Project Platypus

A Collection of Libraries for Optimization, Data Analysis, and Decision Making
Developed by David Hadka

https://github.com/Project-Platypus

Julianne Quinn
jdq8@cornell.edu

What is Project Platypus?

Python extension of the R package:

OpenMORDM

Multiobjective Robust Decision Making in R

http://github.com/OpenMORDM

Rhodium
Python tool for Robust Decision Making, built off of the EMA workbench

PRIM
Stand-alone Python implementation of the Patient Rule Induction Method

Platypus
Python library of Multi-Objective Evolutionary Algorithms (MOEAs)

J3

J3
Java platform for visualizing and analyzing multi-objective trade-offs

https://github.com/Project-Platypus

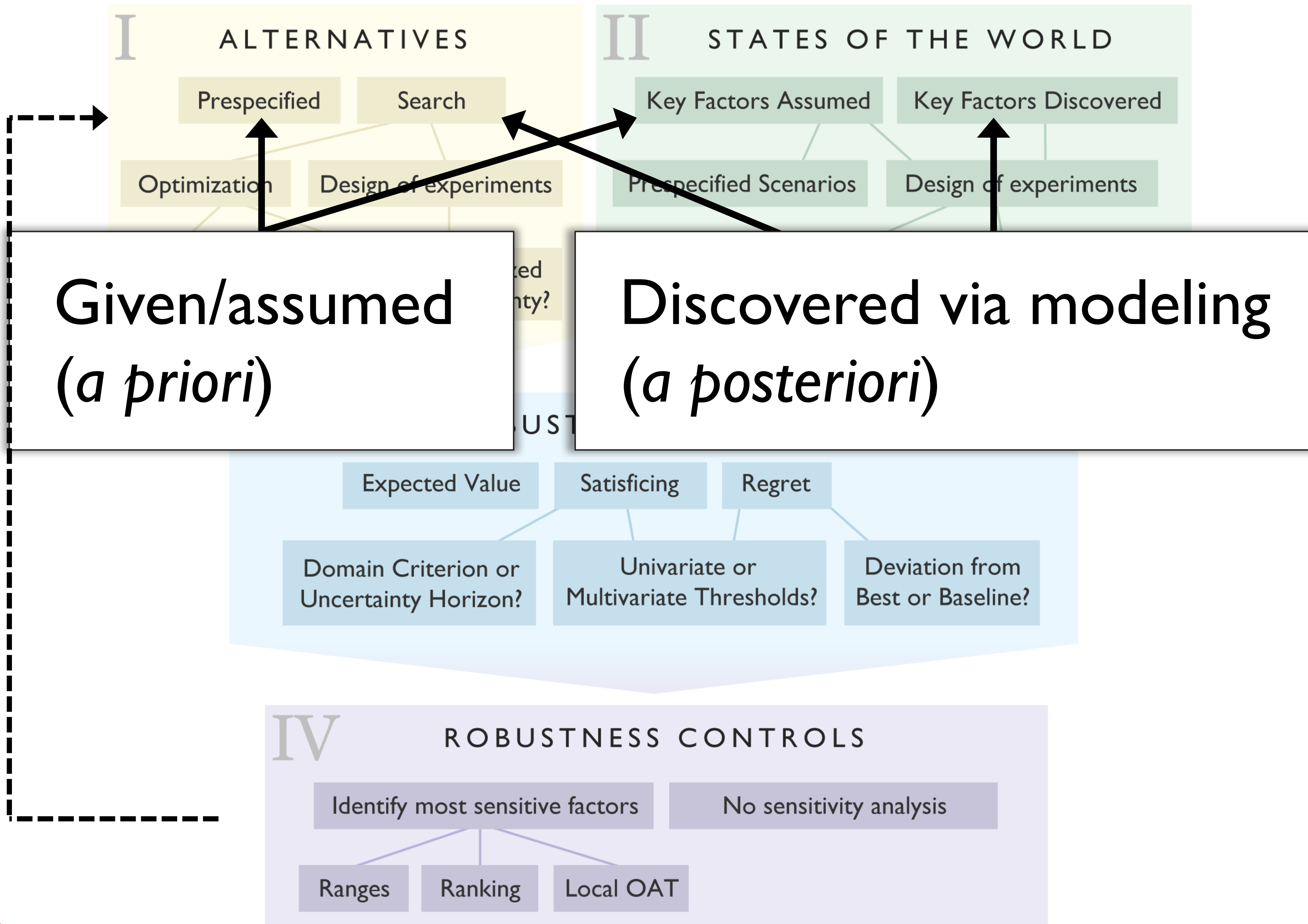
What do robustness analyses have in common?

