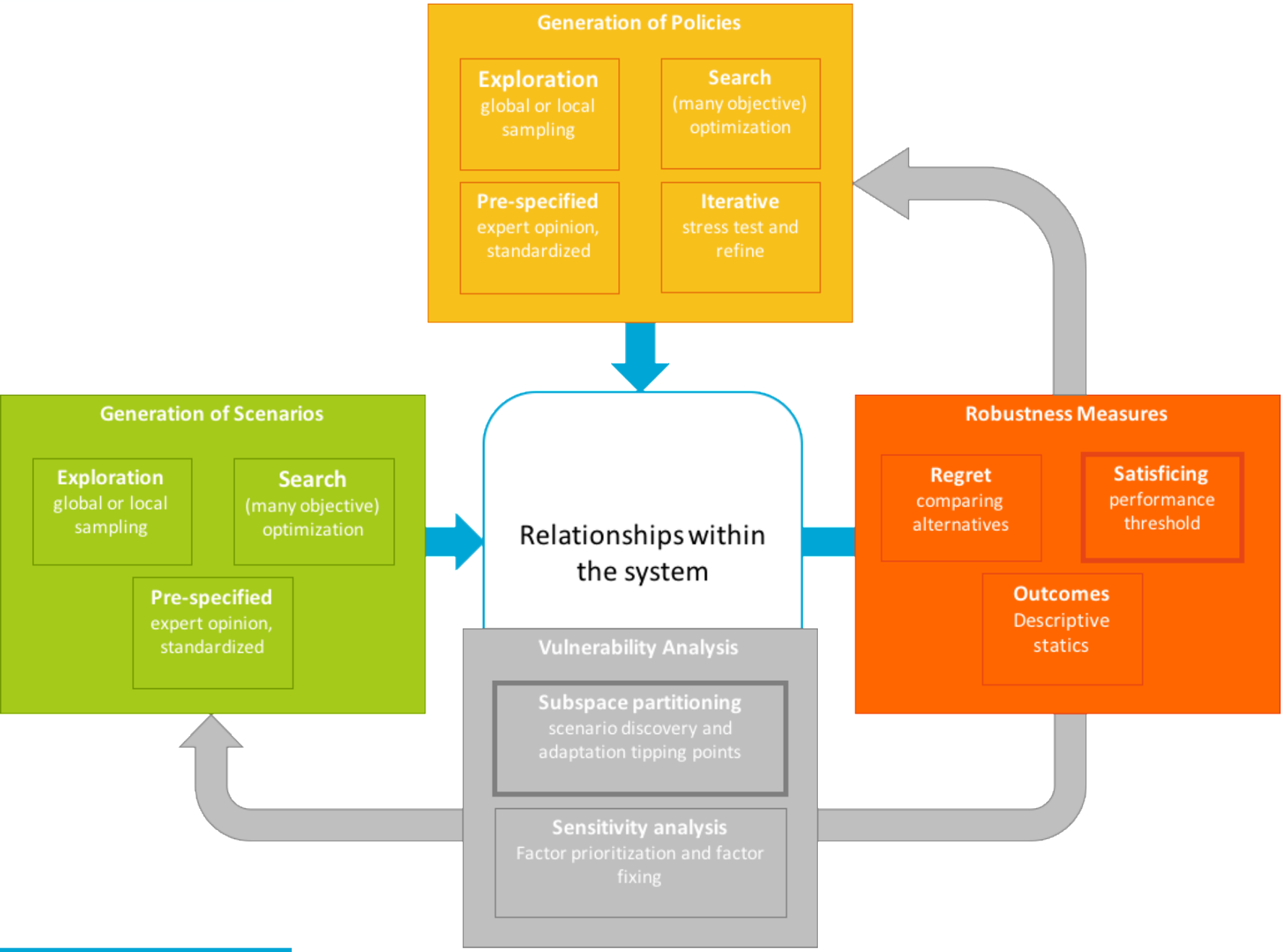
Multi-Objective Robust Decision Making (MORDM)

