Optimizing Fuel Cycle Transitions Under Uncertainty

By Robert Carlsen

Technical Workshop on Fuel Cycle Simulation
Paris - July 6-8, 2016

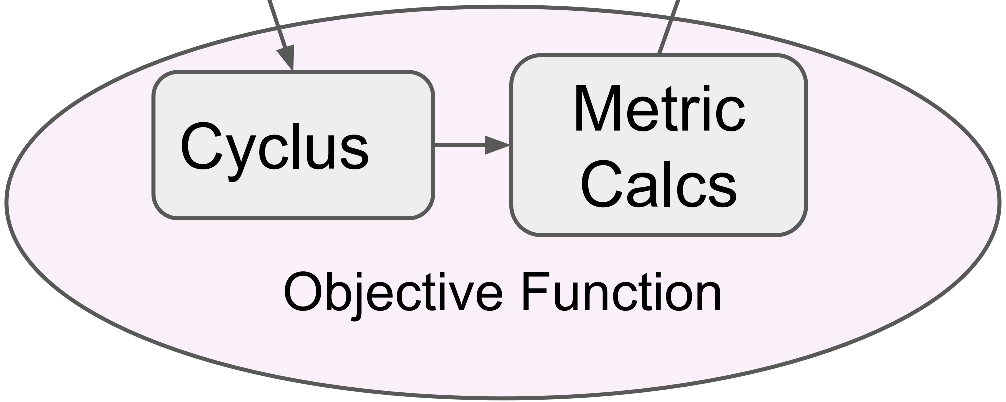
Background:

Fuel Cycle Simulation, Cyclus, and Optimization

Fuel Cycle Optimization: Basics

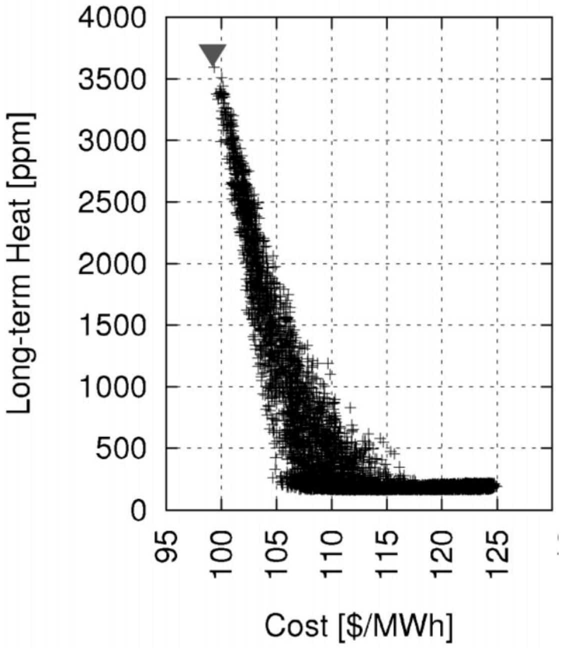
Facility	LWR				Repository			Fuel Fab	Objective
Year	1	2	3	…	1	2	…	…	
Trial 1	5	1	3	…	0	1	…	…	233.6
Trial 2	3	1	2	…	0	0	…	…	

Optimizer



Fuel Cycle Optimization: Hays' Work

- VISION simulator
- 100 discrete variables - fast v light water reactor ratios
- Homegrown simulated annealing
- Multi-objective analysis
- Auto-deployment heuristics
- Hard power capacity curve



Hays, Ross, and Paul Turinsky. 2014. "Stochastic Optimization for Nuclear Facility Deployment Scenarios Using Vision."
Nuclear Technology 186 (1): 76–89, Figure 3

Optimization Requirements

- Single Objective - expensive to evaluate
- Discrete Variables (100s)
- Black-box, derivative free
- Non-linear, discontinuous
- Linear constraints (transition and final state restrictions)

Optimization: Algorithms

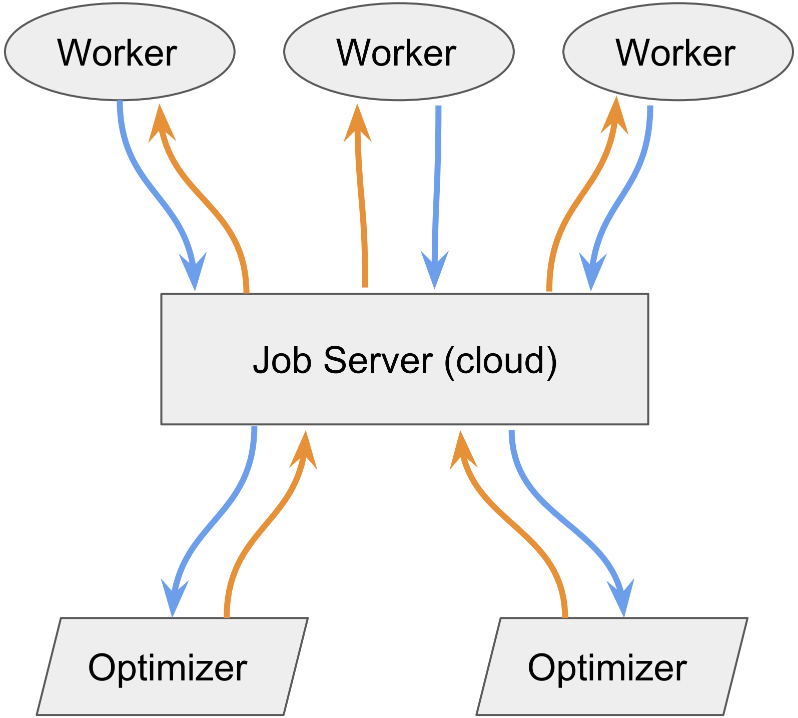
- Direct search (Nelder-Mead, pattern search, etc)
- DIviding RECTangles (DIRECT)
- Swarm (particle swarm, ant colony, etc.)
- Evolutionary algorithms
- Surrogate-based techniques (response sufaces, Kriging, etc.)

Custom PSwarm Optimizer

- Available open source version was unsatisfactory
  - Bugs (e.g. segfaults)
  - Poor code documentation and testing
  - Suboptimal parameters
- Wrote my own implementation in Go:
  - Tests - prevent regressions, benchmark performance with published lit.
  - Modified pattern-search algorithm:
    - Reset/cycle back to original step size (due to variable ⇒ deployments transformation)
    - Poll in multiple random directions at once (due to higher dimensionality)
    - Remember successful polling directions

HTC tooling

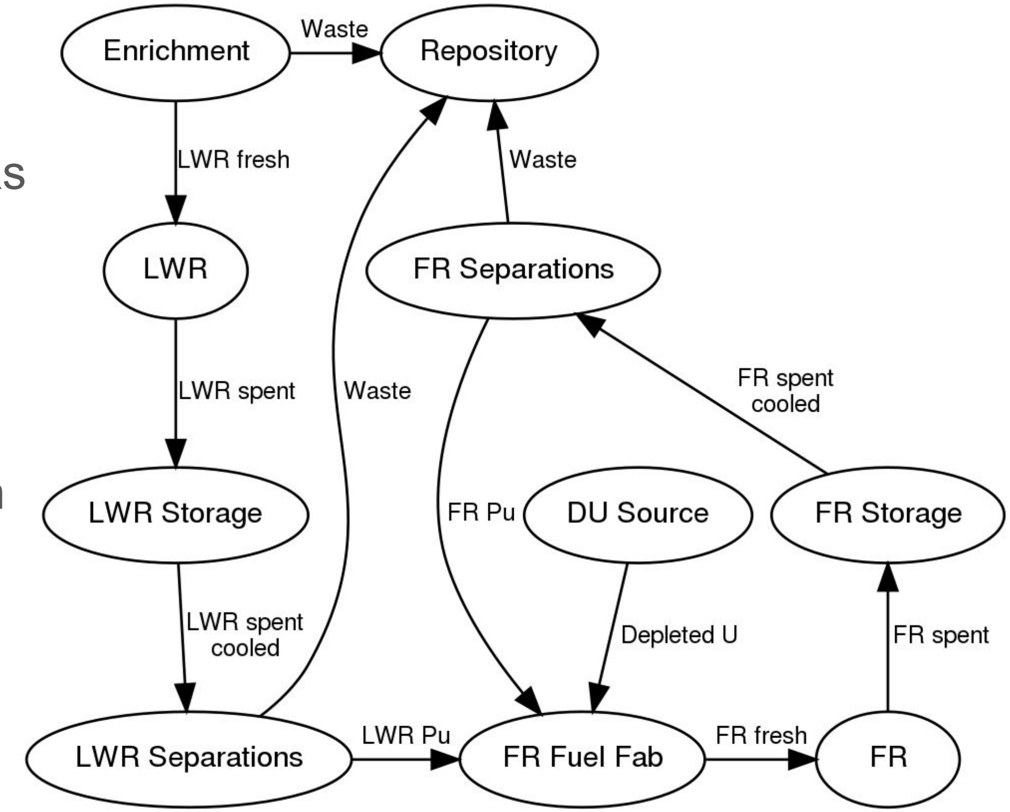
- Deploy worker bots/jobs to HTC infrastructure
- Cloud server M ⇒ N scheduling
  - Heartbeat, timeouts, retries, etc.
- Submit jobs to server from anywhere.
  - My PSwarm optimizer
  - fuelcycle.org
  - curl



Deployment Optimization Basics

Scenario

- Transition from 100 LWRs to all SFRs
- SFRs use recycled fuel
- SFRs available in year 35+
- 200 years
- 1% annual electricity demand growth with +/- 10% bounds



Objective Function

- Penalize LWR energy
- Reward FR energy
- Indirect unfueled FR penalty

$$O_{sim} = \frac{\sum_{t \in sim} E_{t,\,LWR}}{\sum_{t \in sim} E_{t,\,tot}}$$

Input Variables: Basic Encoding

$N(t, f) = V_{t,f}$

With constraints:

- For each t, f :

$$N_{min}(t, f) < V_{t,f} \leqslant N_{max}(t, f)$$

- For each t :

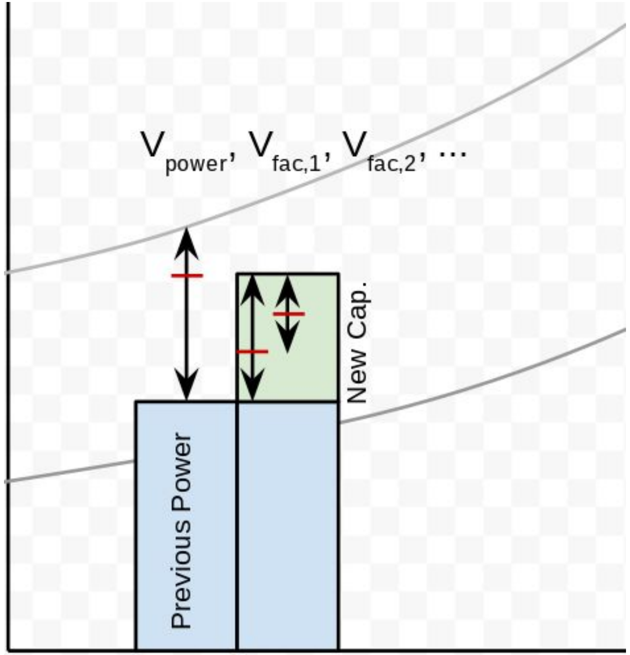
$$P_{min}(t) < \sum_{r \in reactors} C_{t,r} \cdot N_{alive}(t, V_r) < P_{max}(t)$$