Evaluate alternatives in multiple states of the world…

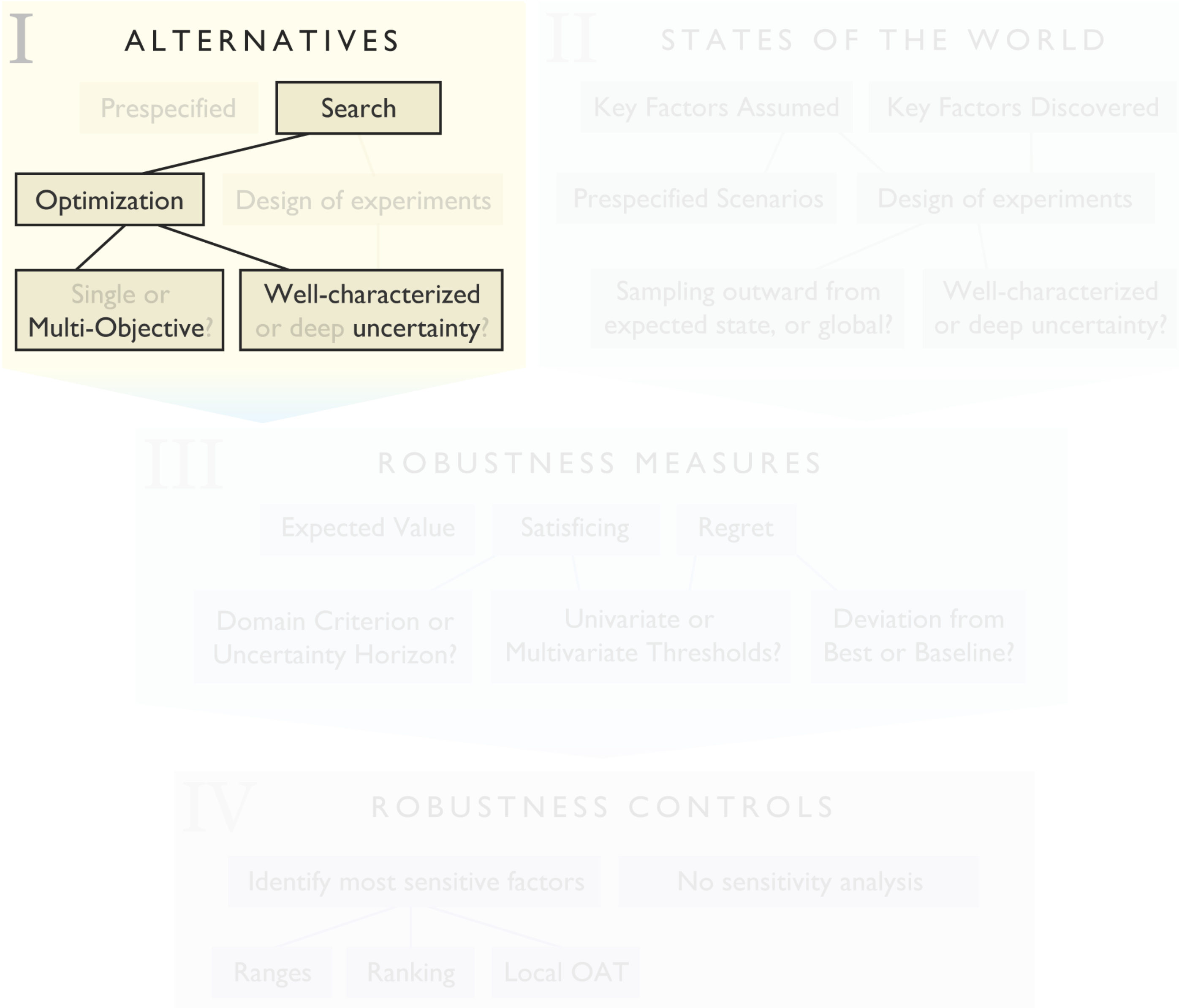
Quantify robustness measures and determine sensitive uncertainties

