Multifidelity and Adaptive Optimization Methods

Karen Willcox

Collaborative work with
Subodh Chaudhari, Jason Daniel, Andy Ko, Leo Ng

Phoenix Integration User Conference
April 15, 2015

Outline

- Learning algorithms to classify optimization problems
  - Which is the better optimizer for a particular problem?
  - How would you distinguish different optimization problems?

- Multifidelity optimization under uncertainty
  - A flexible framework to use different information sources to accelerate optimization under uncertainty

→ Towards goal-driven systematic management of multiple information sources and multiple solvers

OPTIMIZER CLASSIFICATION:

Which is the better optimizer for a particular problem?

Learning algorithms to classify optimization problems

- Offline classification
  - Maps problem types to optimization solution methods
  - Identify classification metrics (problem attributes) and scoring system
  - Build a database by running a range of test problems; use supervised and/or semi-supervised learning algorithms to build mapping
- Online assignment of an optimization method
  - Evaluate metrics and recommend an optimization method
  - Both a priori and adaptively as optimization proceeds
  - Will also support online optimization parameter tuning

Classification System

Captures the performance of the optimization algorithms for numerous test problems

Optimizer classification characterized by:

1. Number of function calls
2. Success rate
3. Convergence score
4. Feasibility score

Design space (optimization problem) classification characterized by

1. Dimension of the design space
2. Global smoothness of the design space

Convergence score

- Convergence score measures the overall convergence rate of one optimization run, using the arithmetic mean of the iteration-specific convergence progress:

$$\bar{C} = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{k}\frac{|f^0 - f^*|}{|f^k - f^*|}$$

- Accounts for distance to best known optimum and number of optimization iterations
- For constrained problems, we consider a separate feasibility score

Feasibility score

- Captures constraint violations for all iterations

$$\bar{V} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i=1}^{C}\max\left(\frac{c_i(x_k) - \alpha_i}{\overline{c_i}}, 0\right)$$

- Constraint functions c_i and tolerances α_i can be written for linear, nonlinear equality and inequality constraints

- $\overline{c_i}$ is a normalizing constant for constraint i

- Score increases penalty for violations in later optimization iterations

DESIGN SPACE CLASSIFICATION:

How would you distinguish different optimization problems?

How well is the design space represented by a polynomial response surface?

- For a model f, a third-order polynomial, p_3, is fitted to the training set computed with $N_{train} = 64$ DOE points.

- The polynomial is evaluated at $N_{test} = 64$ (different) test points x_i and compared to the actual value of the function. This defines a metric M:

$$M = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |p_3(\mathbf{x}_i) - f(\mathbf{x}_i)|$$

CLASSIFICATION EXAMPLE

Offline scoring applied to a database of test problems

- 11 functions from CEC14 tested in dimension 3 on $[-100,100]^3$
- Functions difficult to optimize.