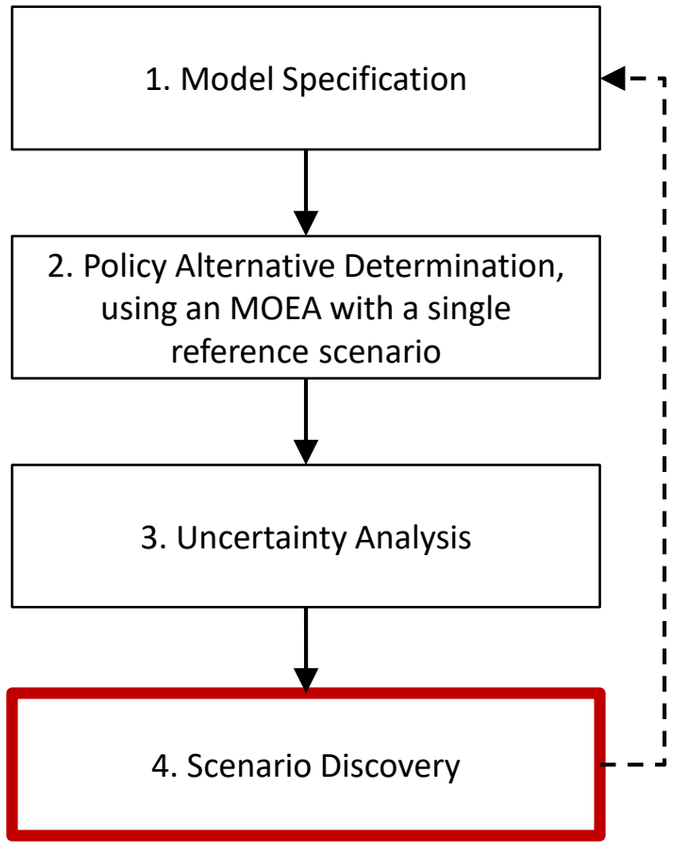




TU Delft

Multi-Objective Robust Decision Making (MORDM)





TU Delft

Multi-scenario MORDM

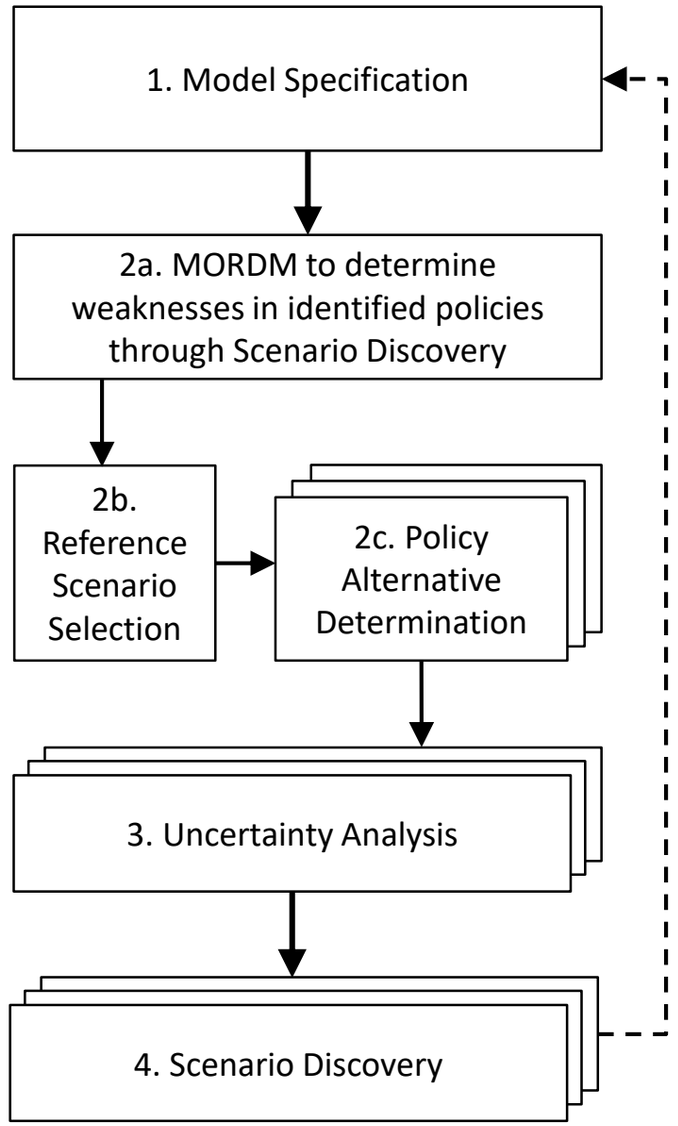
Extension of MORDM
(e.g. Watson & Kasprzyk, 2017)

Updates the search phase to consider multiple reference scenarios

- Goal is to discover a more diverse set of potentially robust policy alternatives

Reference scenario selection based on vulnerabilities found through MORDM

- 4 scenarios that are maximally diverse (e.g. Eker & Kwakkel, 2018)
- Selected from sample of uncertainties in the vulnerable region