Slide credit: Jon Herman

Taxonomy of Robustness Frameworks



Taxonomy of Robustness Frameworks



Platypus

Evolutionary Algorithms / Optimization

Released (Stable)

python

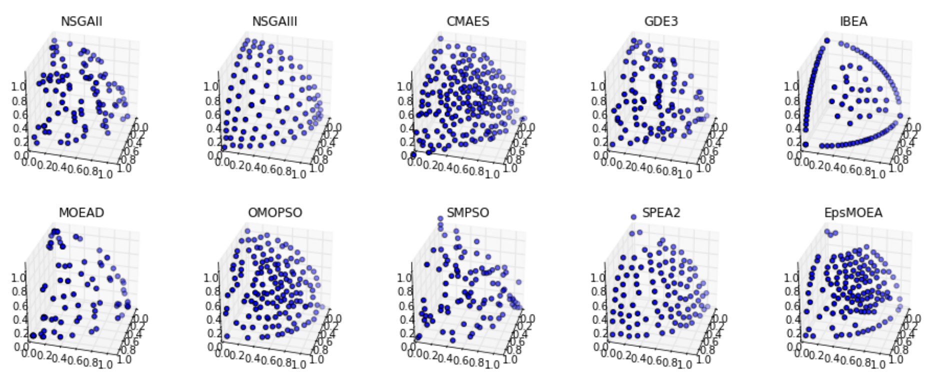
Implementation of popular single and multi-objective EAs:

- NSGA-II, NSGA-III, MOEA/D, IBEA, ε-MOEA, SPEA2, GDE3, OMOPSO, SMPSO, CMA-ES, ε-NSGA-II
- Borg (available separately)

```
from platypus import NSGAIII, DTLZ2

problem = DTLZ2(3)
algorithm = NSGAIII(problem, divisions_outer=12)
algorithm.run(10000)
print(algorithm.result)
```

Supports parallelization

- Threads
- MPI



Experimentation framework:

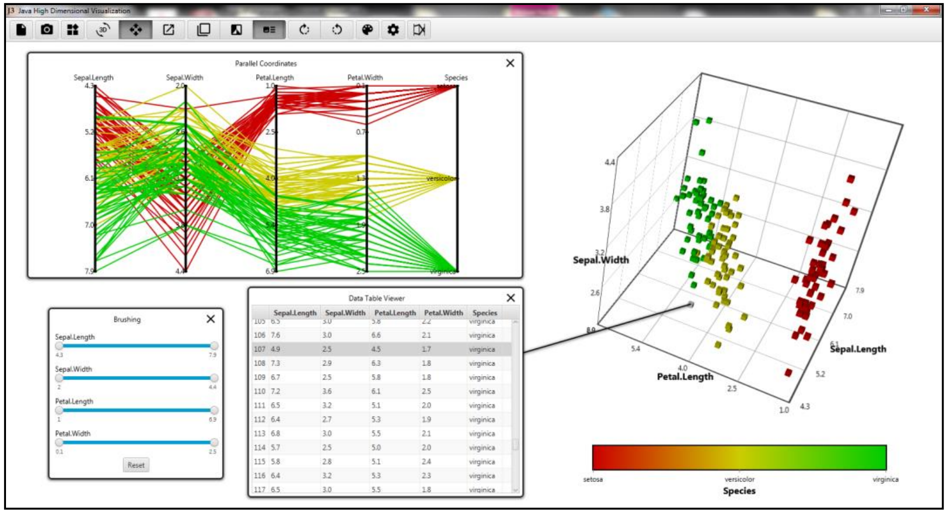
- Execute many algorithms / problems / parameters
- Parallelization support