1 - High conditioned elliptic function

2 - Bent Cigar function

3 - Discuss function

4 - Rosenbrock's funtion

5 - Ackley's function

6 - Weierstrass function

7 - Griewank's function

8 - Rastrigin's function

9 - Modified Schwefel's function

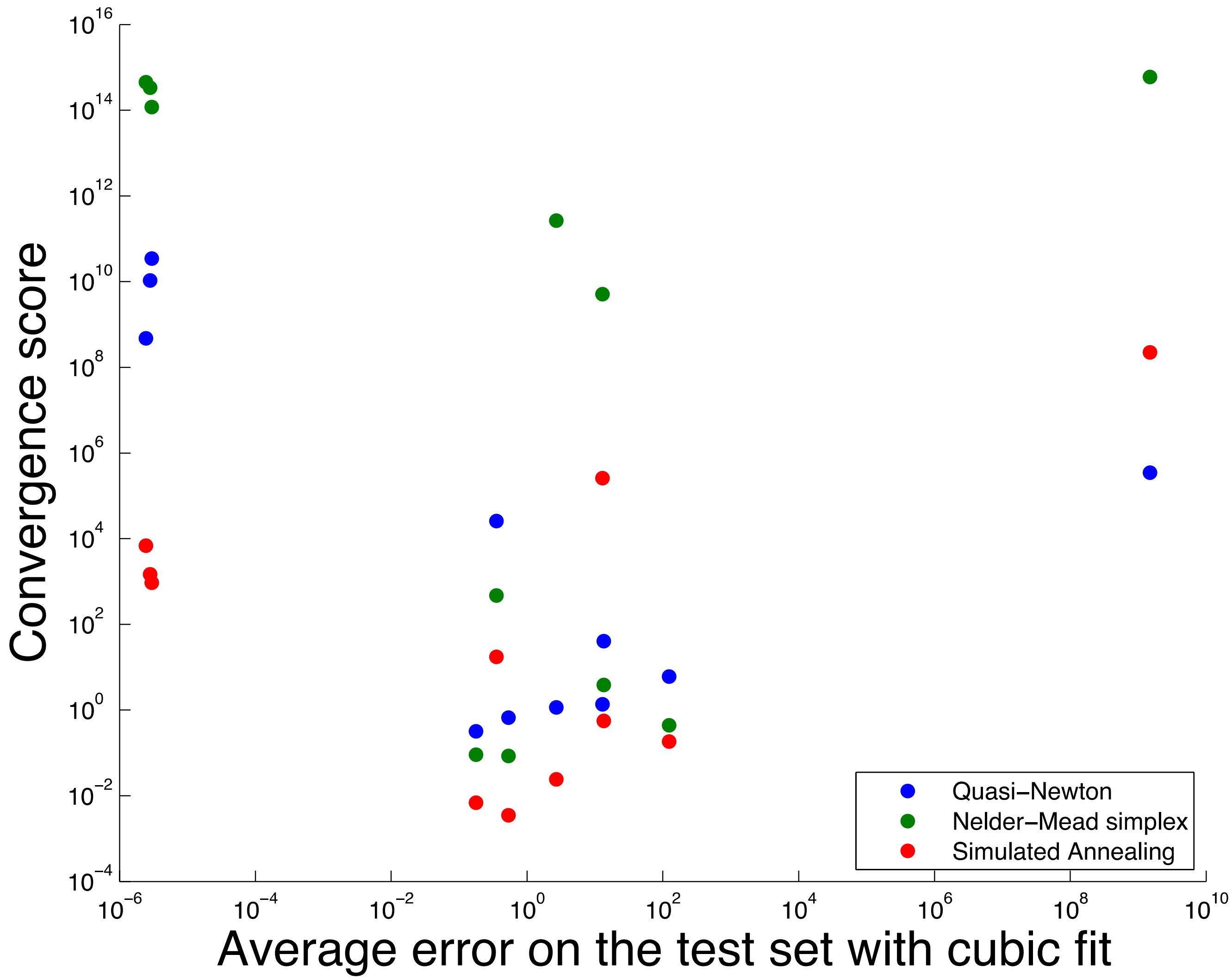
10 - Katsuura funtion

11 - HappyCat function