TU Delft

Multi-scenario MORDM

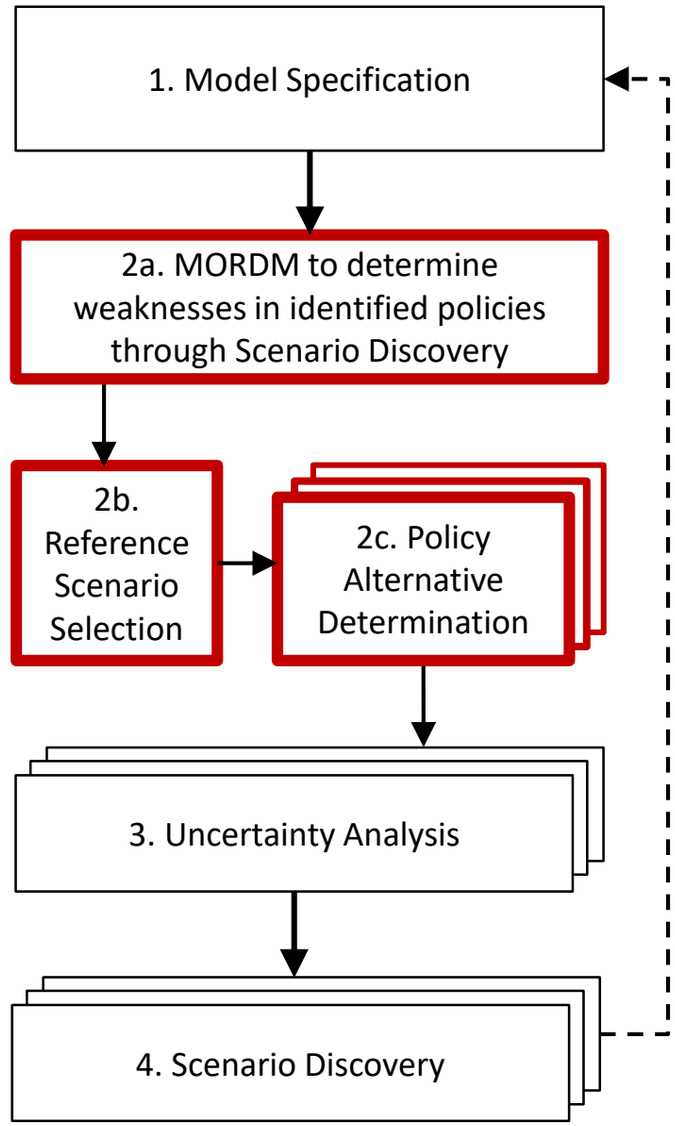
Extension of MORDM
(e.g. Watson & Kasprzyk, 2017)

Updates the search phase to consider multiple reference scenarios

- Goal is to discover a more diverse set of potentially robust policy alternatives

Reference scenario selection based on vulnerabilities found through MORDM

- 4 scenarios that are maximally diverse (e.g. Eker & Kwakkel, 2018)
- Selected from sample of uncertainties in the vulnerable region

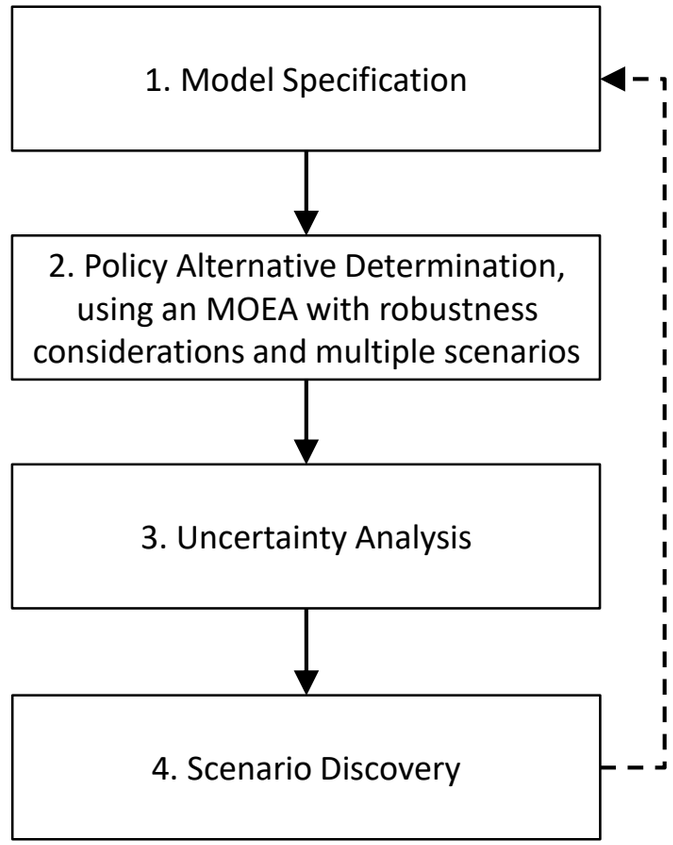


TU Delft

Multi-Objective Robust Optimization (MORO)

