Minpower Toolkit and Stochastic Scheduling

Adam Greenhall

08 March 2012

Alstom Grid

Minpower: a power systems optimization toolkit

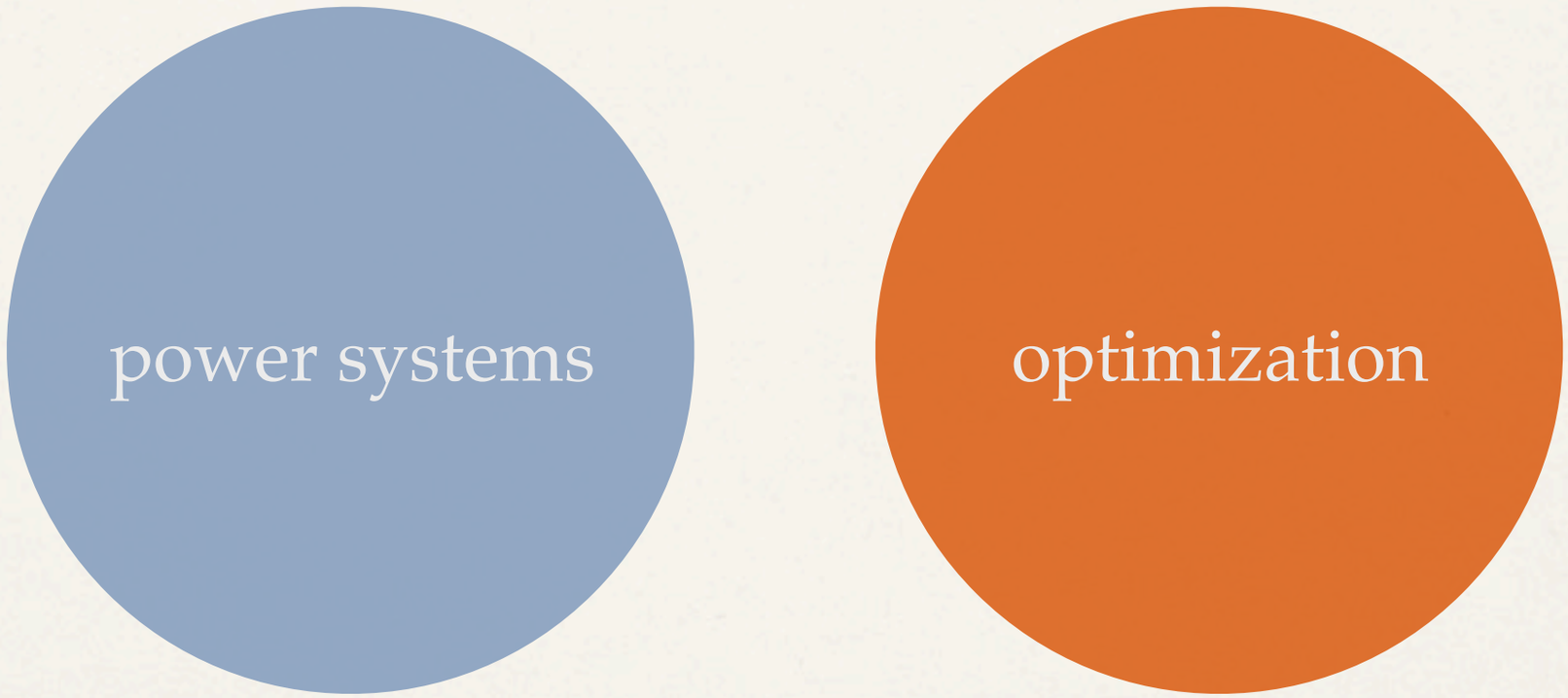
- Imagine starting from “scratch”
- How would you do it?
- What would you design for?

What is in this talk for Alstom Grid?