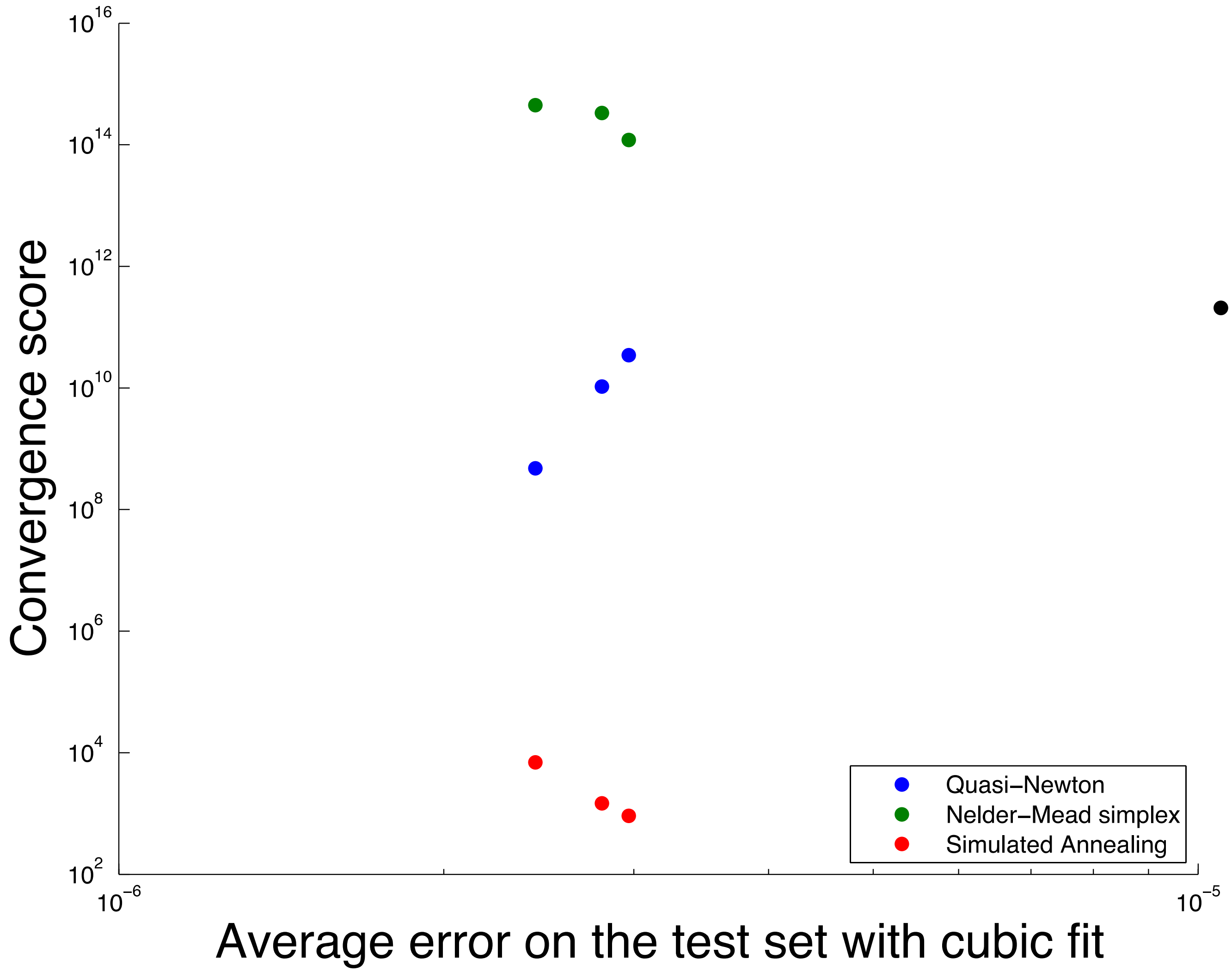
Scoring functions offline

- Each function is optimized using three algorithms:
  - Quasi-Newton
  - Nelder-Mead simplex
  - Simulated Annealing
- 64 initial guesses, selected by DOE
- Averaged convergence score computed for each function, plotted vs. the design space metric