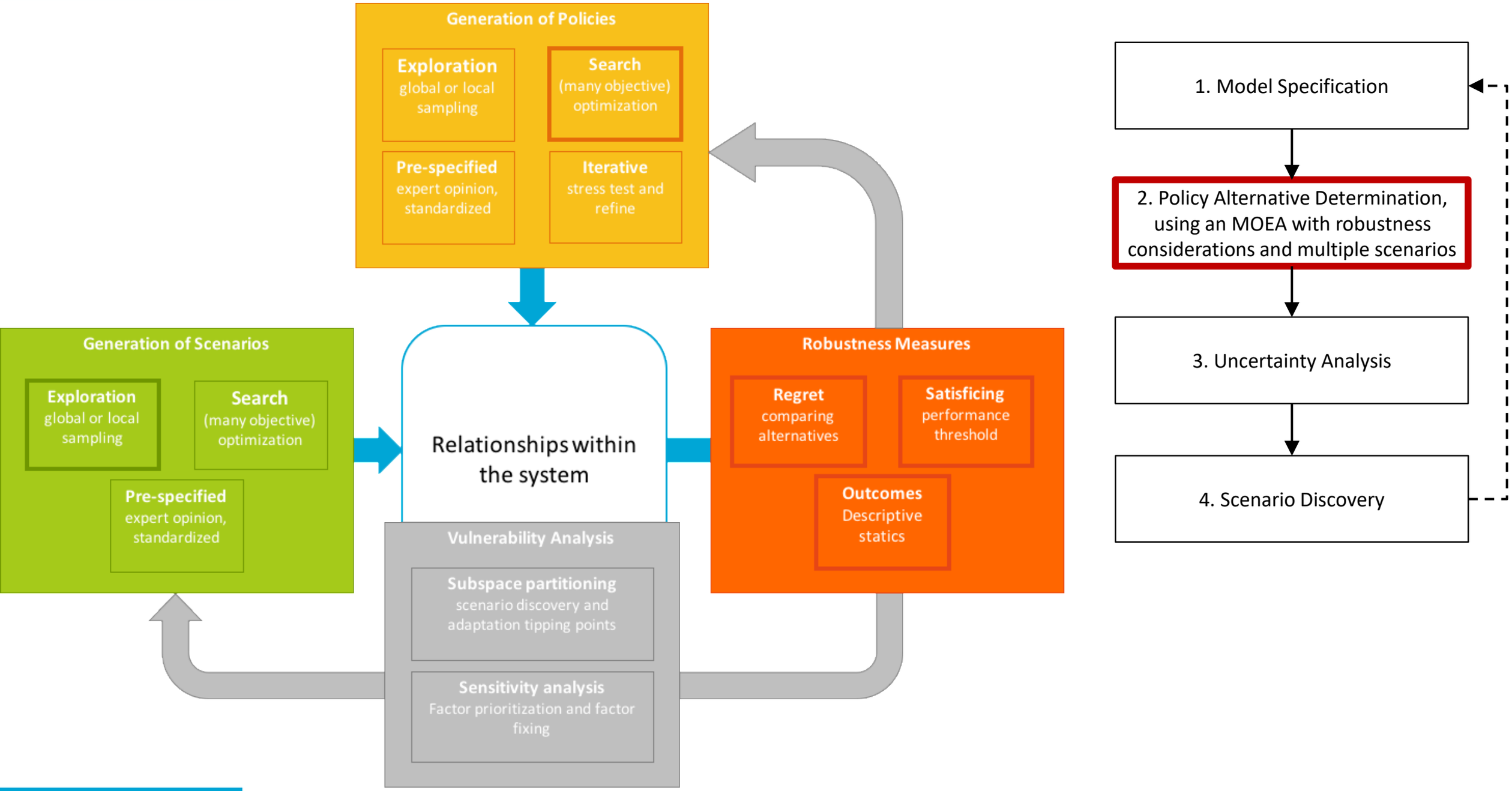
The search phase now selects policy alternatives based on robust outcomes

- Robustness determined by testing each candidate policy on a static set of scenarios