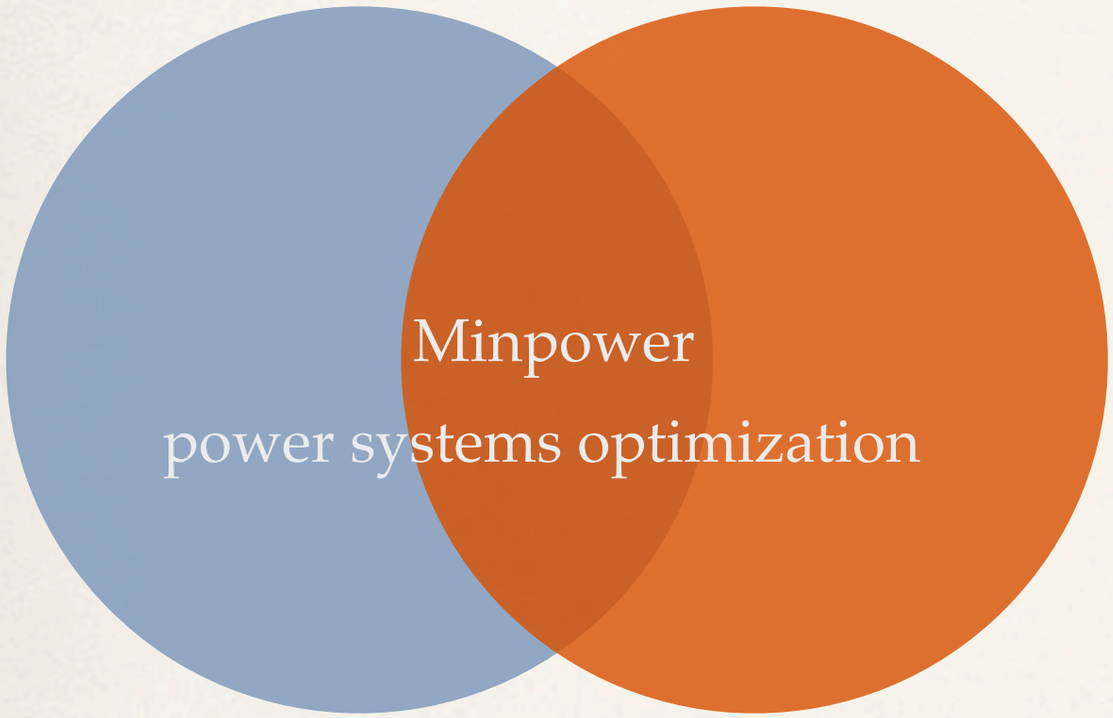
- Introduce you to new tools
- Reconsider design strategies
- A research or prototyping tool

Starting from more than scratch

Many existing tools



Selection Criteria



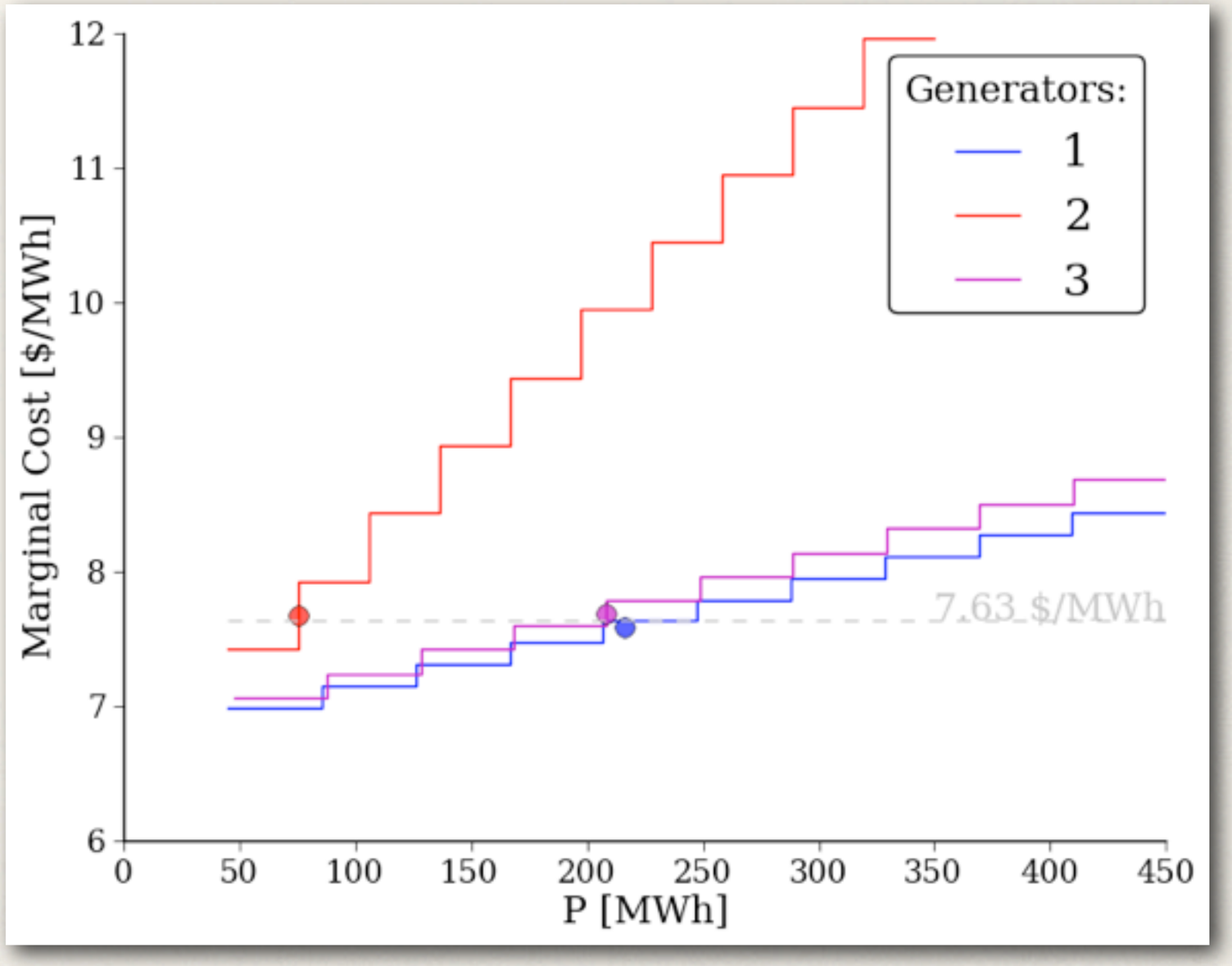
- Utilize what's out there (DRY)
- Short programming time, short solution times
- Publication, teaching, and collaboration