Offline scoring
Convergence score vs. design space metric

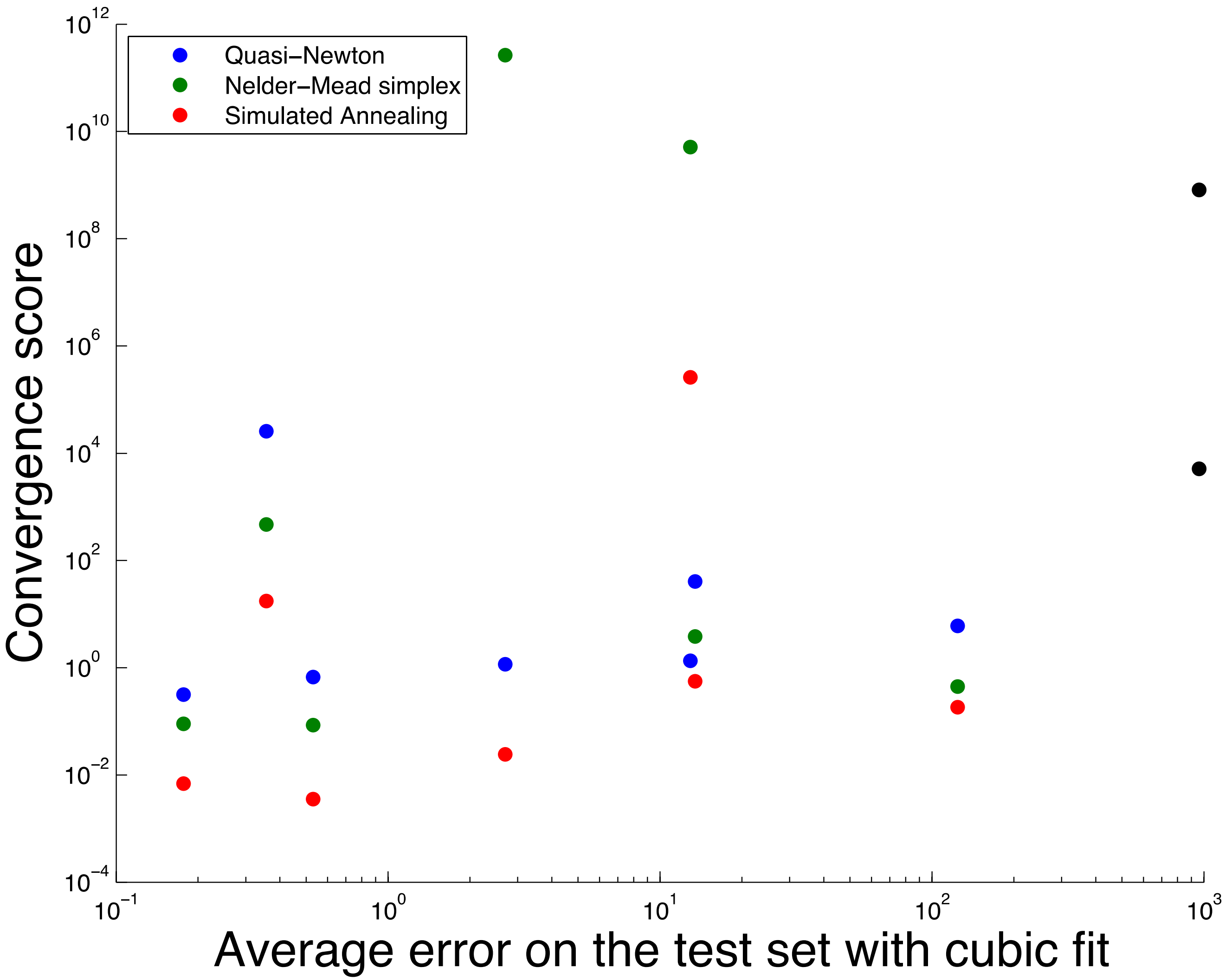


Offline scoring
Cases with low metric value



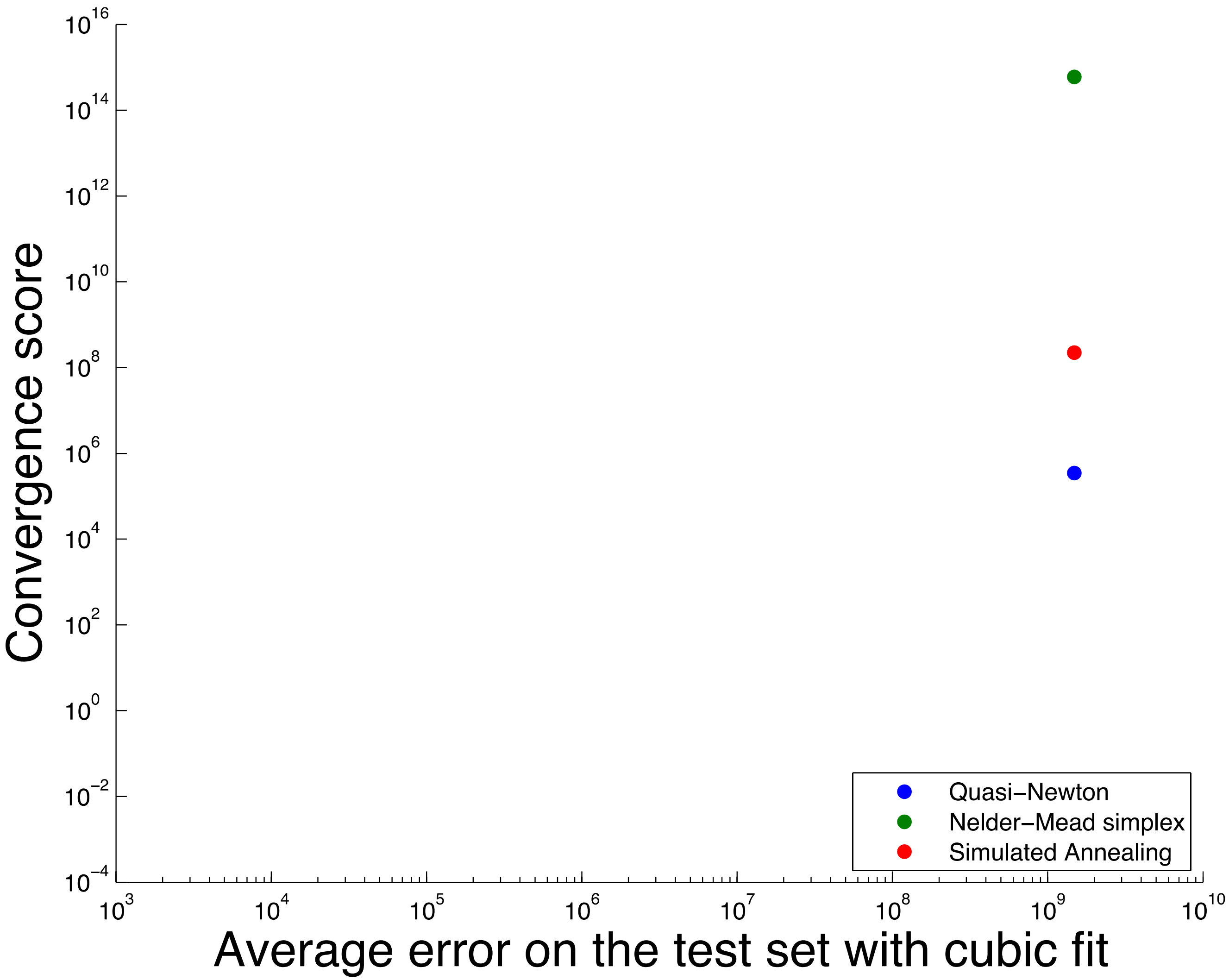
- Nelder-Mead performs well

Offline scoring
Cases with medium metric value



- Quasi-Newton performs well at lower end and higher end
- Nelder-Mead performs well in mid range