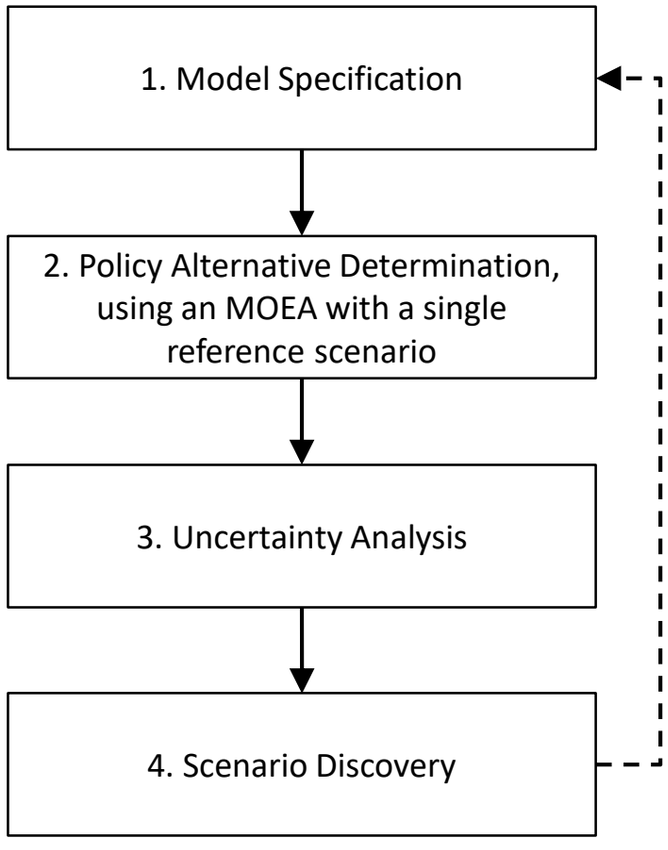


TU Delft

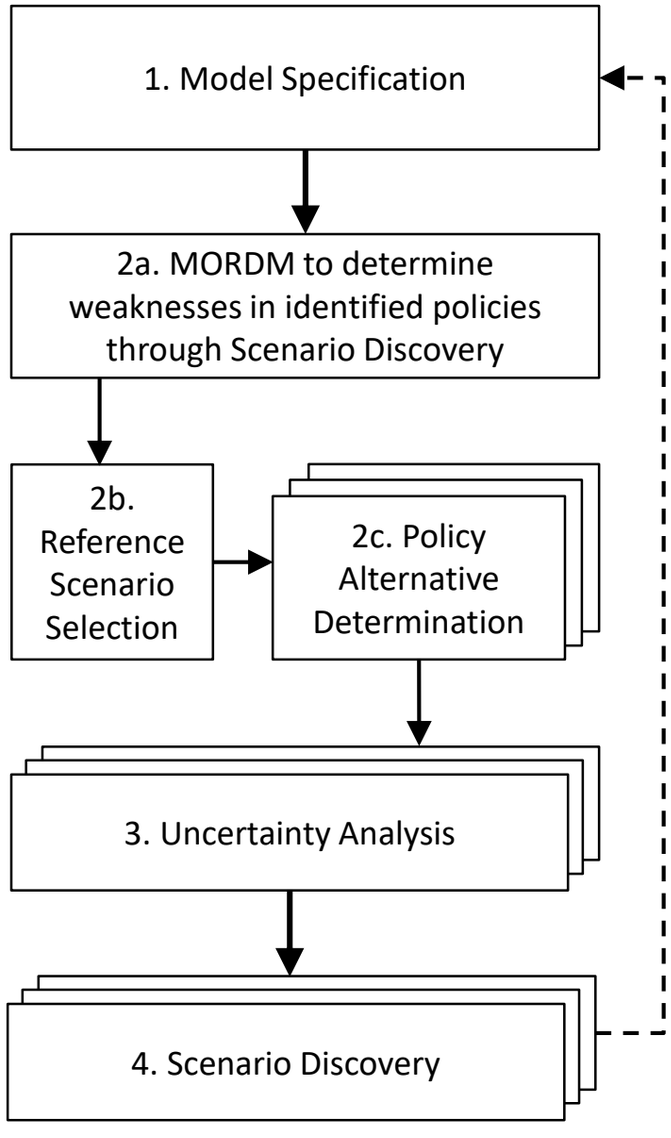
Multi-Objective Robust Optimization (MORO)