Toolkit Purpose

- Economic Dispatch
- Optimal Power Flow
- Unit Commitment

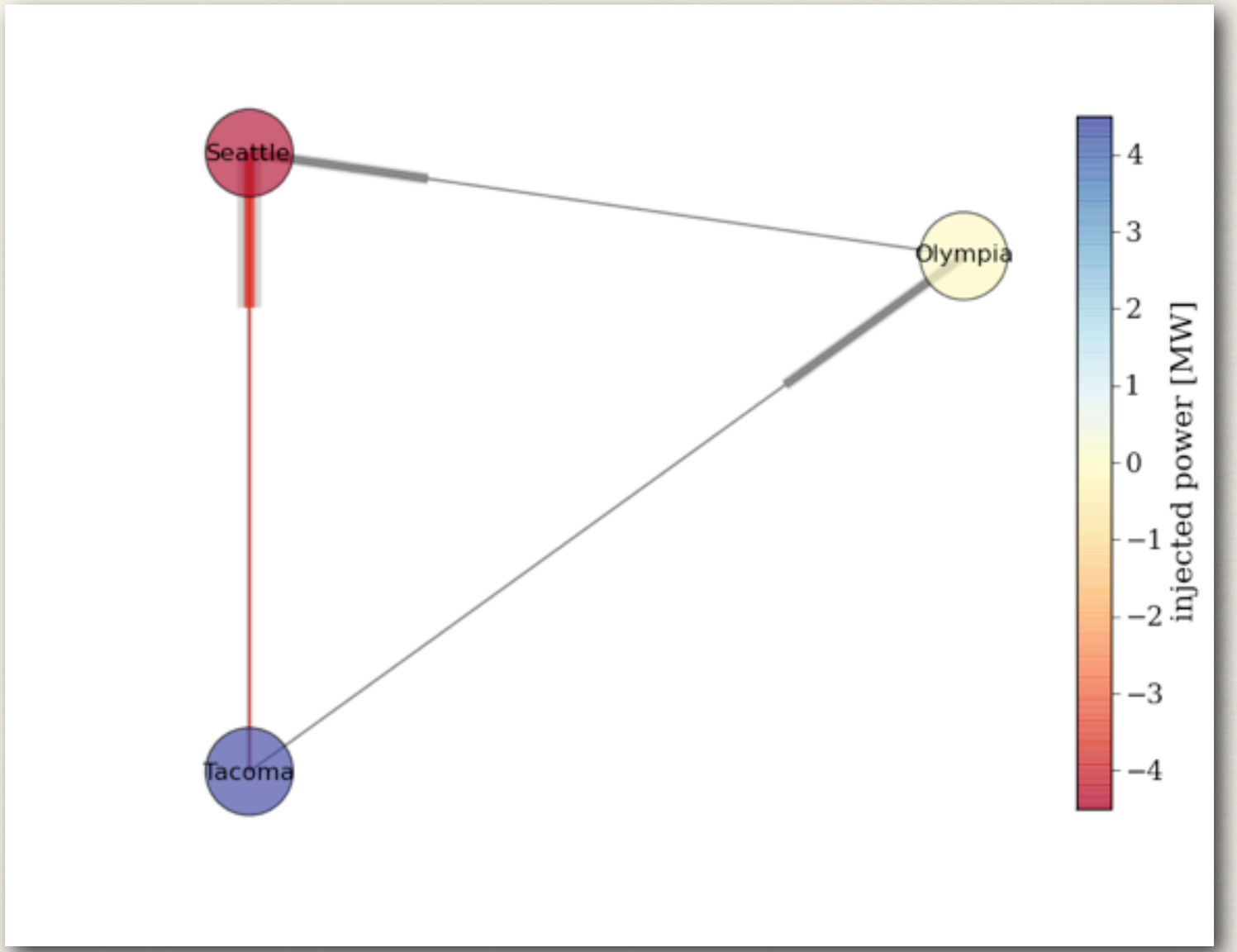
Toolkit Purpose

- Economic Dispatch
- Optimal Power Flow
- Unit Commitment



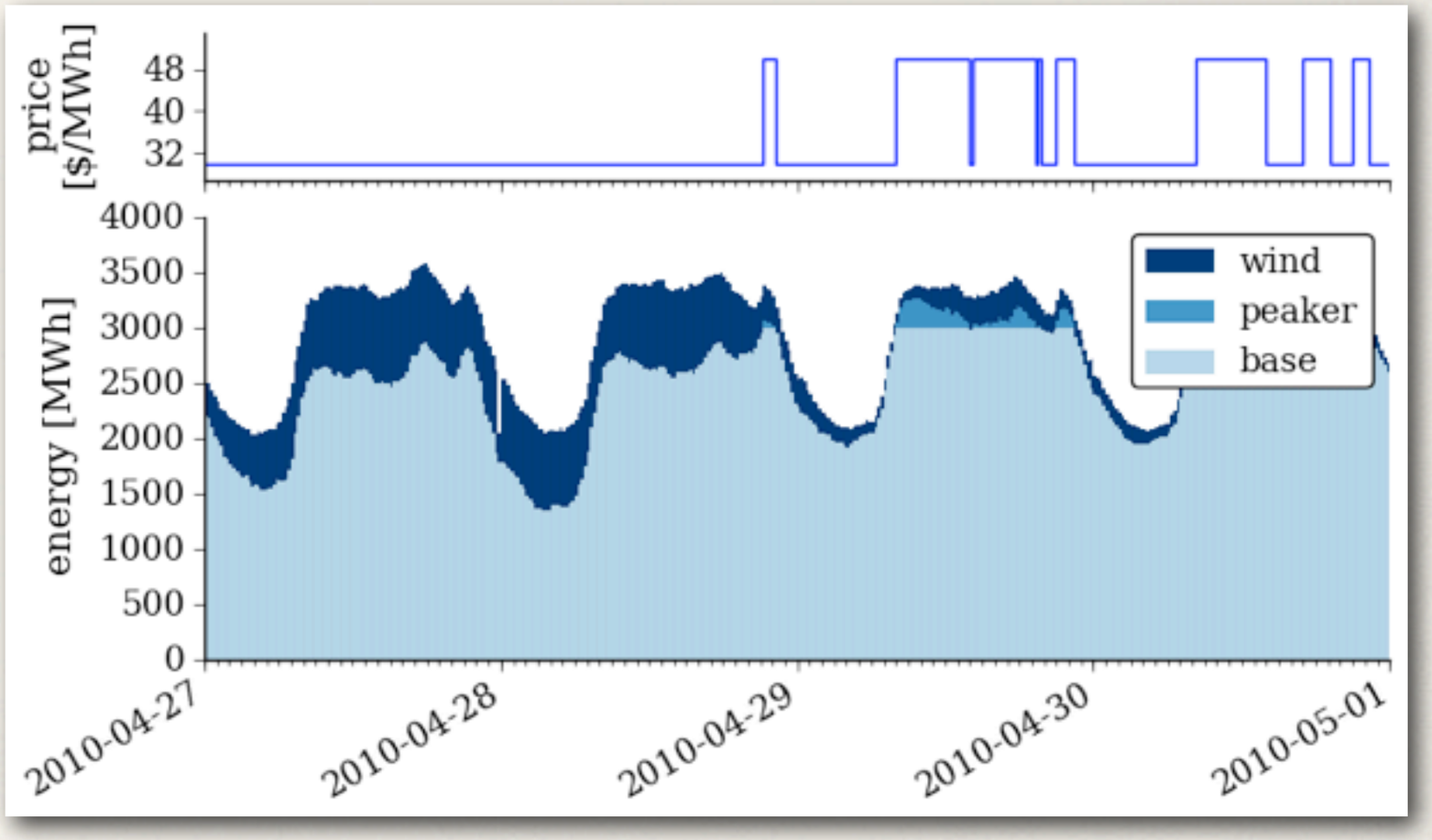
Toolkit Purpose

- Economic Dispatch
- Optimal Power Flow
- Unit Commitment



Toolkit Purpose

- Economic Dispatch
- Optimal Power Flow
- Unit Commitment



How do you use Minpower?

1. create problem:

generators:

	A	B	C	D
1	heat rate equation	P min	P max	fuel cost
2	225+8.4P+0.0025P^2	45	450	0.8
3	729+6.3P+0.0081P^2	45	350	1.02
4	400+7.5P+0.0025P^2	47.5	450	0.9

load:

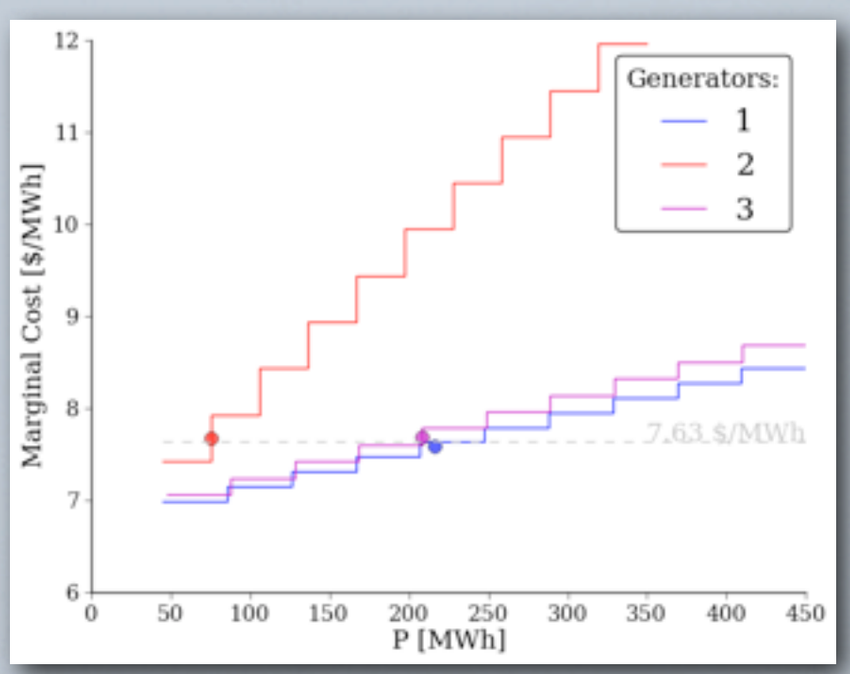
	A	B
1	name	power