Input Variables: Smarter Encoding

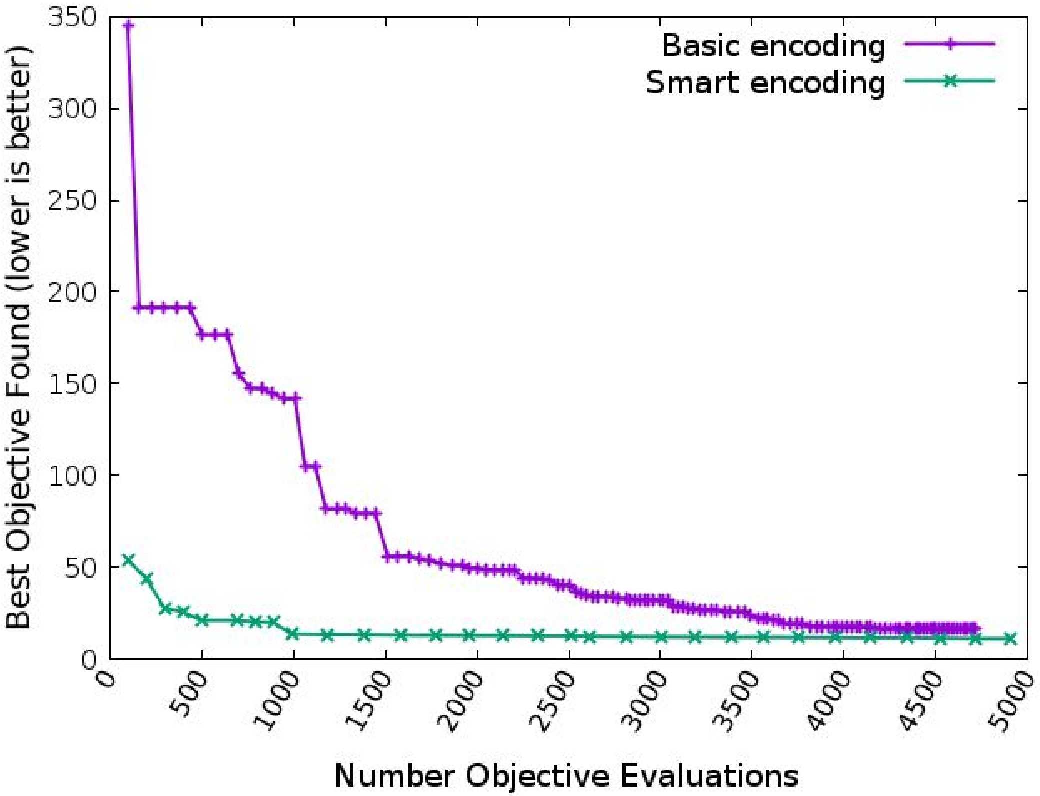
$$P_{new}(t) = \max\left(0, V_{power}(t) \cdot \max\left[0, P_{max}(t) - L(t)\right] + L(t) - P(t^-)\right)$$

$$L(t) = \max(P(t^-), P_{min}(t))$$

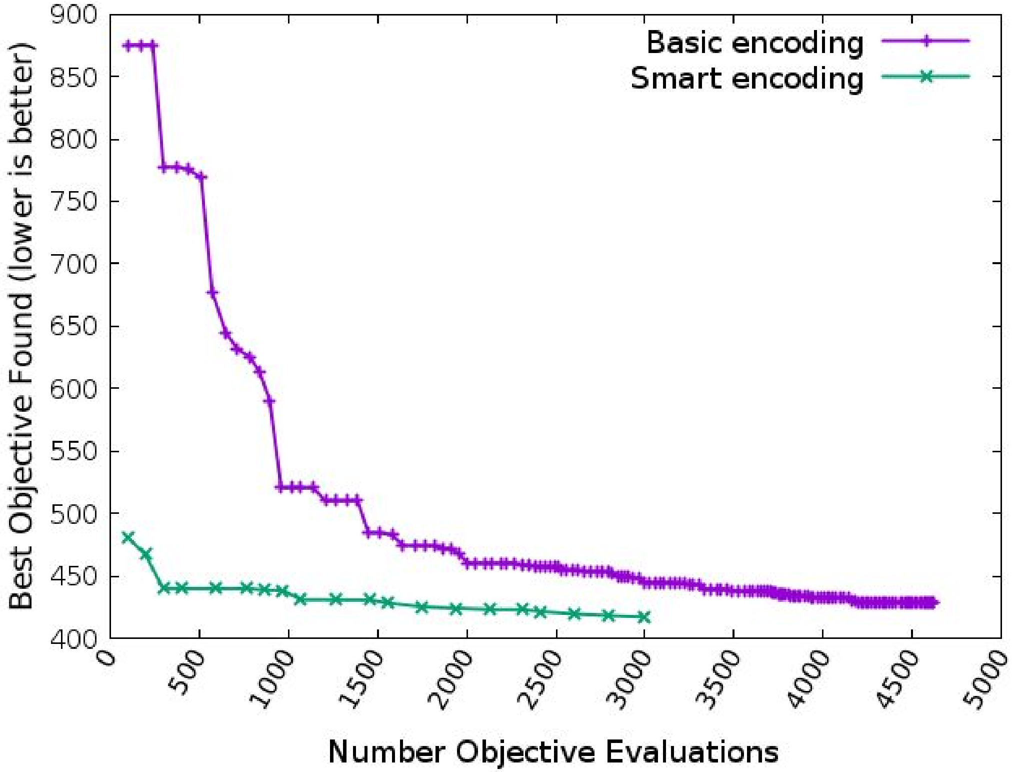
$$N(t,r) = \begin{cases} \text{floor}\left(\frac{V_{fac}(t,r) \cdot \left[P_{new}(t) - \sum_{r'=1}^{r-1} N(t,r') \cdot C(r')\right]}{C(r)} + 0.5\right) & : r > 0 \\ \text{floor}\left(\frac{P_{new}(t) - \sum_{r'=1}^{r_{last}} N(t,r') \cdot C(r')}{C(r)} + 0.5\right) & : r = 0 \end{cases}$$



Input Variables: Comparison - Objective A



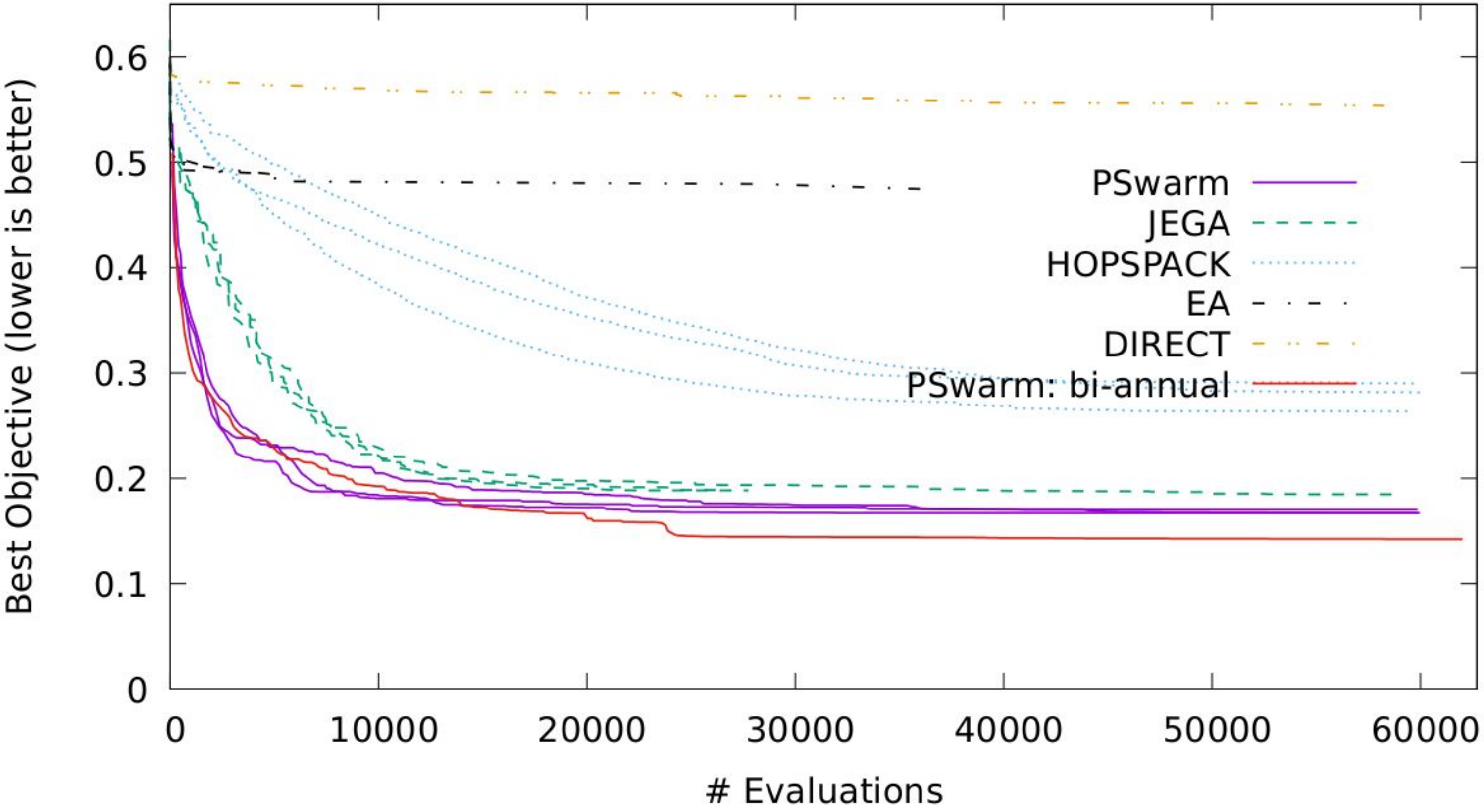
Input Variables: Comparison - Objective B