Offline scoring
Cases with high metric value



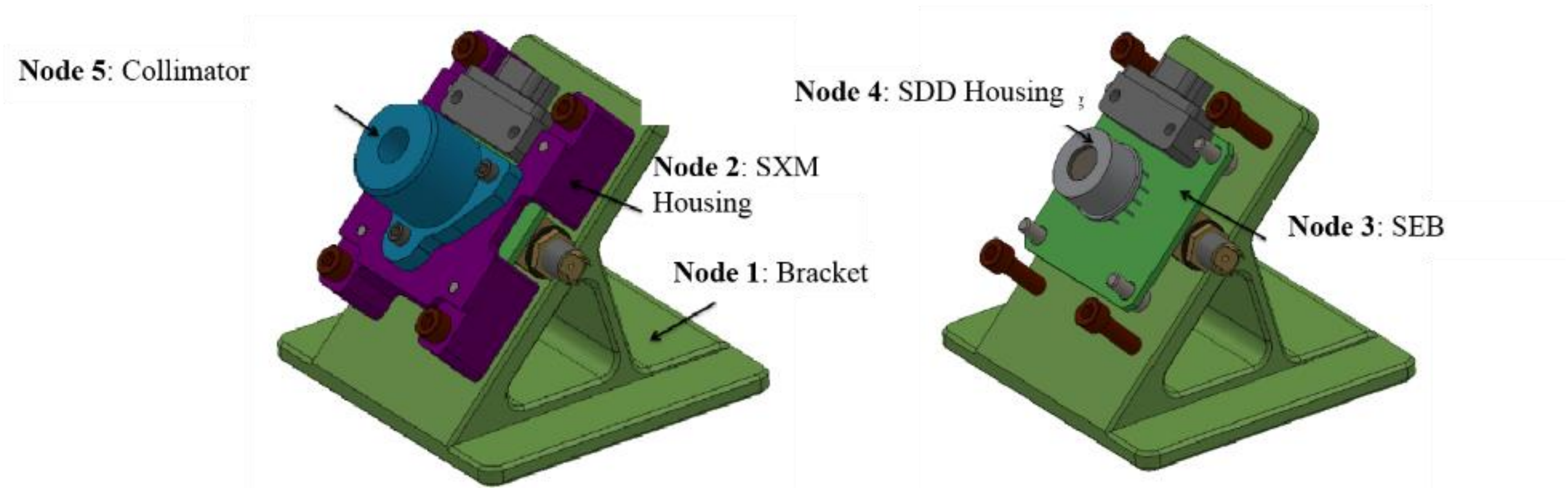
- Nelder-Mead performs well but only one test case