J3
High Dimensional Visualizations in Java

Released (Stable)

Widget-based viewer

- 2D, 3D scatter plots
- Parallel coordinates
- Annotations, brushing
- Animations (New)
- Saving/loading (New)



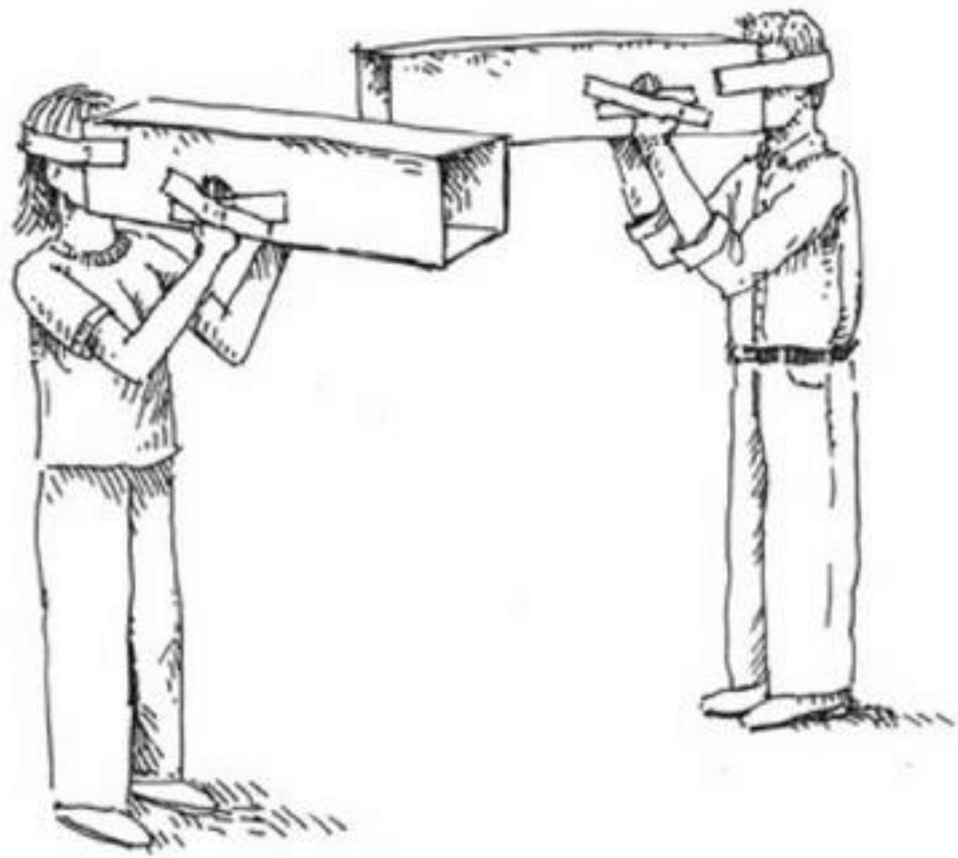
Cross Platform

- Installers for Windows, Mac, Linux

Small Footprint (10 MBs)

Plugin API for adding new widgets

Optimizing to a stationary future: too narrow?