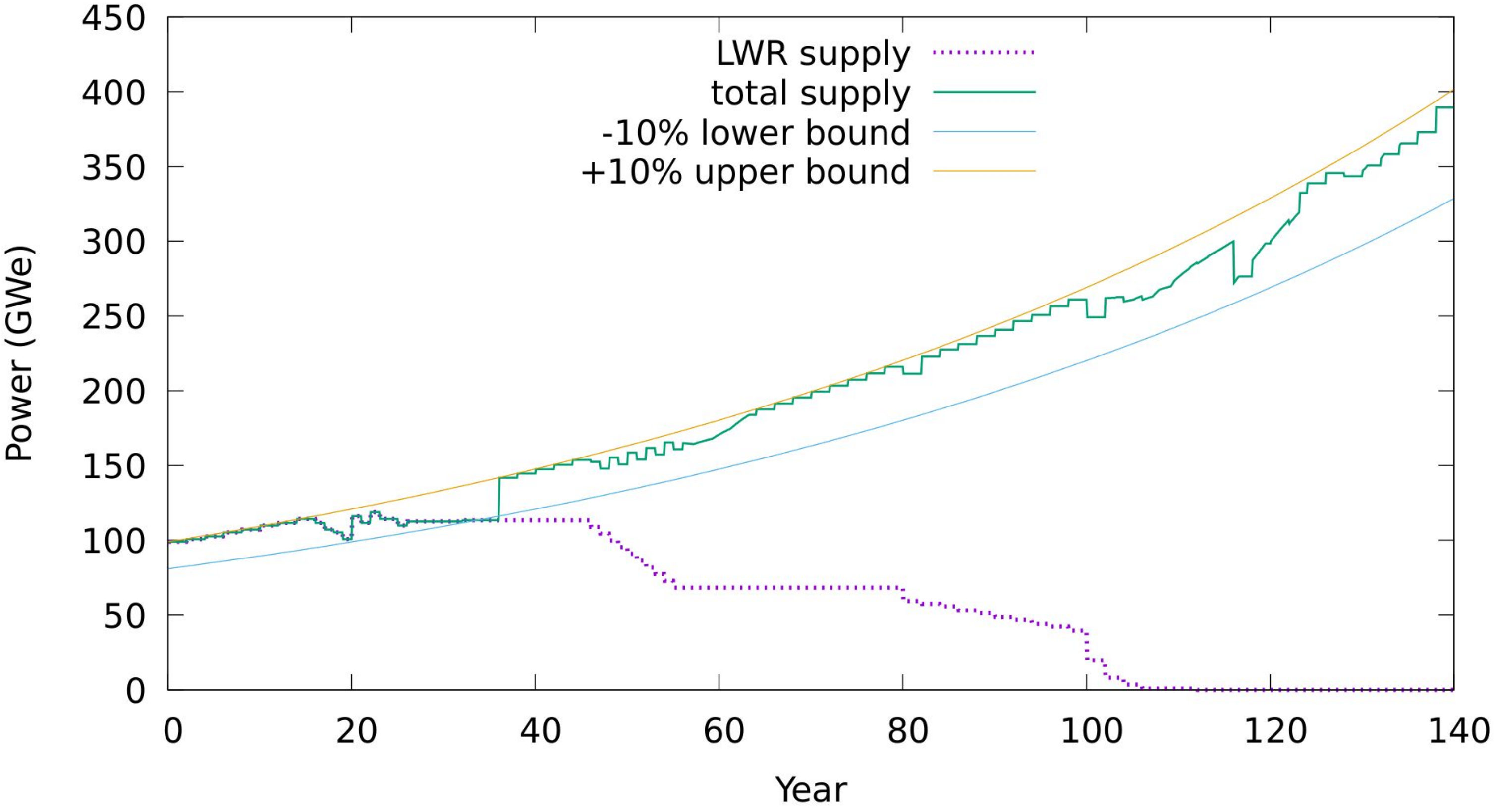
Candidate Optimizers

	1. Max # vars	2. Parallelism	3. Robustness	4. Configuration
Custom PSwarm	Unlimited	Good	Good	Easy
JEGA	Unlimited	Good	Good	Tricky
SCOLIB EA	Unlimited	Good	Good	Tricky
NCSU DIRECT	64	Poor	Okay	Easy
SCOLIB DIRECT	1000s	Great	Okay	Easy
NOMAD	<=1000	Poor	Good	Easy
HOPSPACK	Unlimited	Great	Good	Easy
Surrogate methods	10s to 100s	Good	Poor	Tricky

Results: Solver Comparison



Results: Best Transition (Bi-annual)



Hedging Strategies for Disruption

Hedging Overview

- Same scenario as before
- Potential unexpected event at unknown time
- Measure hedging value of deployment strategies
- Find good hedging strategies

Hedging Objective

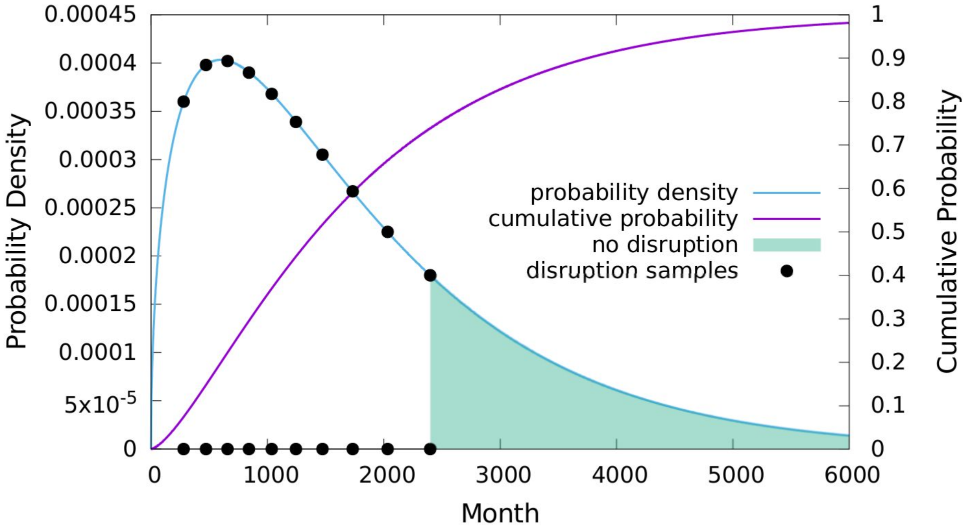
$$S^*(D, t_d) = O[R^*(D, t_d), t_d] \qquad H(D) = \int_0^\infty S^*(D, t_d) \cdot p(t_d) dt_d$$

- D is a deployment schedule (all facs through all time)
- $R^*(D,t_d)$ is D with optimal post disruption deployments
- O is some single-objective function, as before
- $S^*(D,t_d)$ is the hedging sub-objective
- p is the disruption probability density function
- H is the expected objective outcome

Calculating H

$$H(D) = \int_0^\infty S^*(D, t_d) \cdot p(t_d) dt_d$$

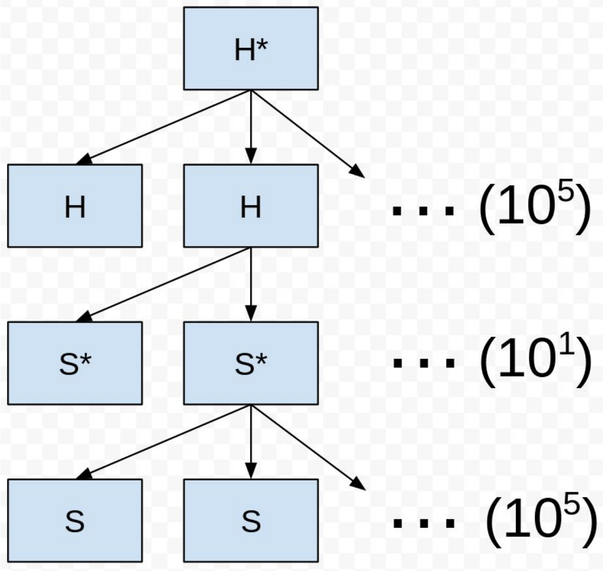
- Discretize on t_d
- t_d's spaced equally in probability space
- Piece-wise linear approximation to S*
- Use mid-point rule to integrate



$$p(t_d) = \frac{600}{\Gamma(1.5) \cdot 2^{1.5}} \left(\frac{t}{600}\right)^{0.5} e^{-\frac{t}{600 \cdot 2}}$$

Hedging Sub-objective: Point Approximation

- one H* search ⇒ many H evaluations
  - one H evaluation ⇒ several S* searches
    - one S* search ⇒ many S (or O) evaluations



- Infeasible nested optimization requires approximations:

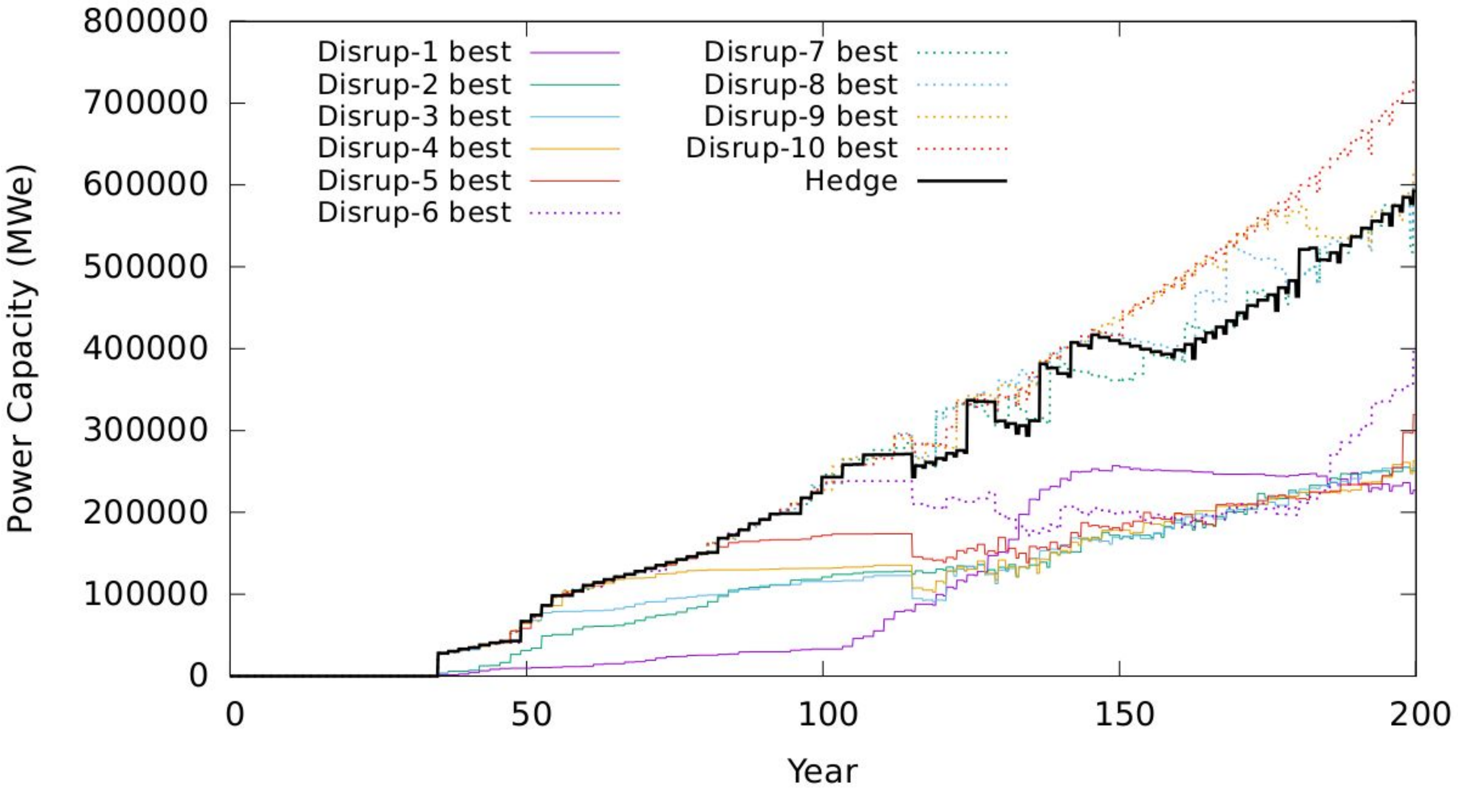
$$S^*(D, t_d) \approx \frac{t_d}{t_{end}} \cdot O(D, t_d) + \frac{t_{end} - t_d}{t_{end}} \cdot O^*(t_d)$$