Offline scoring: Conclusions

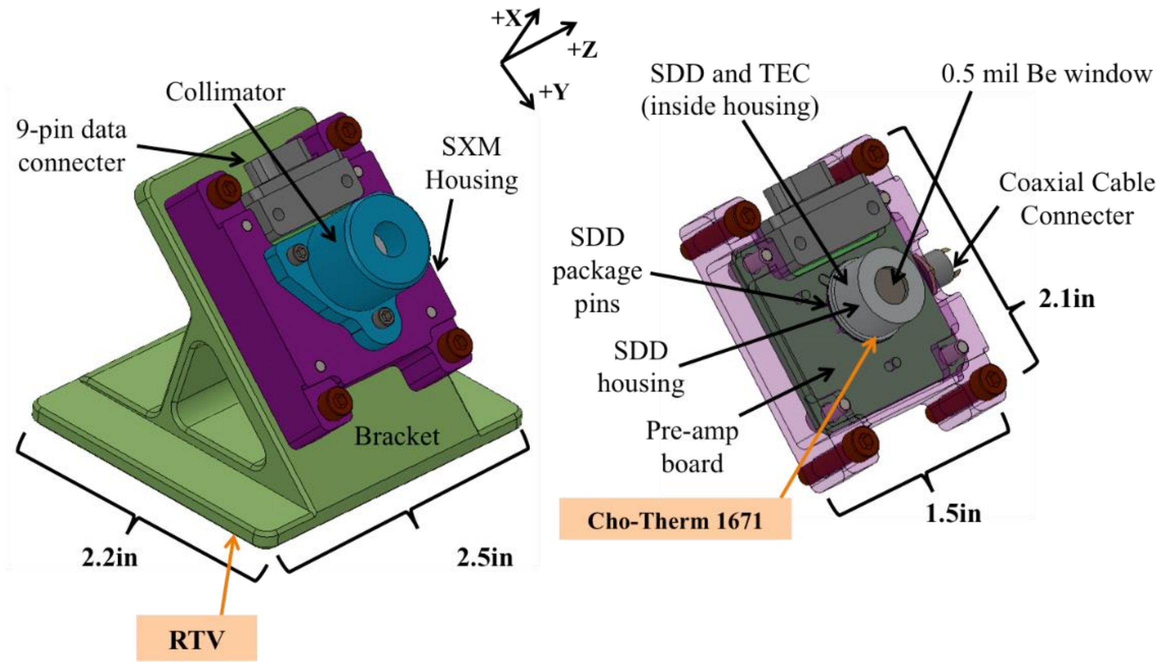
- The offline scoring for the unconstrained test problems suggests that:
  - Nelder-Mead performs well over a large range of the problems tested
  - There is an intermediate range of metric values where Quasi-Newton performs well
  - Local deterministic optimizers have consistently higher convergence scores than the global heuristic optimizer over these test problems

Thermal design demonstration problem

- REXIS Solar X-ray Monitor (SXM)
- Thermal model computes the temperature distribution and heat flux between the parts of the SXM instrument using a five-node lumped parameter thermal model [Stout, 2015]
- Model runtime: ~60 secs for thermal analysis over 7-day modeling period