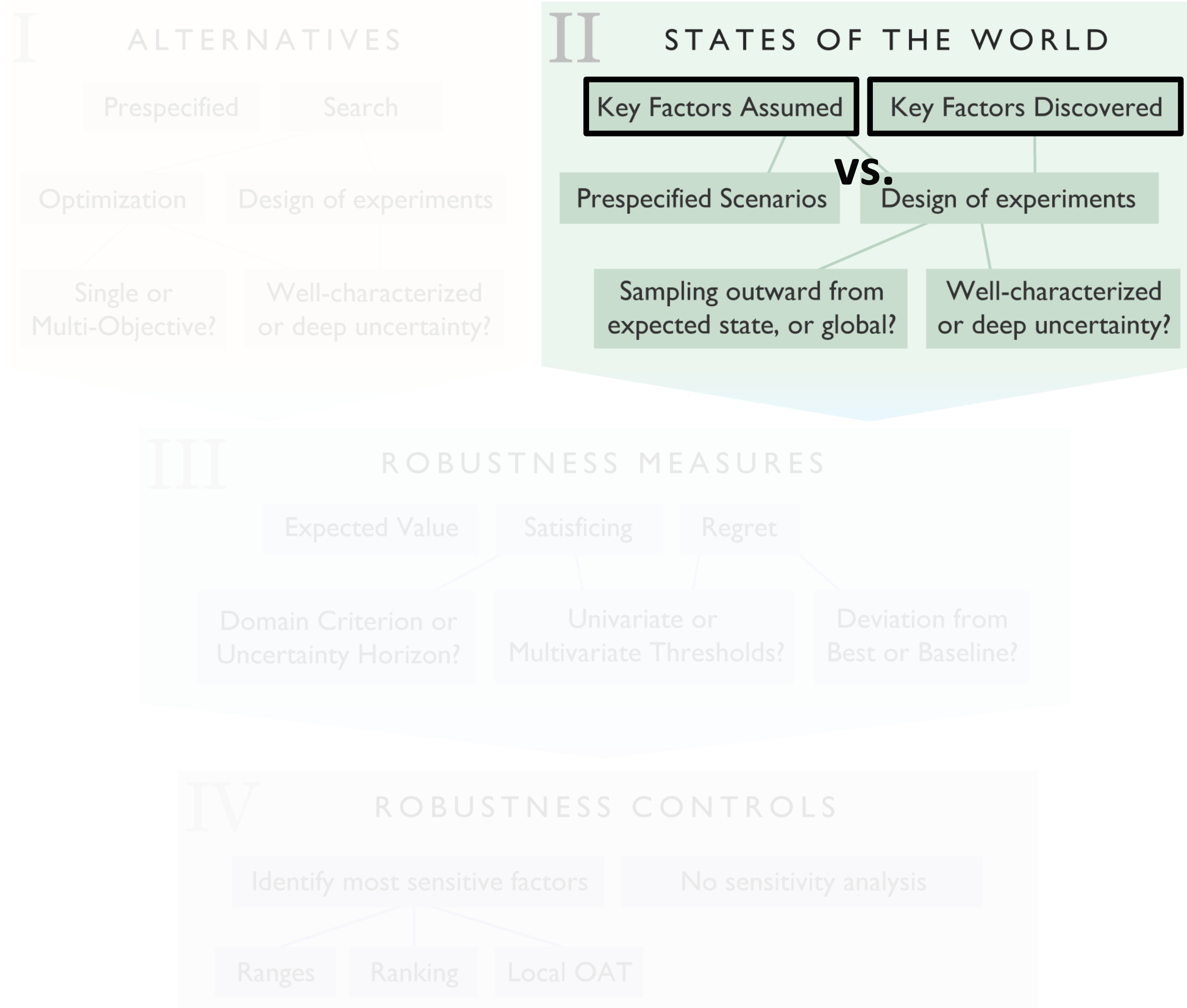


Which solutions are robust? How to decide?