2	load	500

2. solve:

```
adam@wirl:~$ minpower mydispatch/
```

3. view solution:

	A	B	C
1	u	P	IC
2	1	216	7.584
3	1	75.5	7.673562
4	1	208.5	7.68825



Two alternate ways to use

- code

Source code for powersystems

```
"""
Defines models for power systems components, including
:class:`~powersystems.PowerSystem`, :class:`~powersystems.Bus`,
:class:`~powersystems.Generator`, :class:`~powersystems.Load`,
and  :class:`~powersystems.Line`. Each of these objects inherits
an optimization framework from :class:`~optimization.OptimizationObject`.
"""

from optimization import value,dual,OptimizationObject
from commonscripts import hours,drop_case_spaces,flatten,getattrL,unique,
import config, bidding
from schedule import FixedSchedule
import logging
#import threading
import numpy

[docs]def makeGenerator(kind='generic',**kwargs):
    """
    Create a :class:`~powersystems.Generator` object
    (or a :class:`~powersystems.Generator_nonControllable`
    object depending on the kind). Set defaults
    depending on the kind (default values come from :mod:`config`).

    :param kind: define the kind of generator (all
        kinds are defined in :data:`config.generator_kinds`)

    Other parameters are detailed in :class:`~powersystems.Generator`.

    :returns: a :class:`~powersystems.Generator` object
    """

    def parse_args(kind,**inputs):
```