Scenario Details

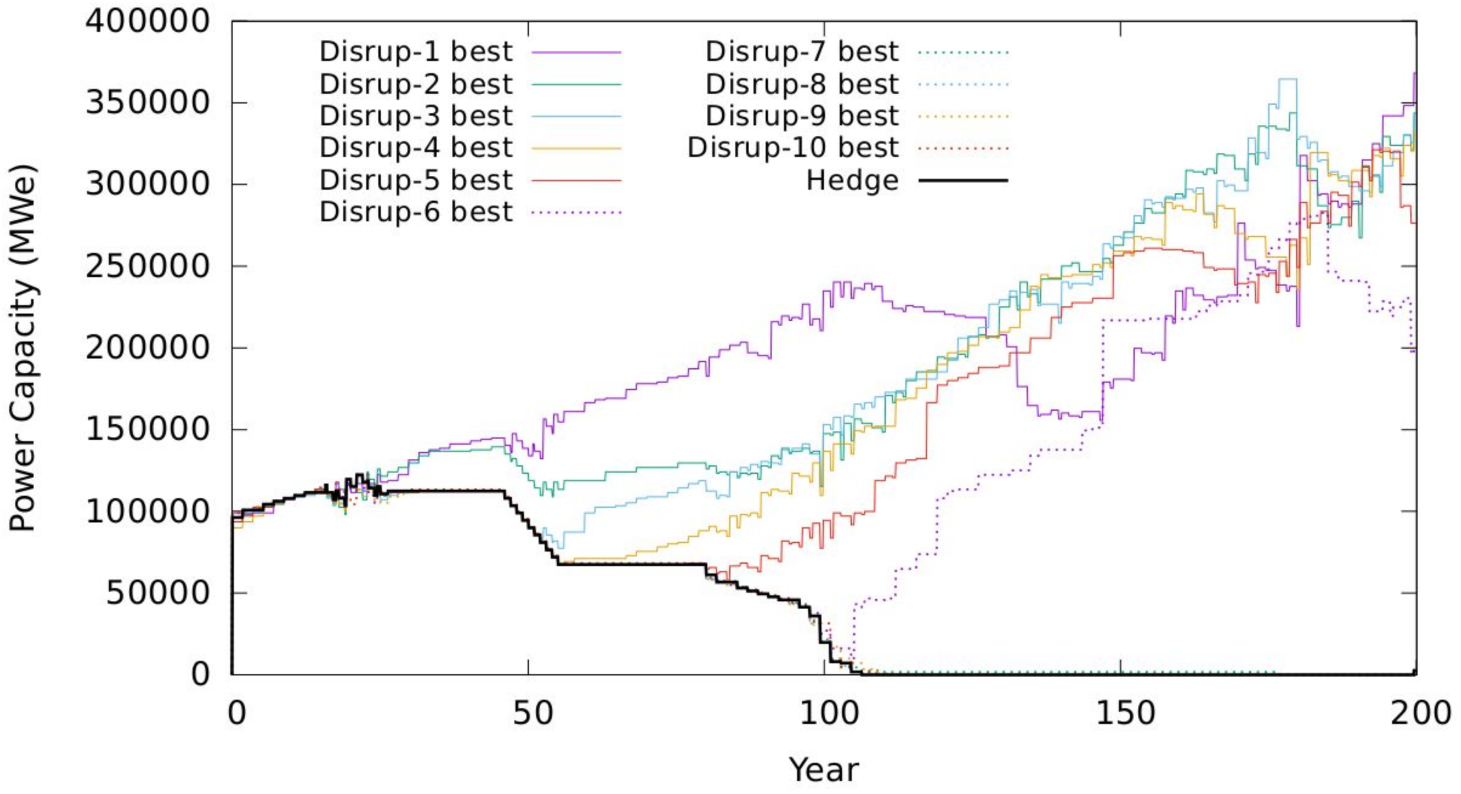
- Same scenario sans LWR reprocessing limits.
- Disruption reduces Pu generation rate by 33% permanently.
- Objective:
    - Lower ⇒ better
    - LWR energy penalty
    - Explicit low capacity factor penalty (needed for disruption)

$$O_{sim} = \frac{\sum_{t \in sim} E_{t,\,LWR}}{\sum_{t \in sim} E_{t,\,tot}} \cdot \frac{\sum_{t \in sim} C_{t,\,tot}}{\sum_{t \in sim} E_{t,\,tot}}$$

Results: SFR Build Schedule Comparison



Results: LWR Build Schedule Comparison



Results: Best Achievable Objectives

Deployment Strategy	Disruption Time (year)									
	23	39	55	70	86	104	123	144	170	200
$D^*(t_d = 23)$	**0.654**									
$D^*(t_d = 39)$		**0.633**								
$D^*(t_d = 55)$			**0.618**							
$D^*(t_d = 70)$				**0.586**						
$D^*(t_d = 86)$					**0.545**					
$D^*(t_d = 104)$						**0.479**				
$D^*(t_d = 123)$							**0.384**			
$D^*(t_d = 144)$								**0.209**		
$D^*(t_d = 170)$									**0.152**	
$D^*(t_d = 200)$										**0.142**
D^*_H										

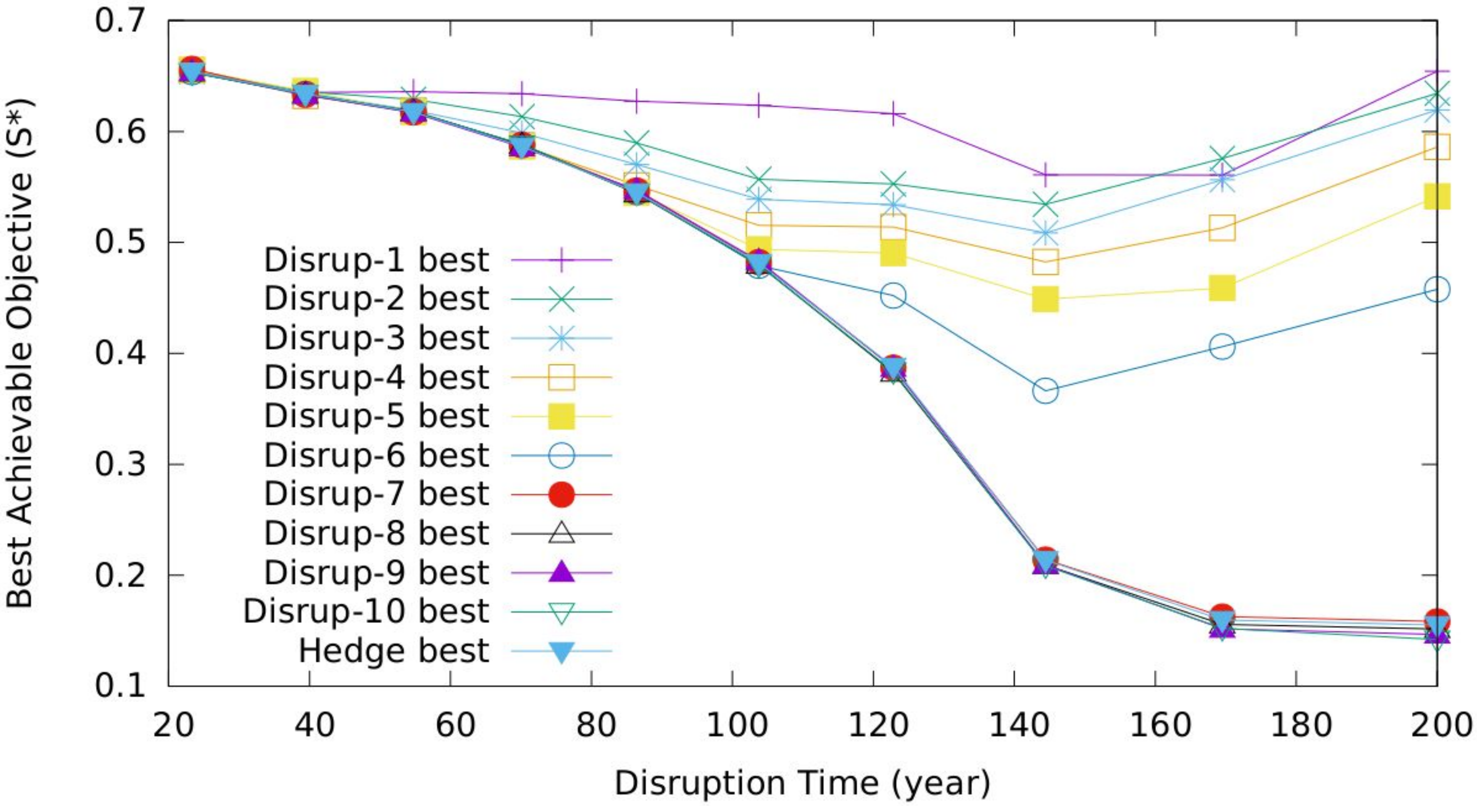
Results: Best Achievable Objectives

Deployment Strategy	Disruption Time (year) 23	39	55	70	86	104	123	144	170	200
$D^*(t_d = 23)$	**0.654**									
$D^*(t_d = 39)$		**0.633**								
$D^*(t_d = 55)$			**0.618**							
$D^*(t_d = 70)$				**0.586**						
$D^*(t_d = 86)$					**0.545**					
$D^*(t_d = 104)$						**0.479**				
$D^*(t_d = 123)$							**0.384**			
$D^*(t_d = 144)$								**0.209**		
$D^*(t_d = 170)$									**0.152**	
$D^*(t_d = 200)$										**0.142**
D^*_H	0.654	0.634	0.618	0.586	0.545	0.481	0.388	0.214	0.160	0.155