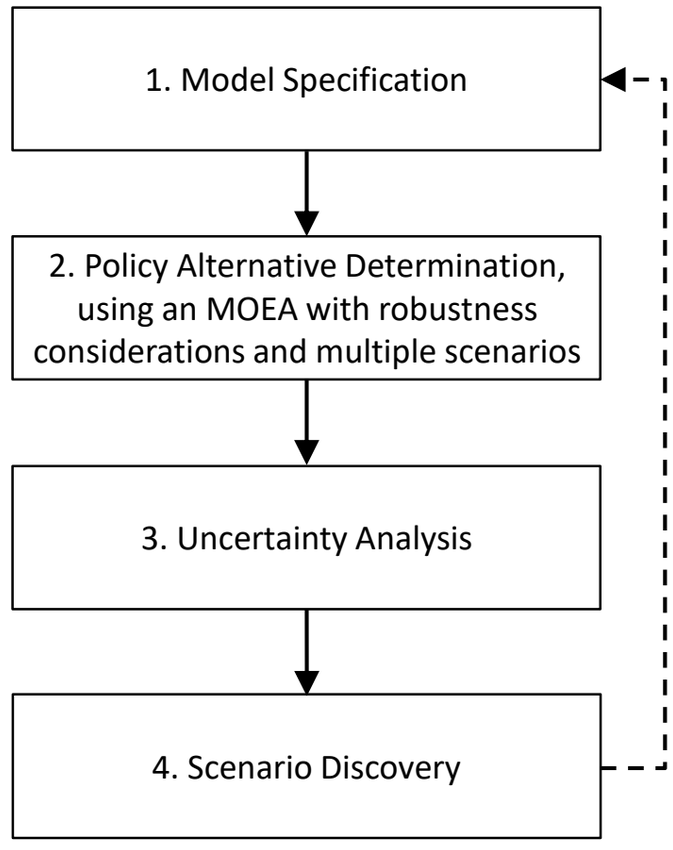
MORDM



Multi-scenario MORDM



MORO



TU Delft

THE EMA WORKBENCH

TU Delft

The workbench

Implemented in Python

- Support Python 2 and Python 3, new developments are Python 3 only.
- Works with both 32 and 64 bit Python
- Designed for use with Jupyter notebooks

Packages

- **em_framework**: the core classes and functions for defining an interface to a model, setup the experiments, and execute them. Support is available for parallelization on single machine as well as for clusters
- **analysis**: visual analytics and vulnerability analysis techniques
- **util**: saving, loading, logging
- **connectors**: ready made connectors for Vensim, Netlogo, and Excel

TU Delft

Online resources

code: https://github.com/quaquel/EMAworkbench

documentation: http://emaworkbench.rtfd.io/en/latest/?badge=latest

demo: https://github.com/quaquel/lake_problem

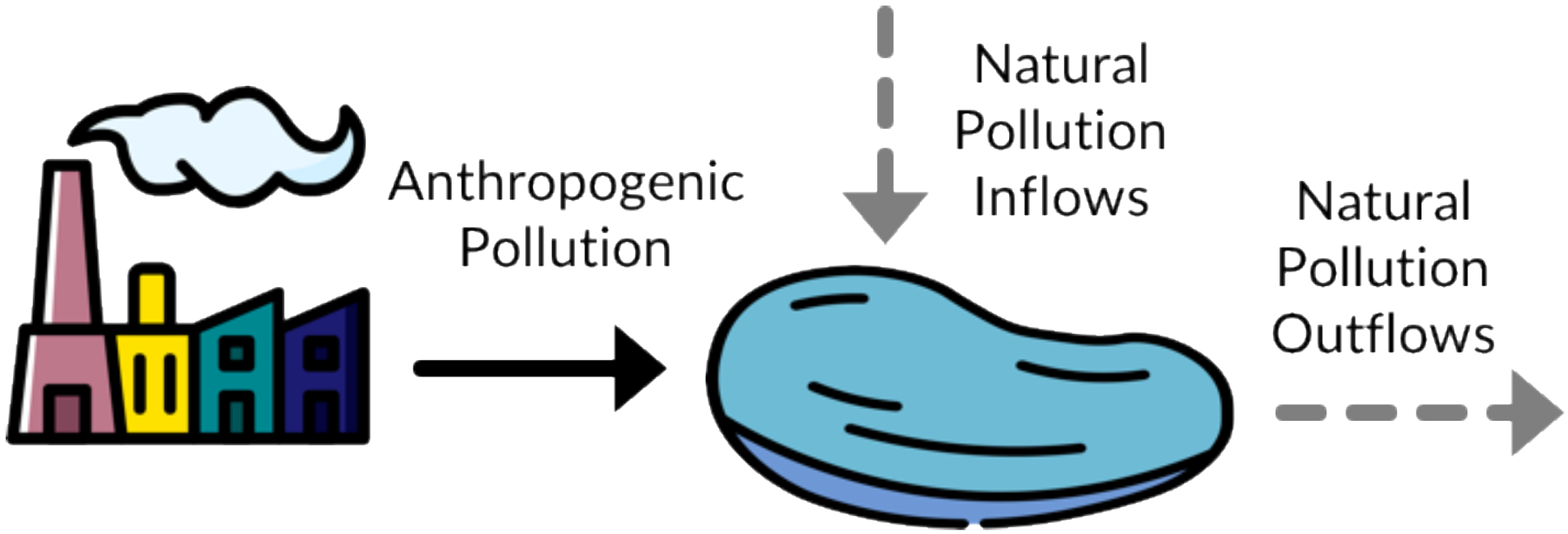
Many of my recent papers are open access and have online resources, so check the repositories of my github account (https://github.com/quaquel)