Sample deeply uncertain states of the world (inflows, demand, etc.)

http://www.hockscqc.com/articles/tunnelvision/tunnel-vision.jpg

Taxonomy of Robustness Frameworks



Taxonomy of Robustness Frameworks

I ALTERNATIVES

- Prespecified
- Search
  - Optimization
  - Design of experiments
    - Single or Multi-Objective?
    - Well-characterized or deep uncertainty?

II STATES OF THE WORLD

- Key Factors Assumed
- Key Factors Discovered
  - Prespecified Scenarios
  - Design of experiments
    - Sampling outward from expected state, or global?
    - Well-characterized or deep uncertainty?

III ROBUSTNESS MEASURES

- Expected Value
- Satisficing
- Regret
  - Domain Criterion or Uncertainty Horizon?
  - Univariate or Multivariate Thresholds?
  - Deviation from Best or Baseline?

IV ROBUSTNESS CONTROLS

- Identify most sensitive factors
- No sensitivity analysis
  - Ranges
  - Ranking
  - Local OAT

Released (Stable)

Rhodium

Robust Decision Making

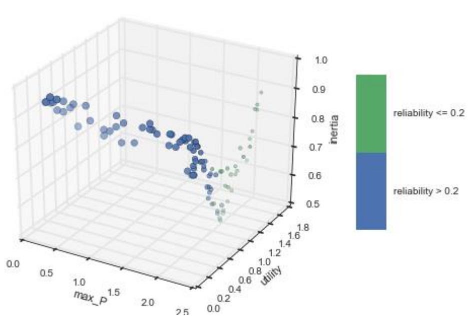
Decision support framework

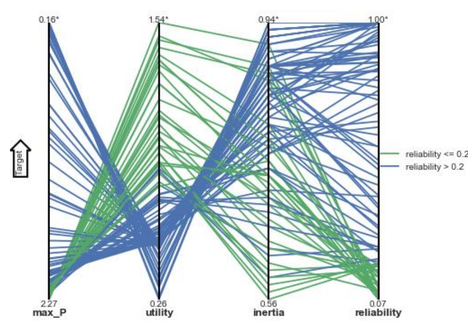
- Based on XLRM (Rand Corporation)
- Model-based, declarative design

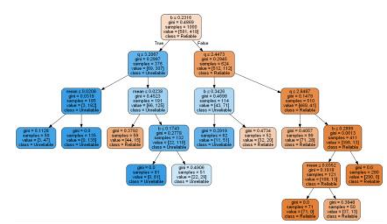
Acts as the "glue" between your model and analysis codes

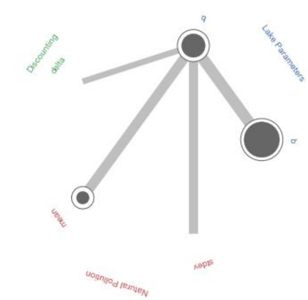
- Connects to models in C, C++, Fortran, Excel, R, Python, …
- Integrates with Pandas, Numpy, Platypus, PRIM, SALib, Scipy, ...

```
model = Model(lake_problem)

model.parameters = [Parameter("pollution_limit"),
                    Parameter("b"),
                    Parameter("q"),
                    Parameter("mean"),
                    Parameter("stdev"),
                    Parameter("delta")]

model.responses = [Response("max_P", MINIMIZE),
                   Response("utility", MAXIMIZE),
                   Response("inertia", MAXIMIZE),
                   Response("reliability", MAXIMIZE)]

model.constraints = [Constraint(
                     "reliability >= 0.95 and utility >= 0.25")]

model.levers = [RealLever("pollution_limit", 0.0, 0.1, length=100)]

model.uncertainties = [UniformUncertainty("b", 0.1, 0.45),
                       UniformUncertainty("q", 2.0, 4.5),
                       UniformUncertainty("mean", 0.01, 0.05),
                       UniformUncertainty("stdev", 0.001, 0.005),
                       UniformUncertainty("delta", 0.93, 0.99)]
```









Taxonomy of Robustness Frameworks

I ALTERNATIVES

- Prespecified
- Search
  - Optimization
    - Single or Multi-Objective?
    - Well-characterized or deep uncertainty?
  - Design of experiments

II STATES OF THE WORLD

- Key Factors Assumed
- Key Factors Discovered
  - Prespecified Scenarios
  - Design of experiments
    - Sampling outward from expected state, or global?
    - Well-characterized or deep uncertainty?