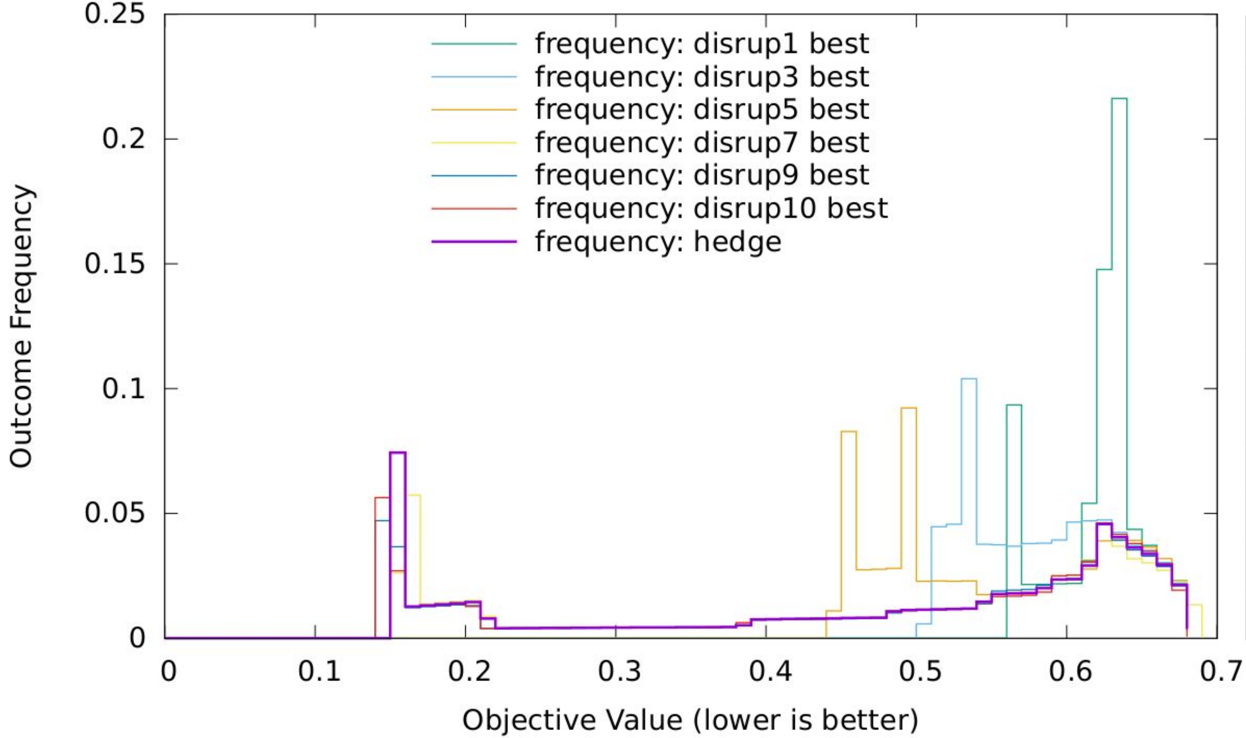
Results: Best Achievable Objectives

Deployment Strategy	Disruption Time (year)									
	23	39	55	70	86	104	123	144	170	200
$D^*(t_d = 23)$	**0.654**	0.635	0.636	0.634	0.627	0.624	0.616	0.561	0.561	0.654
$D^*(t_d = 39)$	0.654	**0.633**	0.629	0.613	0.590	0.557	0.553	0.534	0.576	0.634
$D^*(t_d = 55)$	0.655	0.635	**0.618**	0.599	0.570	0.539	0.534	0.509	0.557	0.619
$D^*(t_d = 70)$	0.655	0.632	0.619	**0.586**	0.551	0.515	0.514	0.483	0.513	0.586
$D^*(t_d = 86)$	0.656	0.637	0.618	0.587	**0.545**	0.494	0.490	0.449	0.459	0.542
$D^*(t_d = 104)$	0.654	0.634	0.618	0.588	0.545	**0.479**	0.452	0.366	0.406	0.458
$D^*(t_d = 123)$	0.656	0.633	0.617	0.588	0.547	0.482	**0.384**	0.214	0.163	0.158
$D^*(t_d = 144)$	0.653	0.634	0.618	0.588	0.544	0.480	0.383	**0.209**	0.156	0.151
$D^*(t_d = 170)$	0.654	0.633	0.617	0.586	0.547	0.483	0.387	0.209	**0.152**	0.147
$D^*(t_d = 200)$	0.653	0.634	0.618	0.588	0.545	0.480	0.384	0.209	0.152	**0.142**
D^*_H	0.654	0.634	0.618	0.586	0.545	0.481	0.388	0.214	0.160	0.155

Results: S* Curves



Results: Outcome Distributions



Deployment Strategy	Mean Objective Outcome
$D^*(t_d = 23)$	0.720
$D^*(t_d = 39)$	0.665
$D^*(t_d = 55)$	0.655
$D^*(t_d = 70)$	0.644
$D^*(t_d = 86)$	0.630
$D^*(t_d = 104)$	0.562
$D^*(t_d = 123)$	0.385
$D^*(t_d = 144)$	0.381
$D^*(t_d = 170)$	0.379
$D^*(t_d = 200)$	0.373
D^*_H	0.383

Some Stats

- Cyclus simulations: ~15,000,000
- CPU hours: ~300,000
- Simultaneous workers: 200-1000
- Simultaneous optimizations: up to 10
- Dreams in code: several

Summary

- Developed techniques for optimizing fuel cycle transitions.
  - Novel mapping of variables to fuel cycle parameters.
  - Tooling for deployment to highly parallel environments.
- Compared DFO solvers on fuel cycle transitions.
- Developed disruption scenario methodology and workflow
  - disruption PDF, expected outcomes
  - sub-objective approximation techniques
  - Tooling and visualization for measuring and finding hedging strategies
- Investigated hedging properties of several deployment schedules

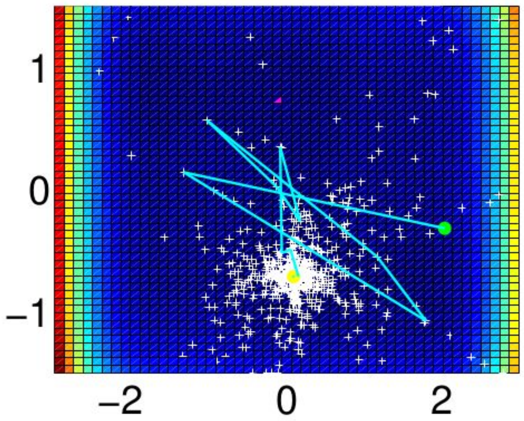
Future Work

- Dimensionality Reduction
  - “Compression” via variables that represent time+capacity points
  - Intra-simulation heuristics to micro-optimize at shorter timescales (e.g. look-ahead)
  - Translate known good (abstract) disruption responses to post-disruption deployments
- Different approximations of S*
- N disruptions or degrees of uncertainty
- More realism...
  - in facility models (e.g. reactor physics)
  - in scenario details (e.g. power demand, reactor parameters, etc.)
  - in disruption/objective details (e.g. objective changes at disruption)

Questions

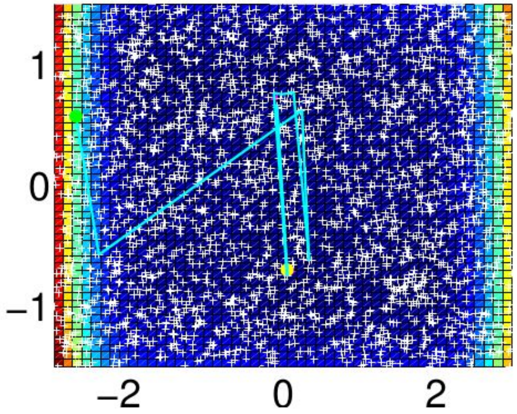
Appendix

Optimization: Solvers



PSWARM

- Pattern search + particle swarm
- Continuous variables
- Search confined to feasible region
- Population-parallel

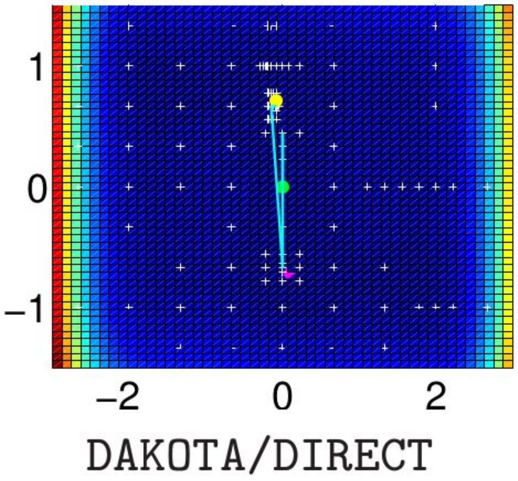


DAKOTA/EA

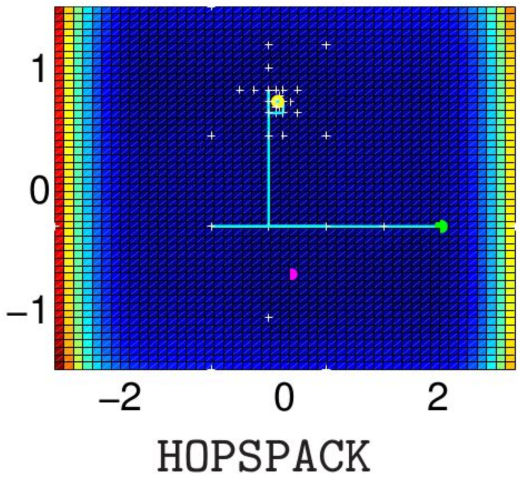
- Evolutionary algorithm
- Discrete+continuous variables
- Penalty-based constraints
- Population-parallel

From Figure 2 of Rios, Luis Miguel, and Nikolaos V. Sahinidis. 2013. “Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations.” *Journal of Global Optimization* 56 (3): 1247–93. doi:10.1007/s10898-012-9951-y.

Optimization: Solvers cont.



- Pattern search + particle swarm
- Continuous variables
- Search confined to feasible region
- Population-parallel



- Pattern search
- Discrete+continuous variables
- Penalty-based constraints
- N-dimensions parallel

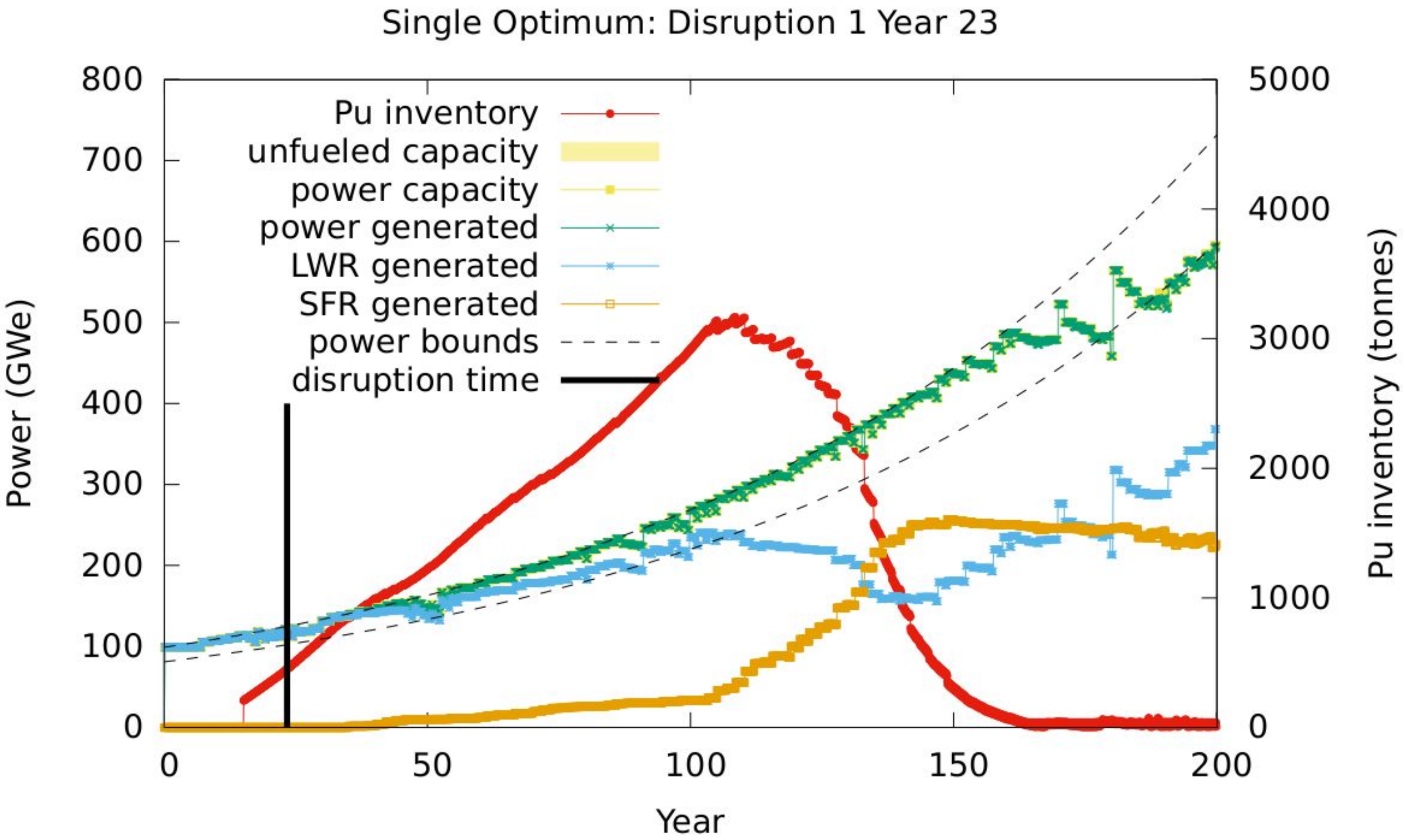
From Figure 2 of Rios, Luis Miguel, and Nikolaos V. Sahinidis. 2013. "Derivative-Free Optimization: A Review of Algorithms and Comparison of Software Implementations." *Journal of Global Optimization* 56 (3): 1247–93. doi:10.1007/s10898-012-9951-y.