Also check out https://waterprogramming.wordpress.com

TU Delft

THE LAKE PROBLEM

TU Delft



Decision problem: what is a robust rule for determining the annual anthropogenic pollution?

- Maximize utility
- Maximize reliability
- Maximize inertia
- Minimize maximum pollution

TU Delft

Code of the lake model

https://emaworkbench.readthedocs.io

→ general introduction

Or

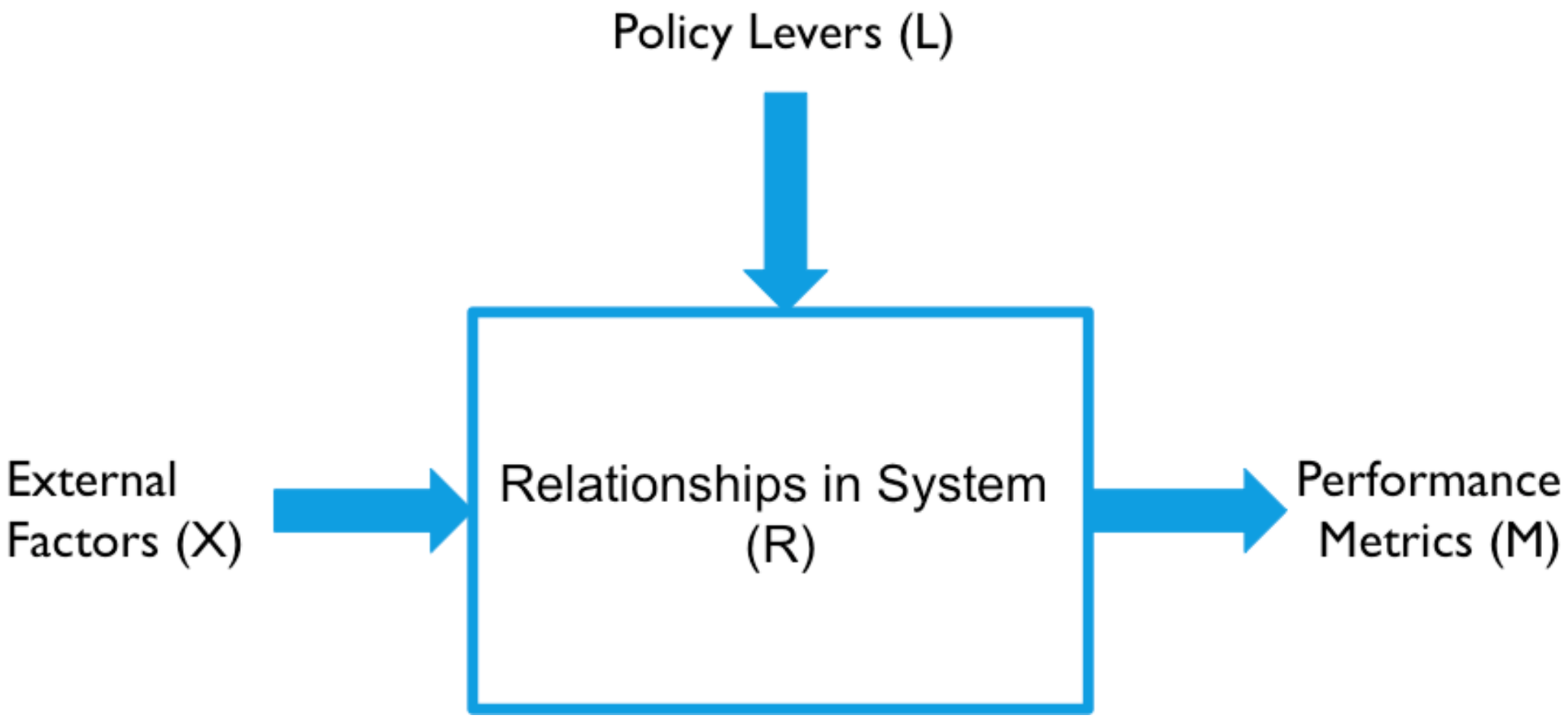
./site_packages/ema_workbench/examples/lake_model_dps.py