SXM optimization problem setup

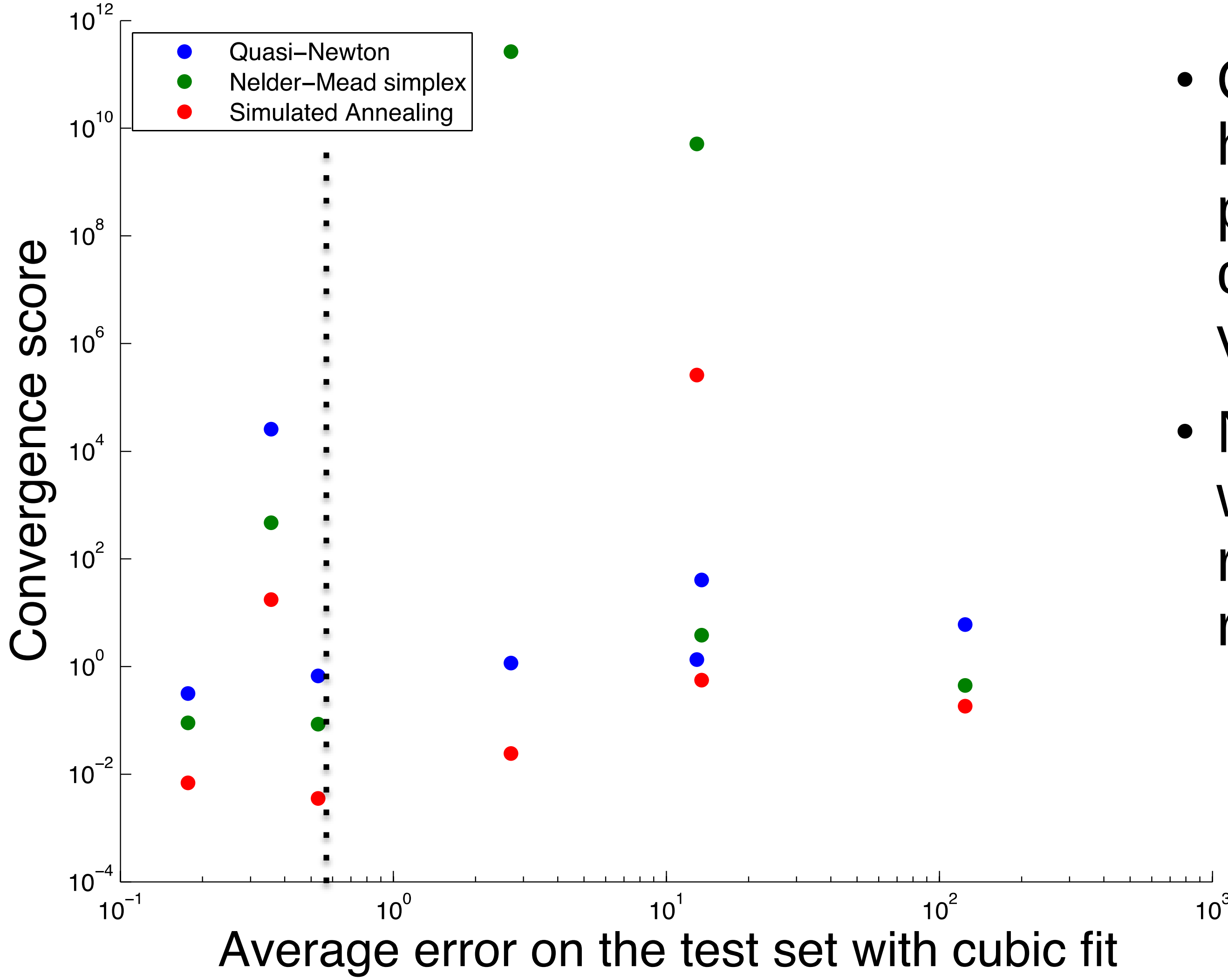


- Objective: Minimize thermal power transfer between SXM and spacecraft
- Constraint: Upper bound on maximum error between the target and actual SXM temperature
- Design variables: controller gain, controller frequency, voltage of thermal controller

SXM optimization problem

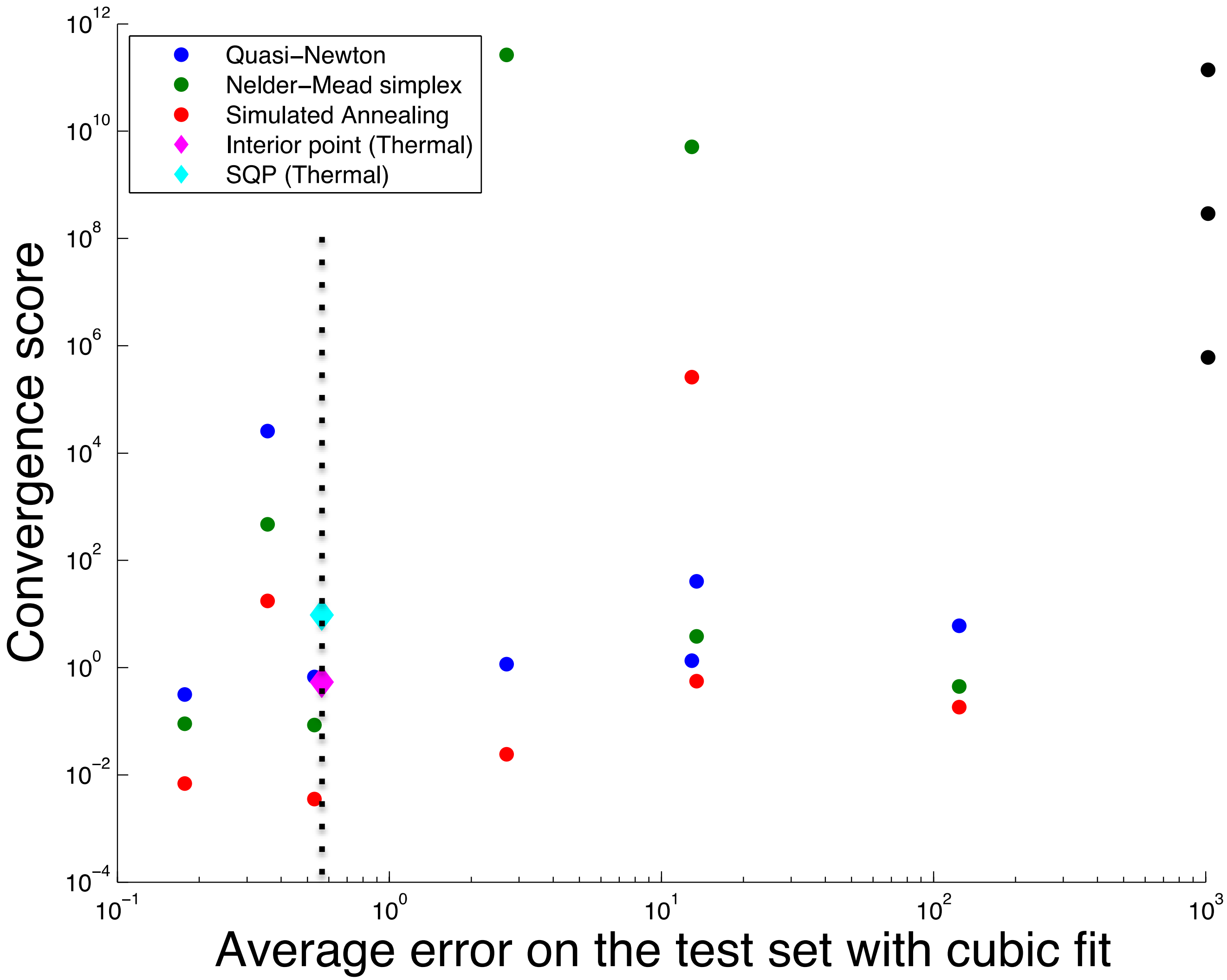
- The metric M is computed using 64 training points and 64 test points: for the SXM problem we find $M = 0.56$
- This informs an *a priori* recommendation to choose a local deterministic optimizer
- To check the recommendation, we optimize using GA, SQP, and interior point. In each case, the convergence score is computed and averaged over 10 initial guesses