Results: S* Approximations

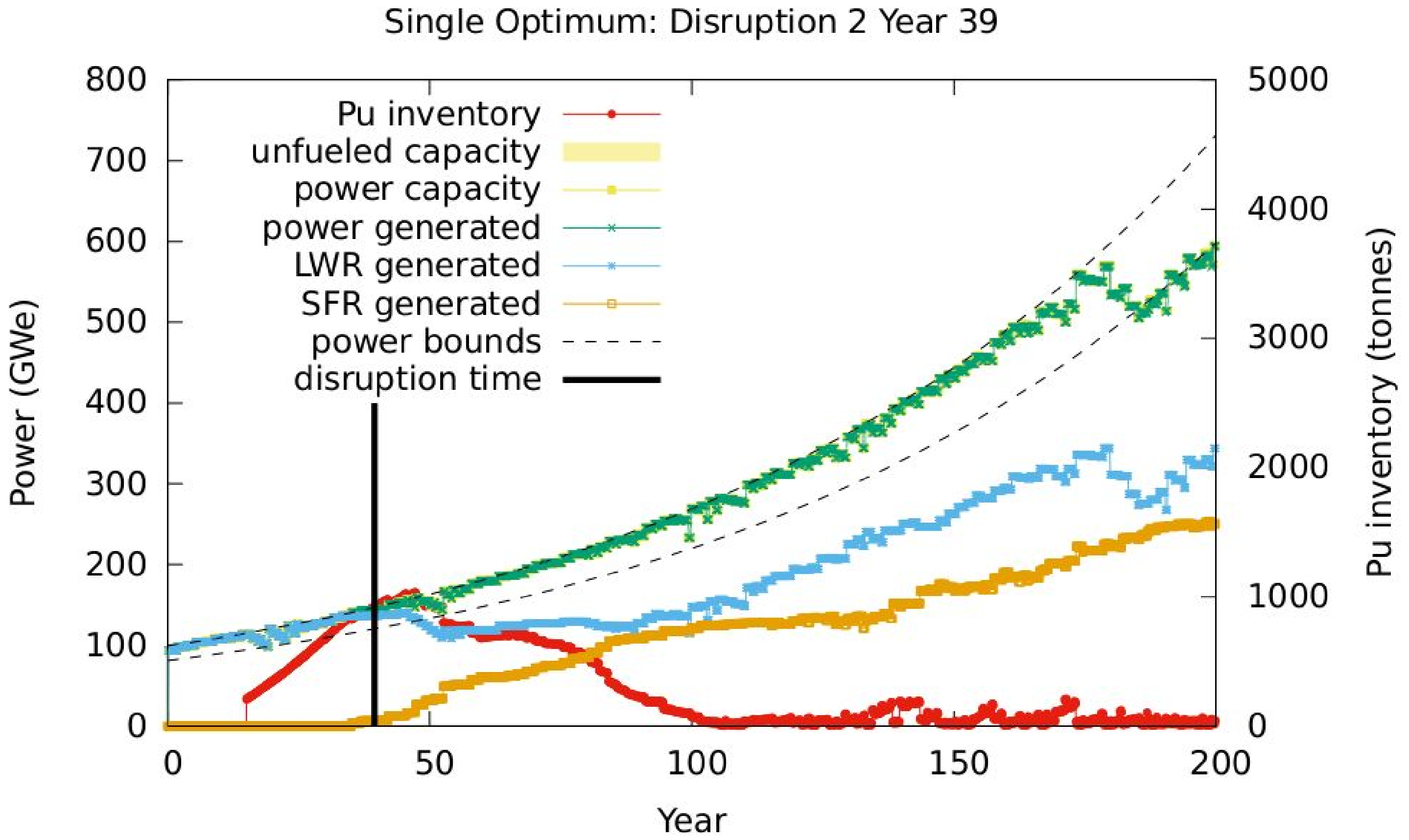
- $R^*(D,t_d) = D$ is a good reference (upper bound)
- t_d linear interpolation is okay, but not great
- In limit $t_d \rightarrow t_{end}$, approximations converge to actual

Disruption	S* Approximations		
Time (year)	Reference (Eq. 5.8)	t_d interp. (Eq. 5.9)	Actual
23	2.269	0.843	0.654
39	1.614	0.828	0.634
55	1.372	0.826	0.618
70	1.019	0.740	0.586
86	0.723	0.623	0.545
104	0.485	0.483	0.481
123	0.393	0.390	0.388
144	0.214	0.213	0.214
170	0.161	0.160	0.160
200	0.155	0.155	0.155