→ search for lake_model_dps.py in your file browser

RECAP

TU Delft

XLRM framework



TU Delft

Running a model as a function

$$!\,(!\,,!\,) = !$$

X : uncertainties

- Model structure uncertainties as well as exogenous forces
- Continuous or categorical

L : policies

- Policies to be tested over the uncertainties

M : outcomes

- Outcomes of interest, scalar or time series

TU Delft

Generation of Policies

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Iterative
stress test and refine

Generation of Scenarios

Exploration
global or local sampling

Search
(many objective) optimization

Pre-specified
expert opinion, standardized

Relationships within the system

Robustness Measures

Regret
comparing alternatives

Satisficing
performance threshold

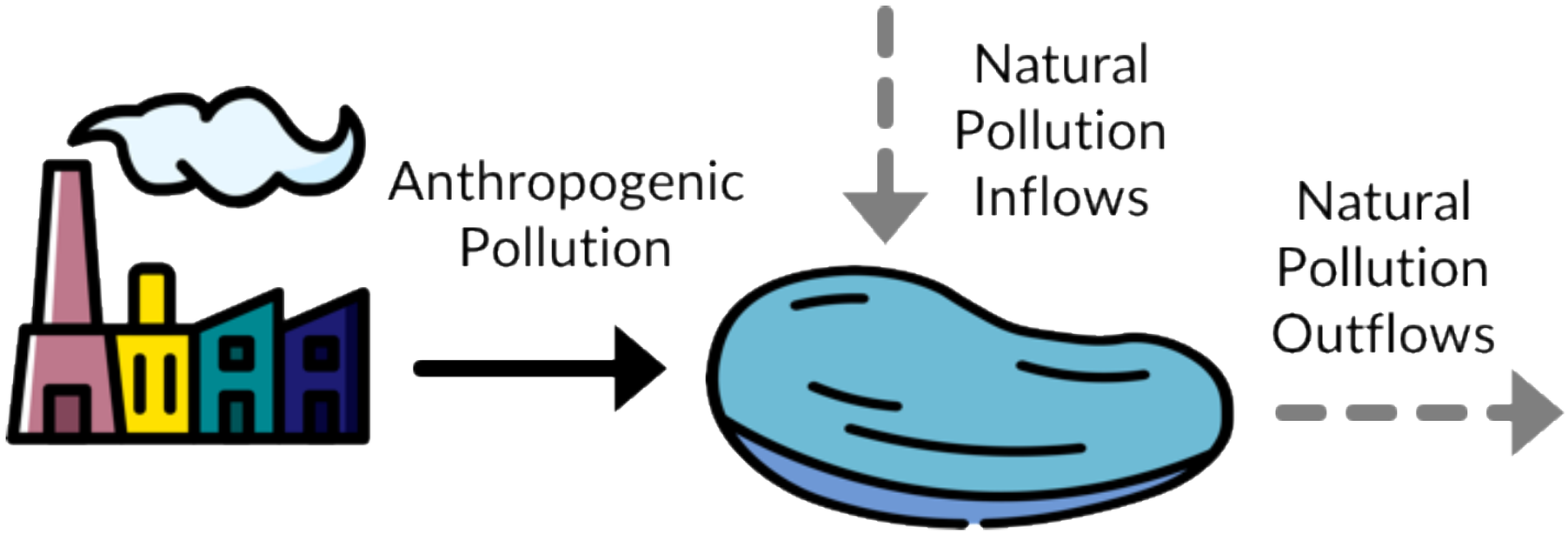
Outcomes
Descriptive statics

Vulnerability Analysis

Subspace partitioning
scenario discovery and adaptation tipping points

Sensitivity analysis
Factor prioritization and factor fixing

TU Delft



Decision problem: what is a robust rule for determining the annual anthropogenic pollution?

- Maximize utility
- Maximize reliability
- Maximize inertia
- Minimize maximum pollution

TU Delft

Code of the lake model

https://emaworkbench.readthedocs.io

→ general introduction

Or

./site_packages/ema_workbench/examples/lake_model_dps.py

→ search for lake_model_dps.py in your file browser

TU Delft