III ROBUSTNESS MEASURES

- Expected Value
- Satisficing
- Regret
  - Domain Criterion or Uncertainty Horizon?
  - Univariate or Multivariate Thresholds?
  - Deviation from Best or Baseline?

IV ROBUSTNESS CONTROLS

- Identify most sensitive factors
  - Ranges
  - Ranking
  - Local OAT
- No sensitivity analysis

Released (Stable)

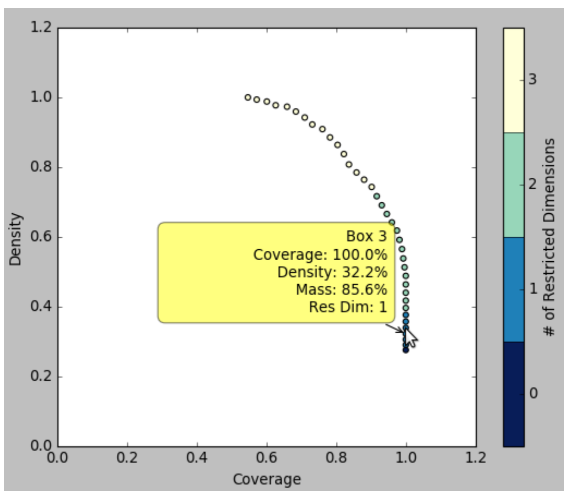
PRIM

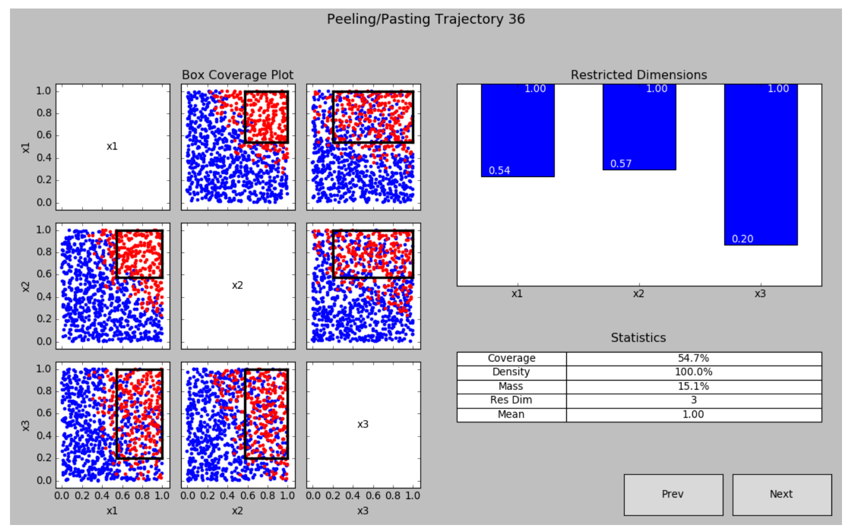
Patient Rule Induction Method

For scenario discovery

- Identify regions, aka “boxes”, containing cases of interest
- Simple rules (min/max bounds)
- Run interactively using Matplotlib or share static figures in IPython Notebooks