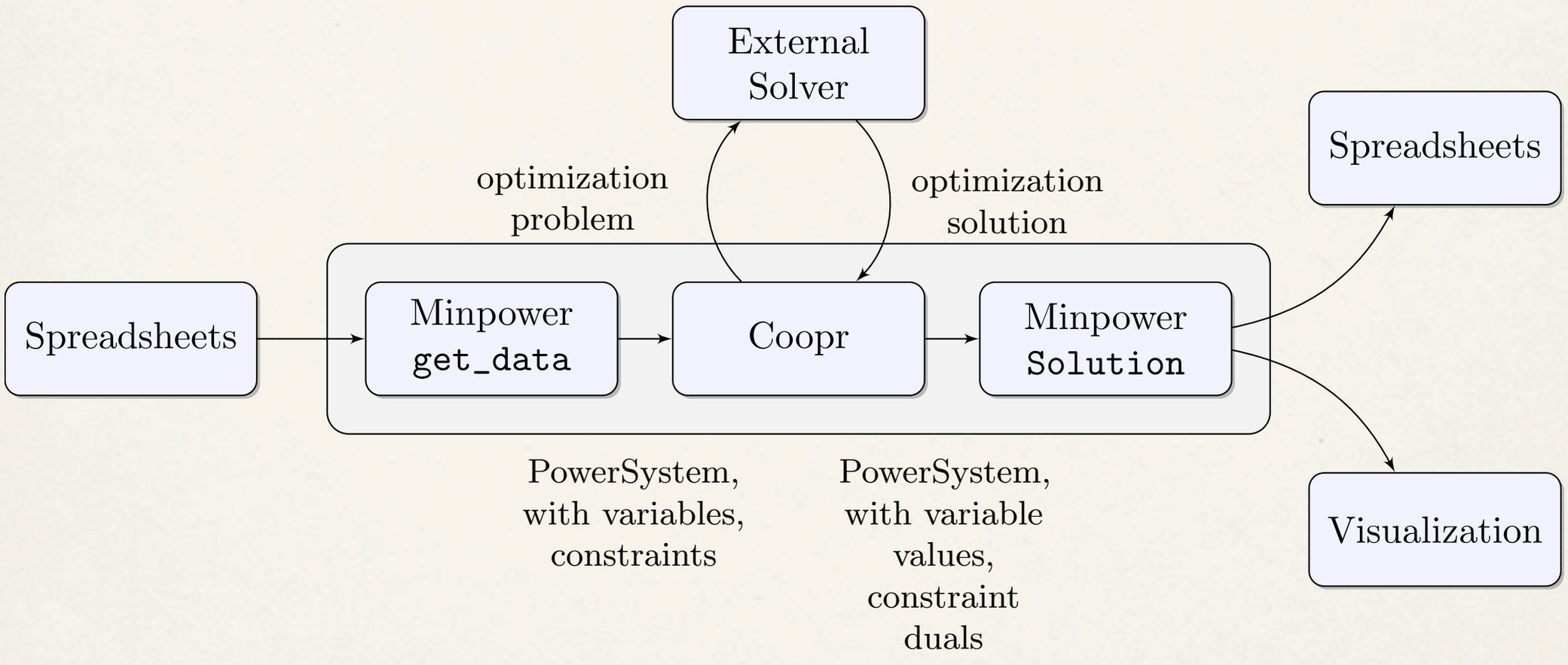
- a new browser-based UI

What's inside?

- Python
  - an open-source programming language
  - lots of science and math tools, including...

- Coopr
  - open-source optimization tools kit in Python
  - by Sandia National Labs - Optimization Dept.
  - work primarily in stochastic optimization
  - excellent collaborators