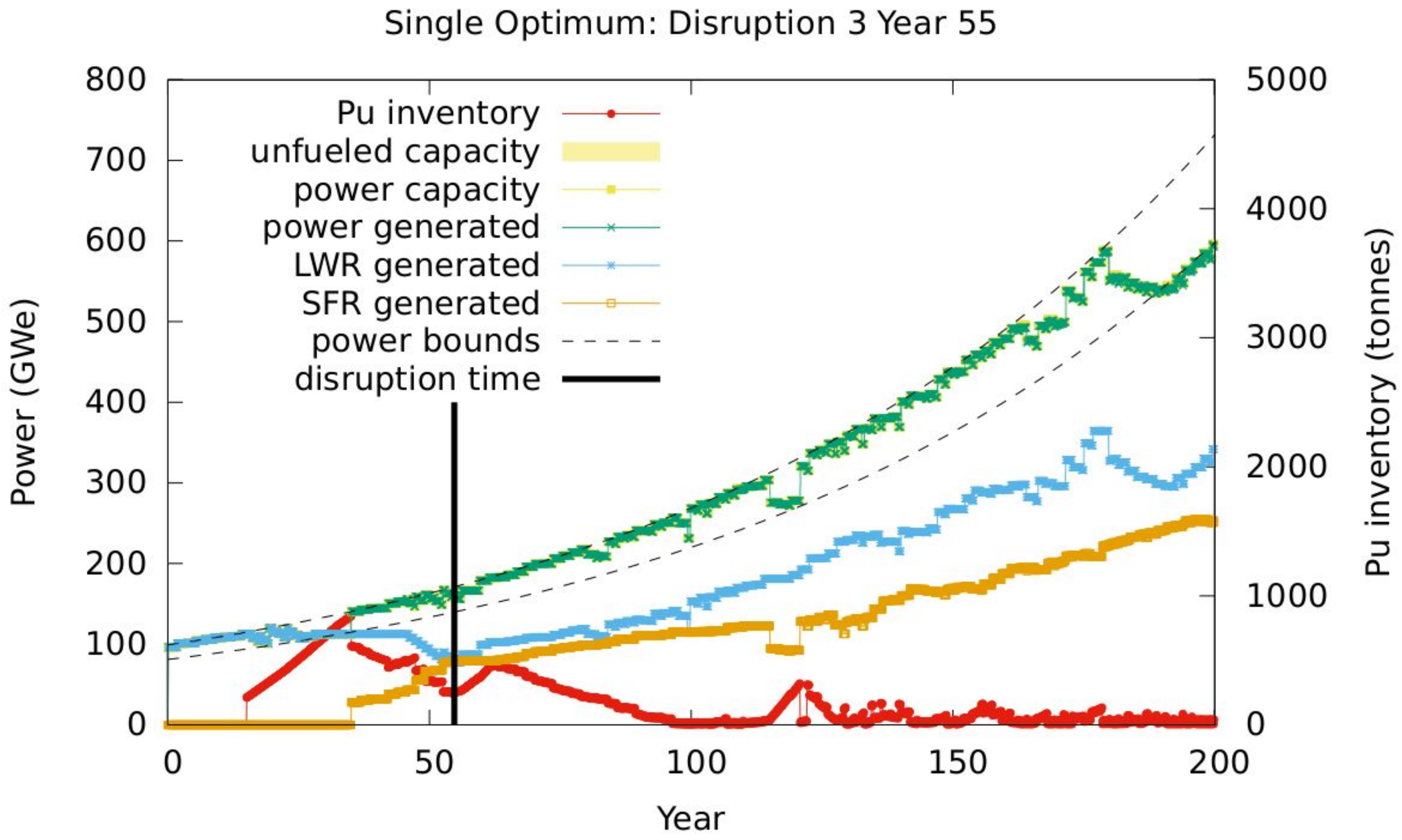
Disruption 1 Detail



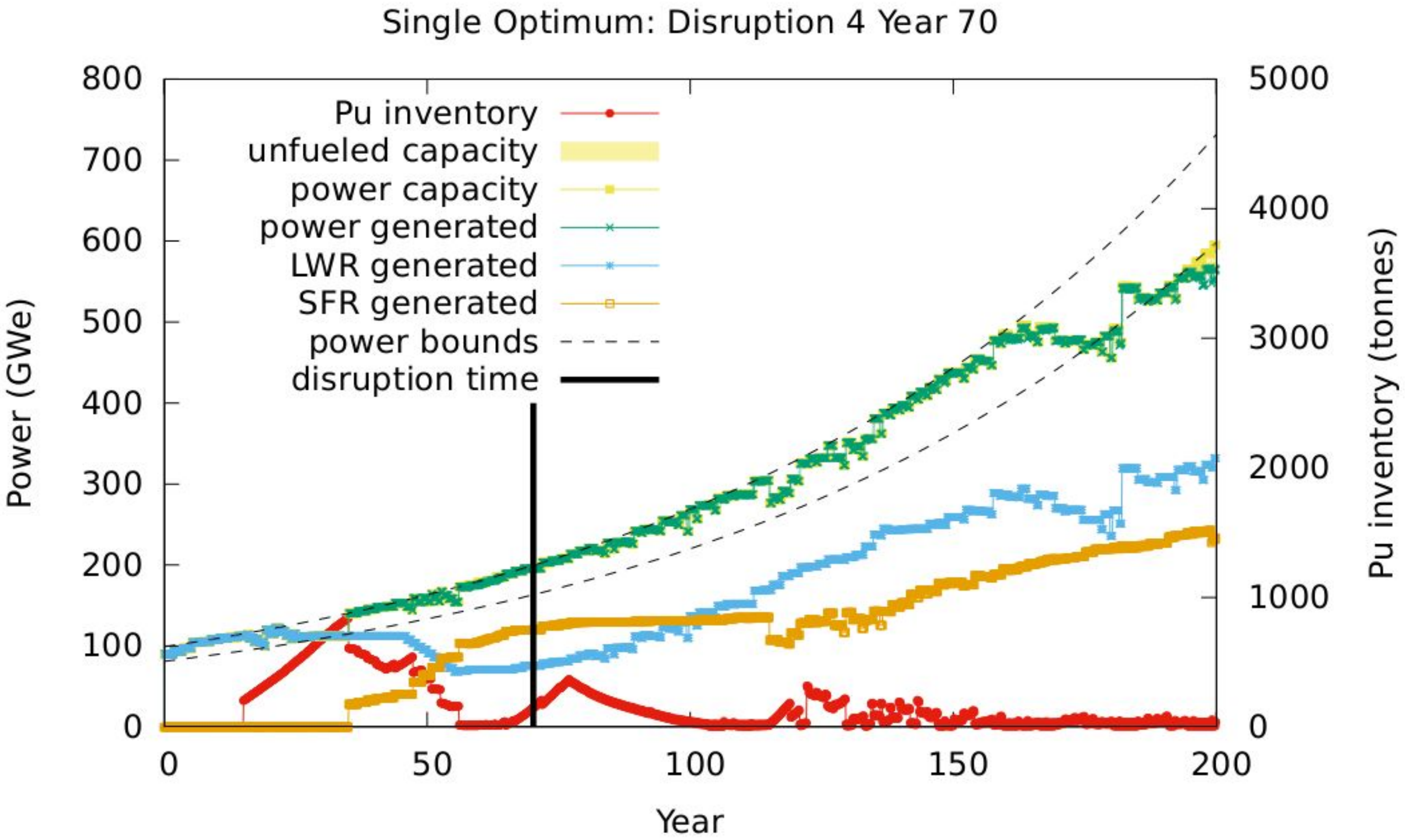
Disruption 2 Detail



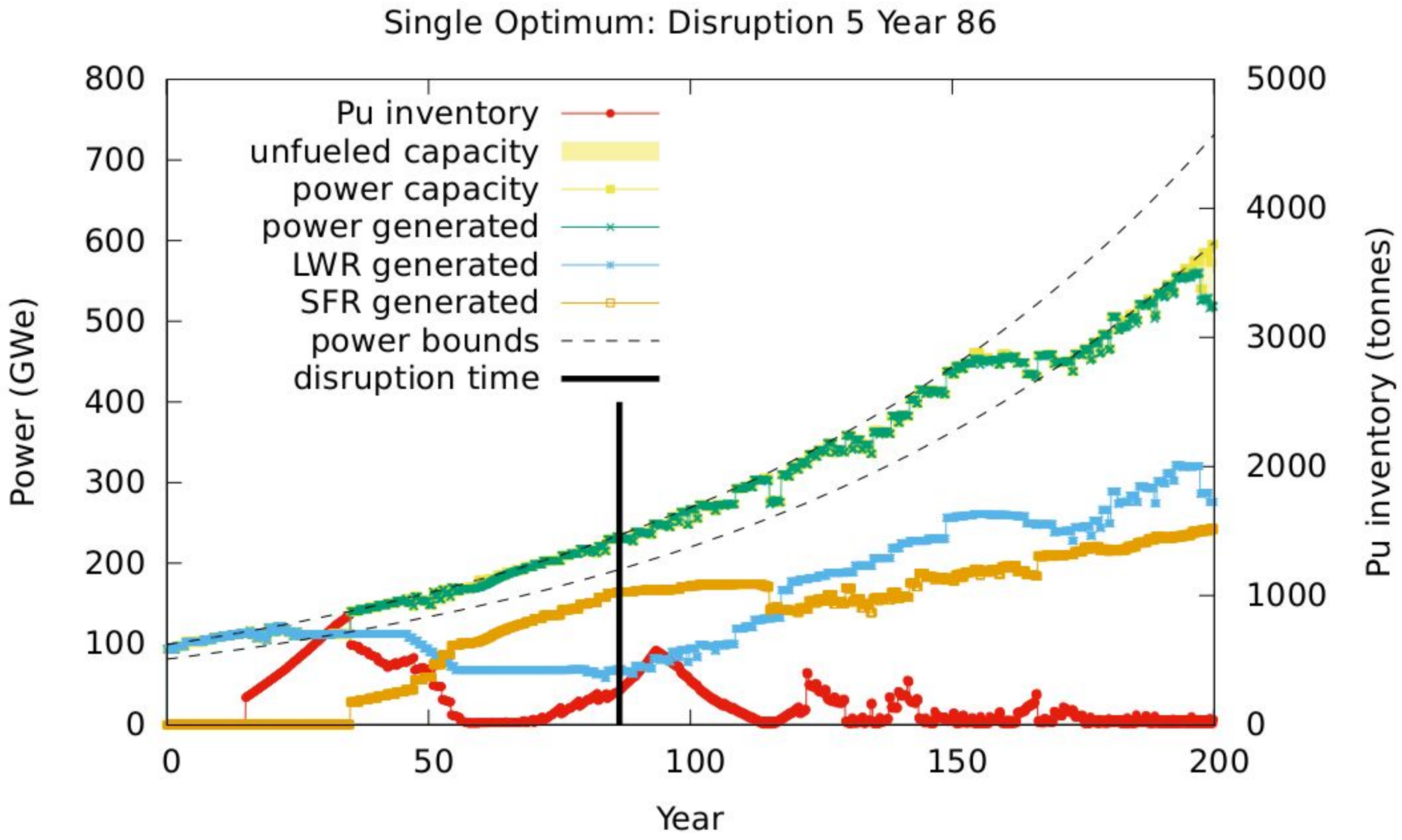
Disruption 3 Detail



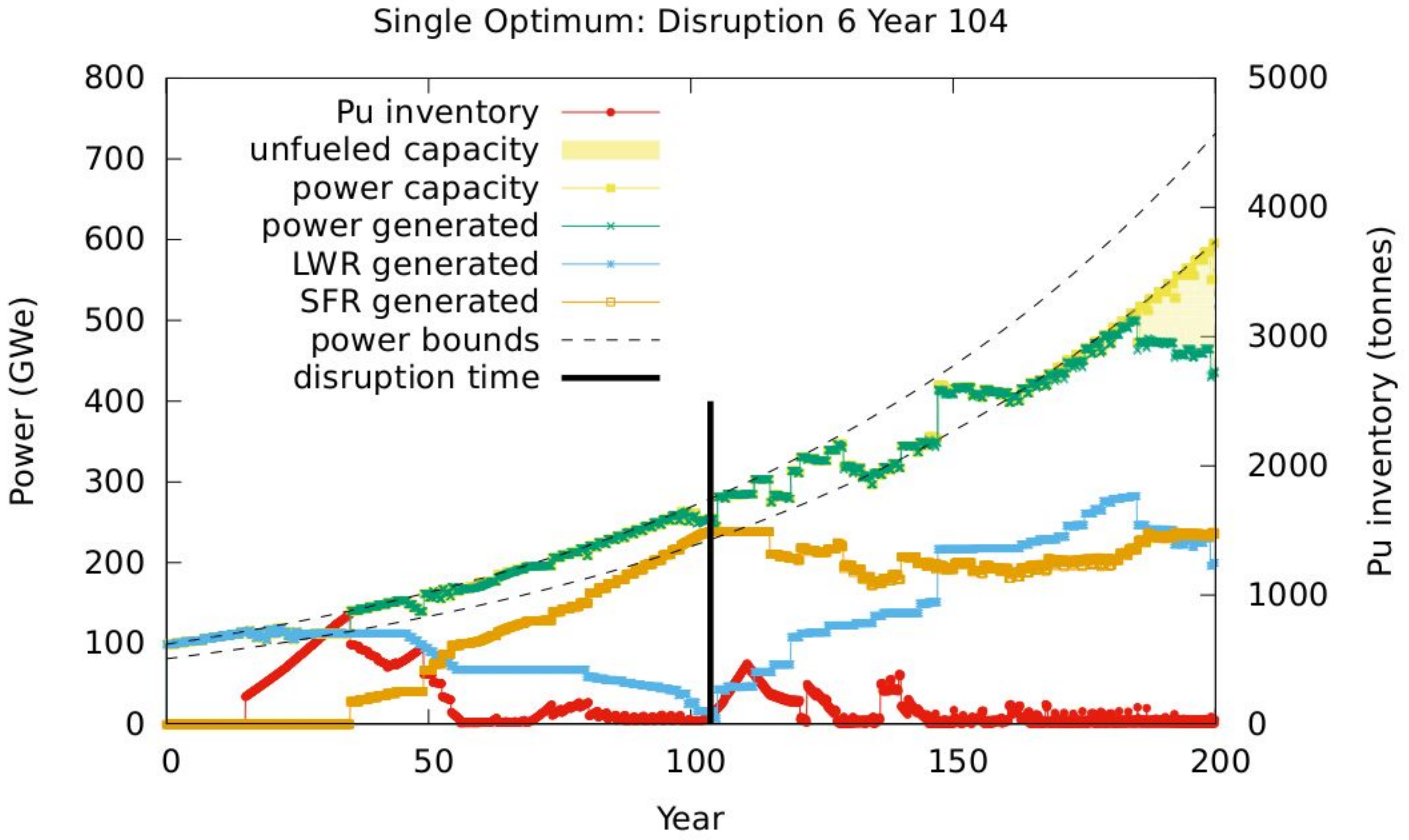
Disruption 4 Detail



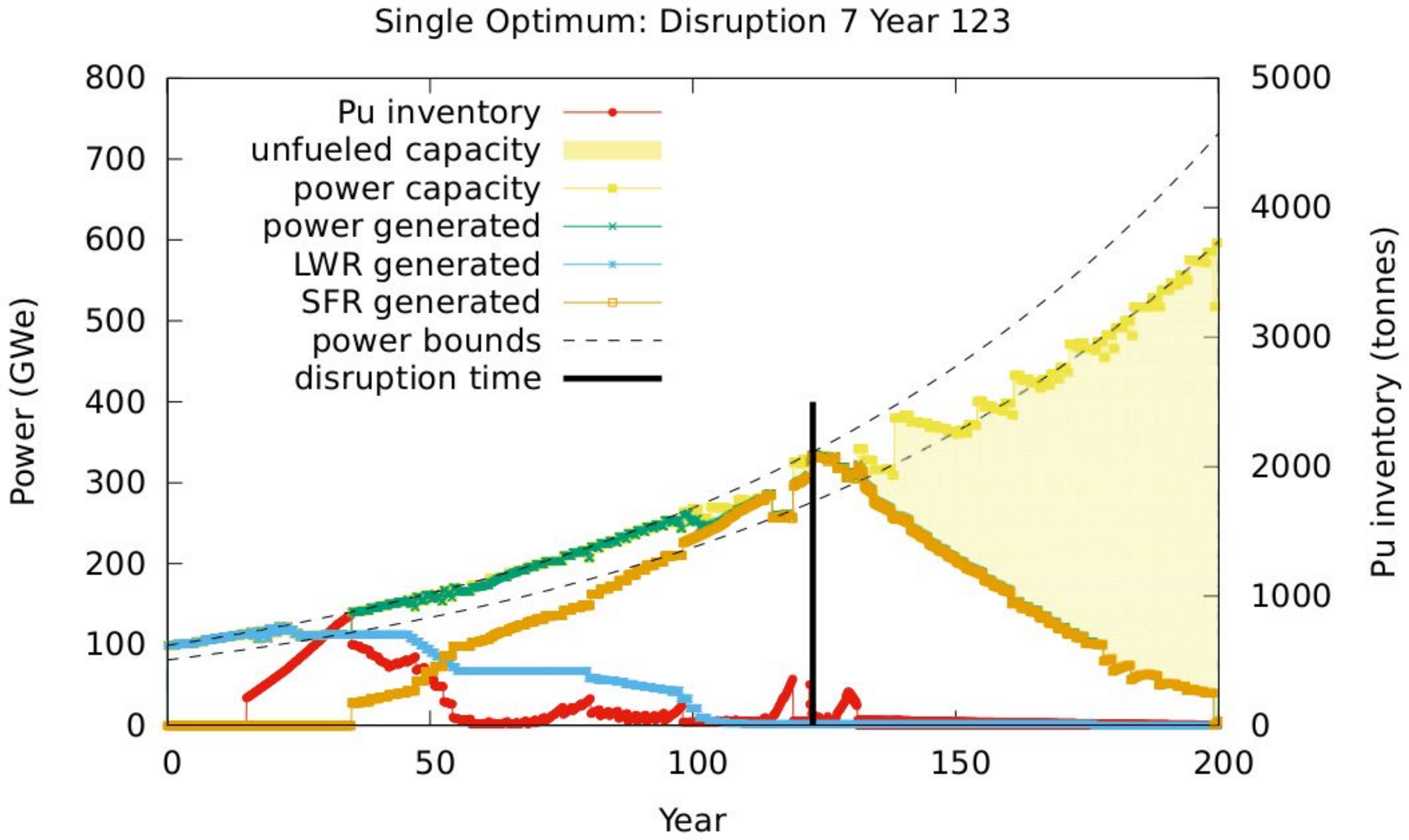