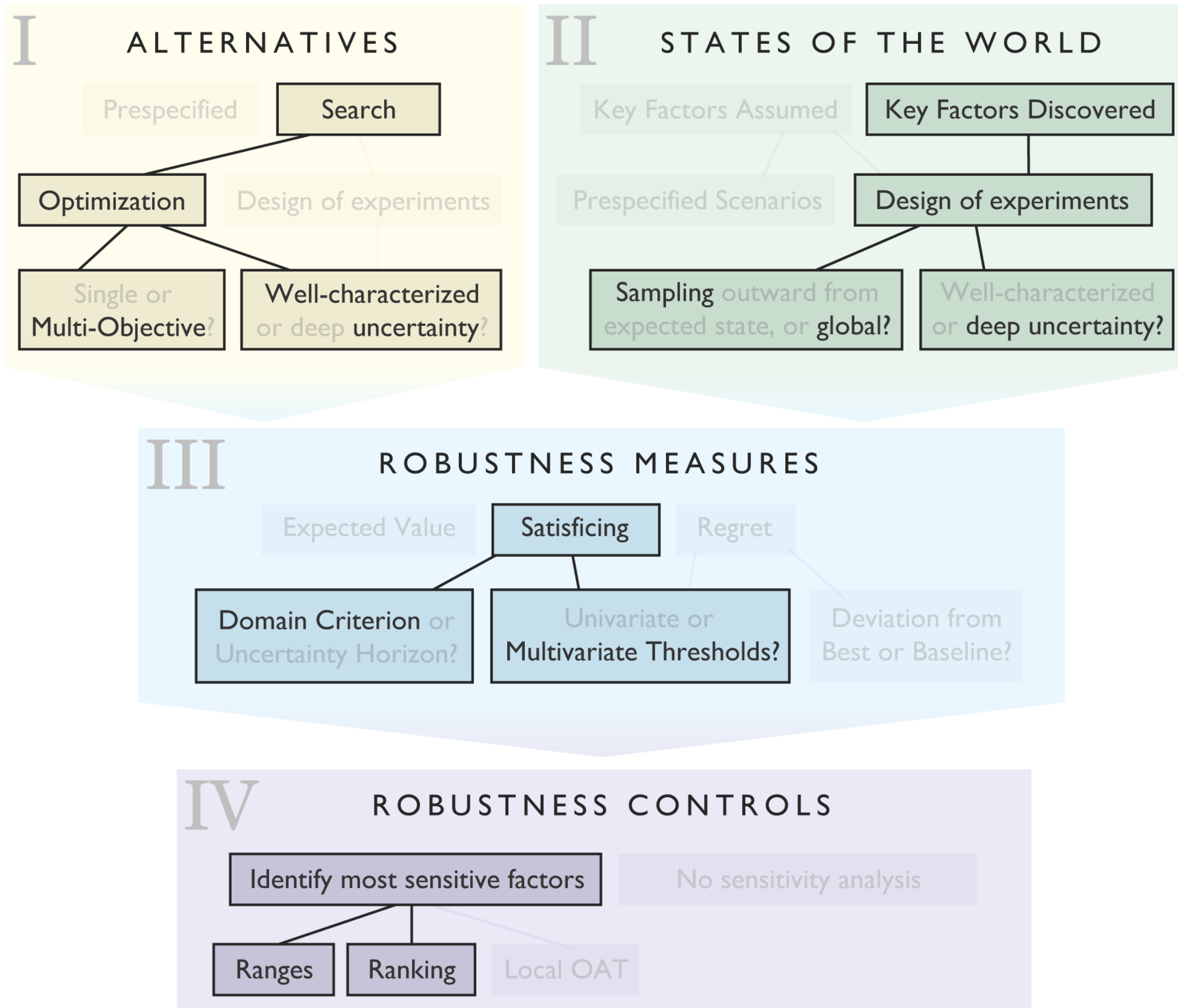
```
from prim import Prim

p = Prim(df, response, threshold=0.5,
         threshold_type=">")

box = p.find_box()
box.show_tradeoff()
```





http://localhost:8890/notebooks/Desktop/Project-Platypus/Rhodium/Rhodium-DPS-Lake.ipynb#Robustness-Analysis

Many-Objective Robust Decision Making



Project Platypus

A Collection of Libraries for Optimization, Data Analysis, and Decision Making

Get it today: https://github.com/Project-Platypus