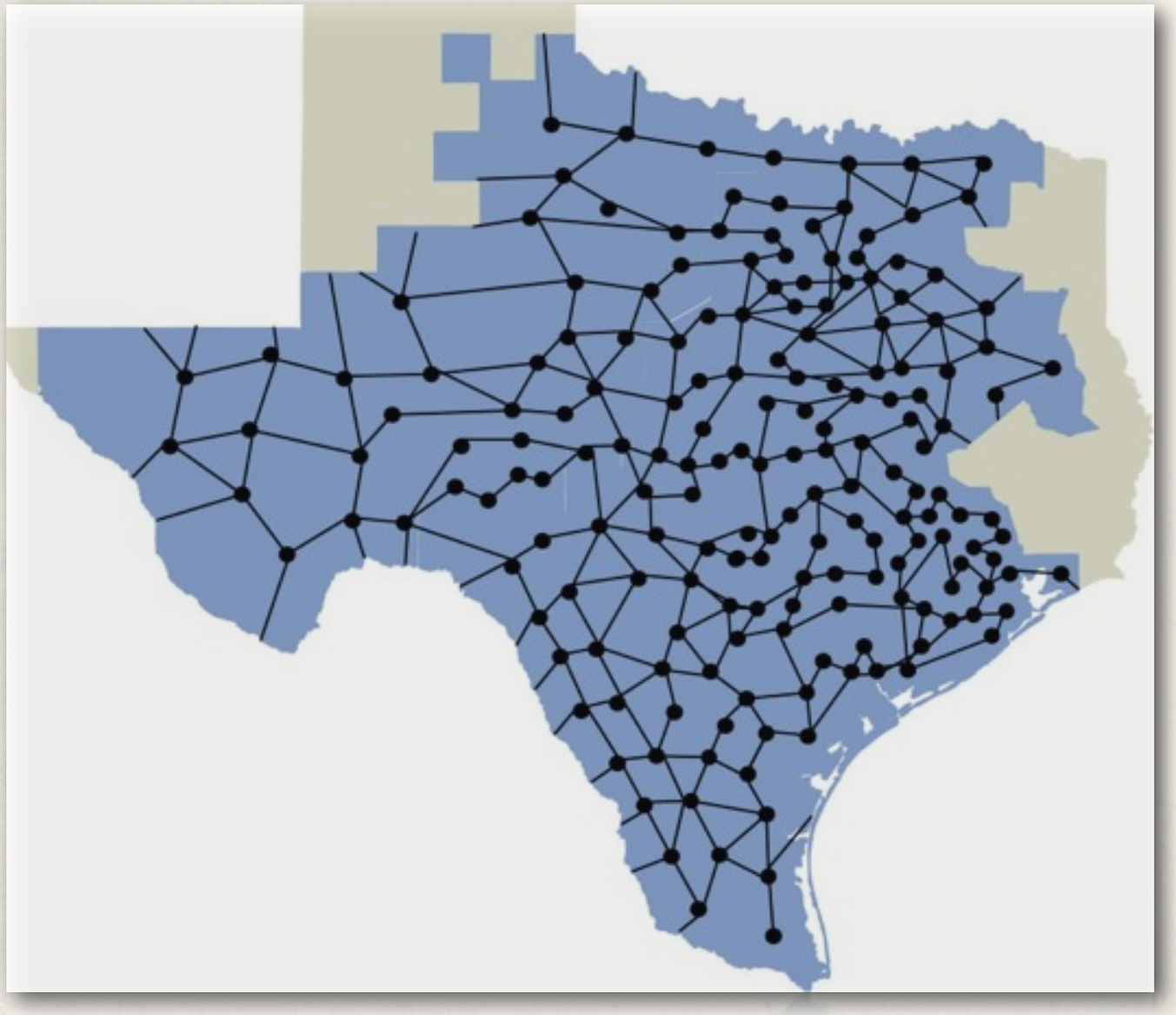
What's inside?



What's so good about that?