Disruption 5 Detail



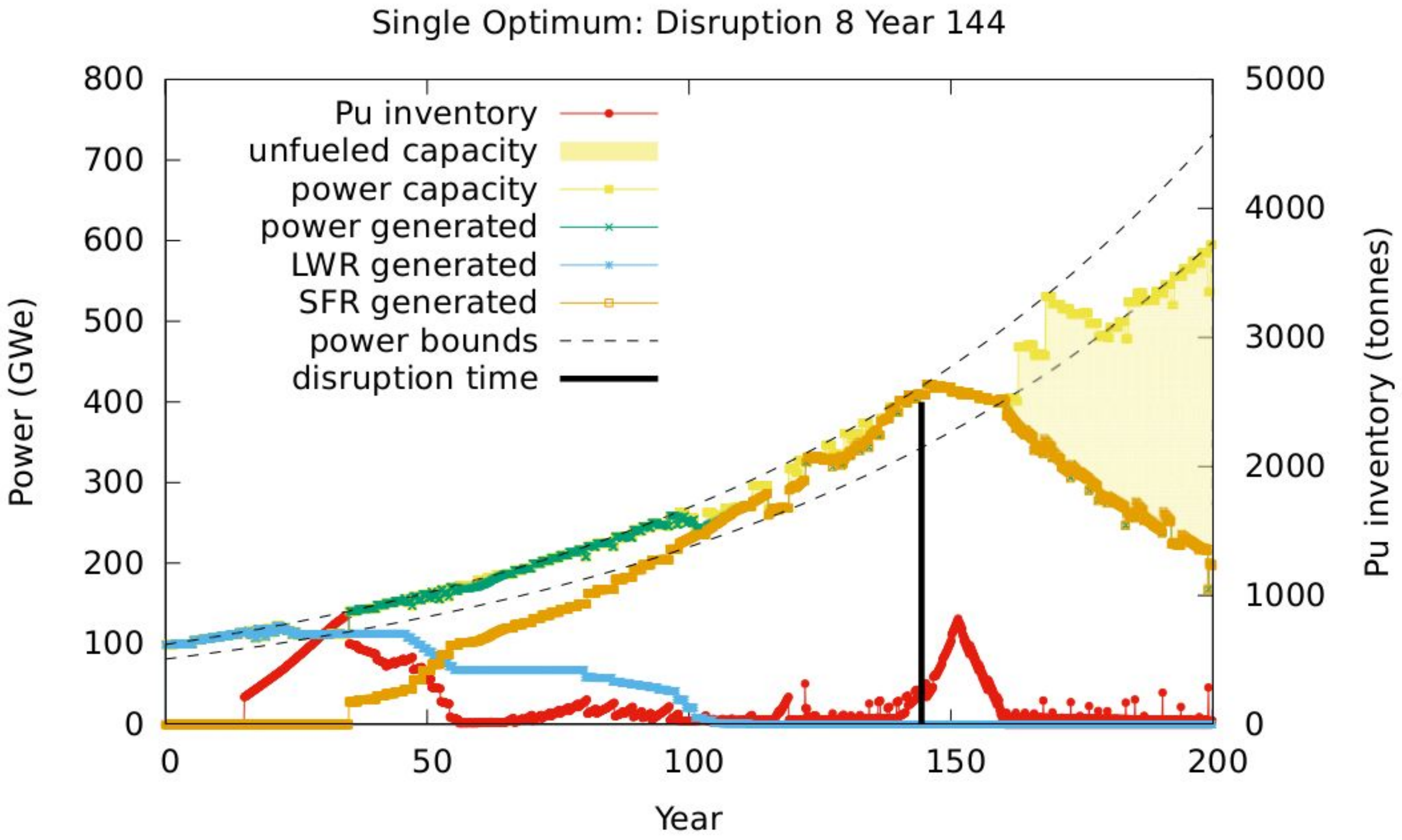
Disruption 6 Detail



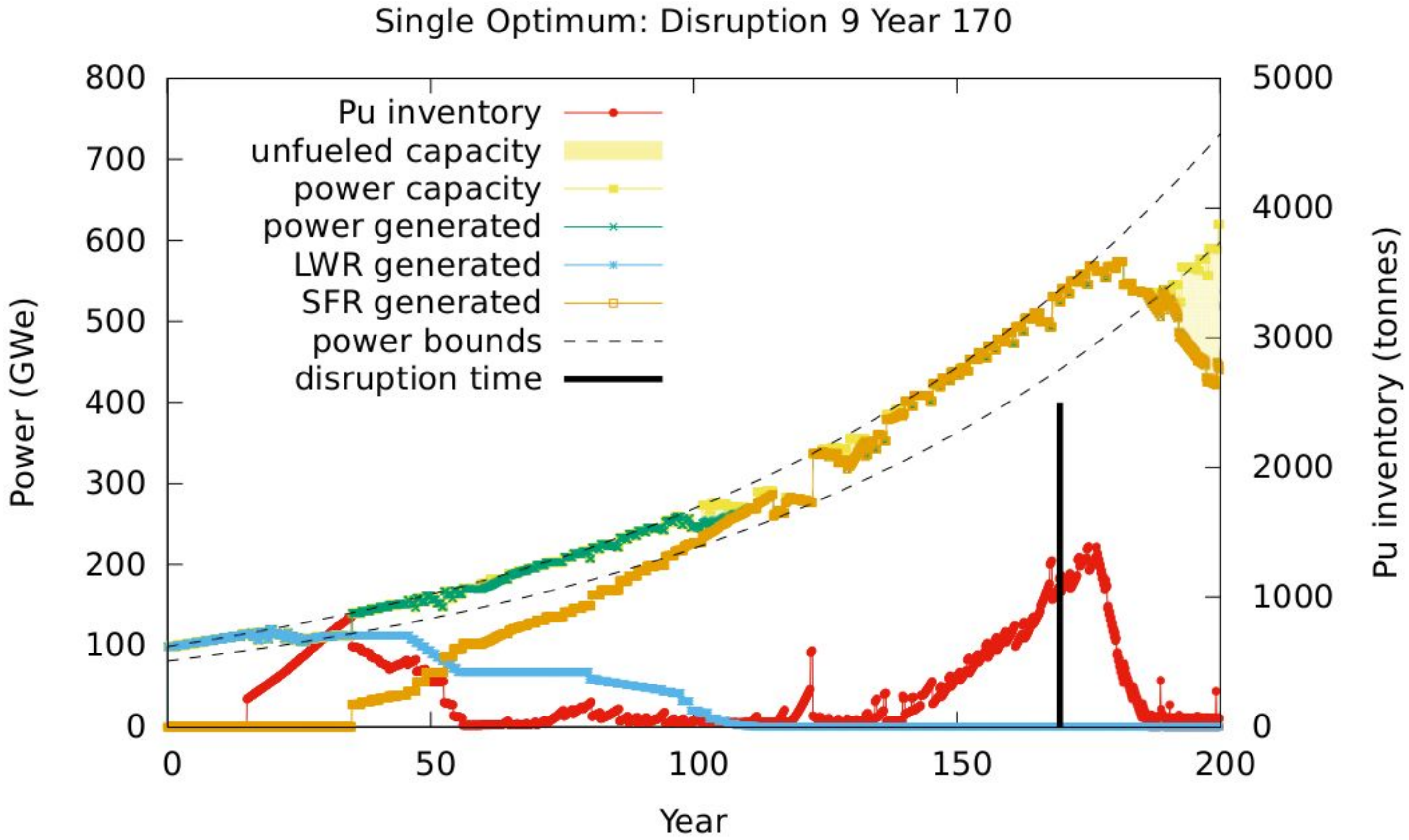
Disruption 7 Detail



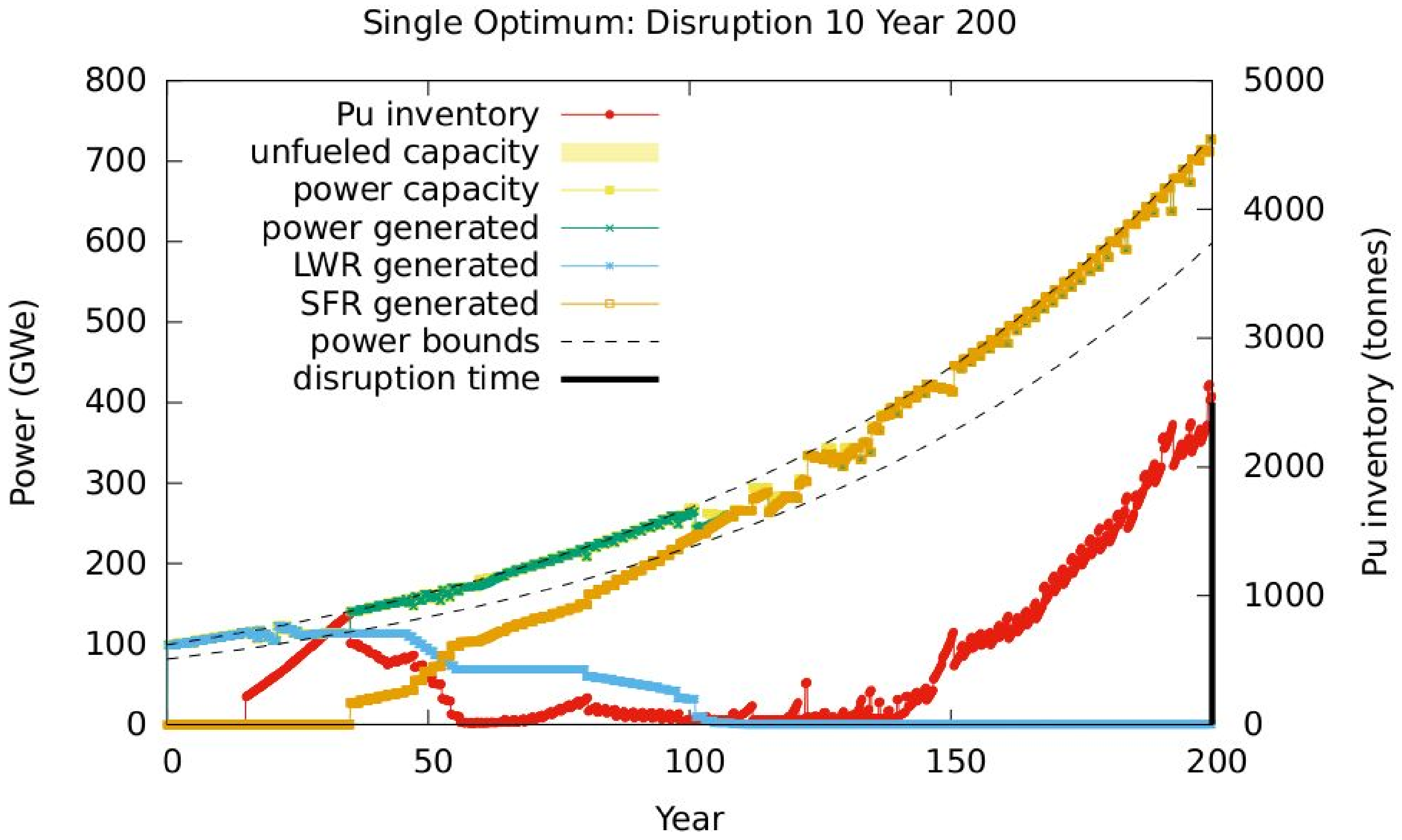
Disruption 8 Detail



Disruption 9 Detail



Disruption 10 Detail



Fast Reactor Age Distributions