- free & open-source
- platform & solver independent
- easy to use
- documented

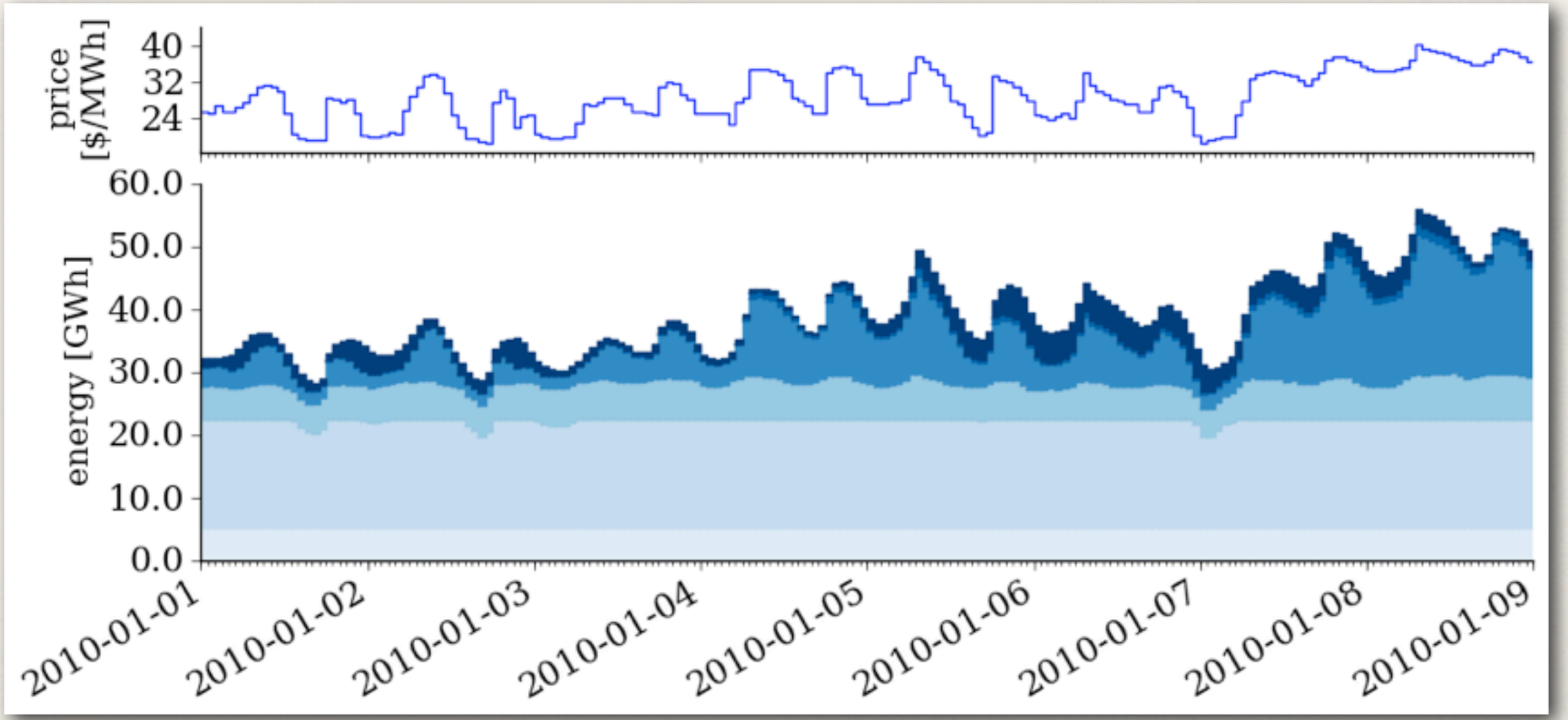
ERCOT model: Testing the limits



- wind (7GW peak in 2010)
- large system (~240 units)
- less complicated to model

ERCOT model: 2010 test results

run-time: 9 h 47 min
cost: $11.467 billion



Full year

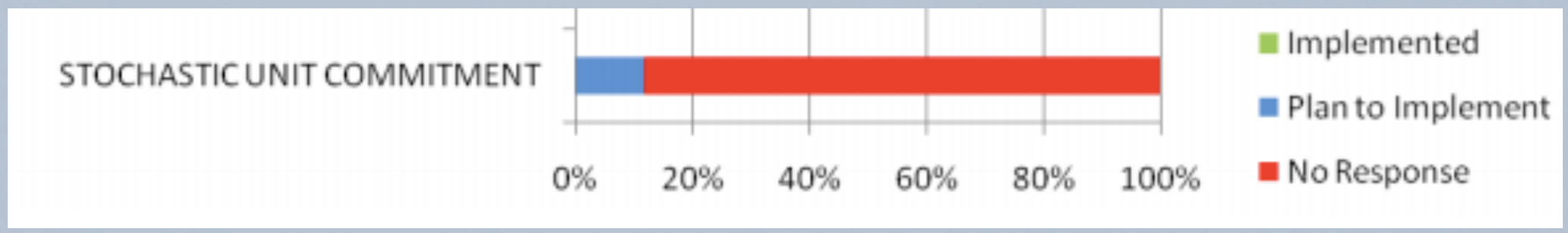
Stochastic Unit Commitment

Why Stochastic UC?

- the power system is becoming more stochastic
- system operators are having difficulty scheduling with wind
- huge potential savings in annual operating costs: ~1% by O'Malley's projection

Challenges: Stochastic UC

Current Plans:



Global System Operator Survey, Alstom Grid, 2012

Challenges: Stochastic UC

- Run-time
- Uncertainty about the solution
- Wind representation

Current work

- Generate realistic scenarios from your wind model
- Split the problem up into groups of scenarios
- Use a iterative decomposition method
  - Tighten the bounds on the solution
  - Find a good trade-off in time vs. certainty

Adam Greenhall

adam.greenhall@gmail.com

adamgreenhall.com

minpowertoolkit.com

Questions?