Offline scoring recommends a local deterministic optimizer



- Quasi-Newton has best performance at closest metric value ($M = 0.56$)
- Nelder-Mead would also be a reasonable recommendation

Offline suggestion is confirmed by online score



- SQP does well on the SXM problem
- GA fails (not shown)
- Comparison limited by the fact that the offline database is based on unconstrained algorithms while the SXM problem is constrained

MULTIFIDELITY OPTIMIZATION UNDER UNCERTAINTY

Multifidelity modeling

Often have available several physical and/or numerical models that describe a system of interest.

- Models may stem from different resolutions, different assumptions, surrogates, approximate models, etc.

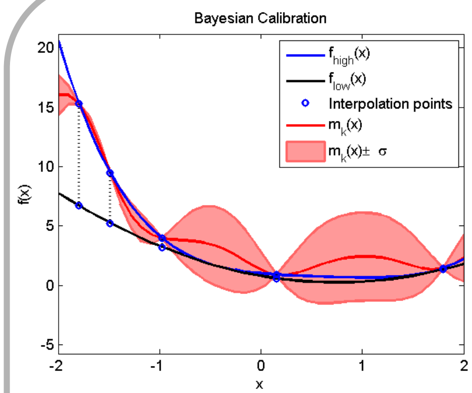
Multi-information source management:
How should we best use all available models and data in concert to achieve

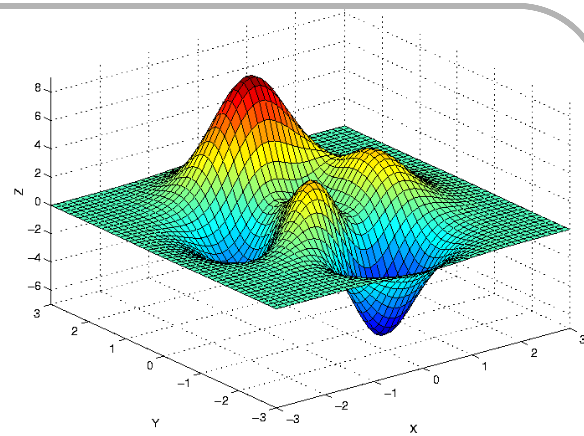
- Better decision-making (optimization, control, design, policy-making)
- Better understanding of modeling limitations
  → guidance for model development, experiments

Multi-information source management : Ingredients

- Multifidelity model construction
  - Building surrogate, hierarchical or competing models
    → exploiting structure
- Quantification of uncertainty and model fidelity
  - How good is a model *for a given purpose*
- Multifidelity model management
  - Which model to use when
  - Balancing computational cost with result quality
  - Convergence guarantees
  - Model-model and model-data fusion
  - Model adaptation

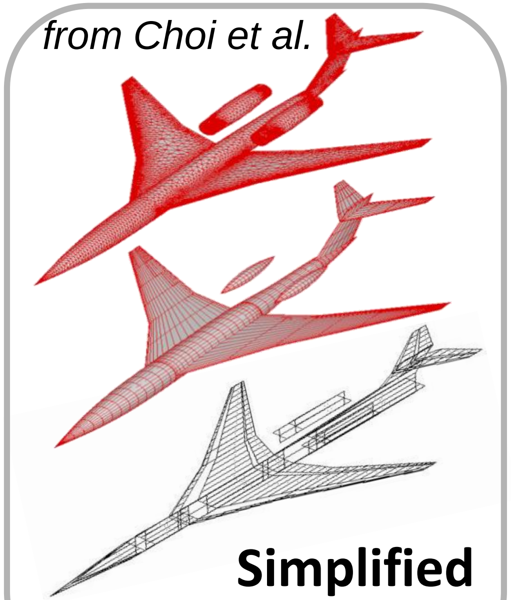
Multifidelity models



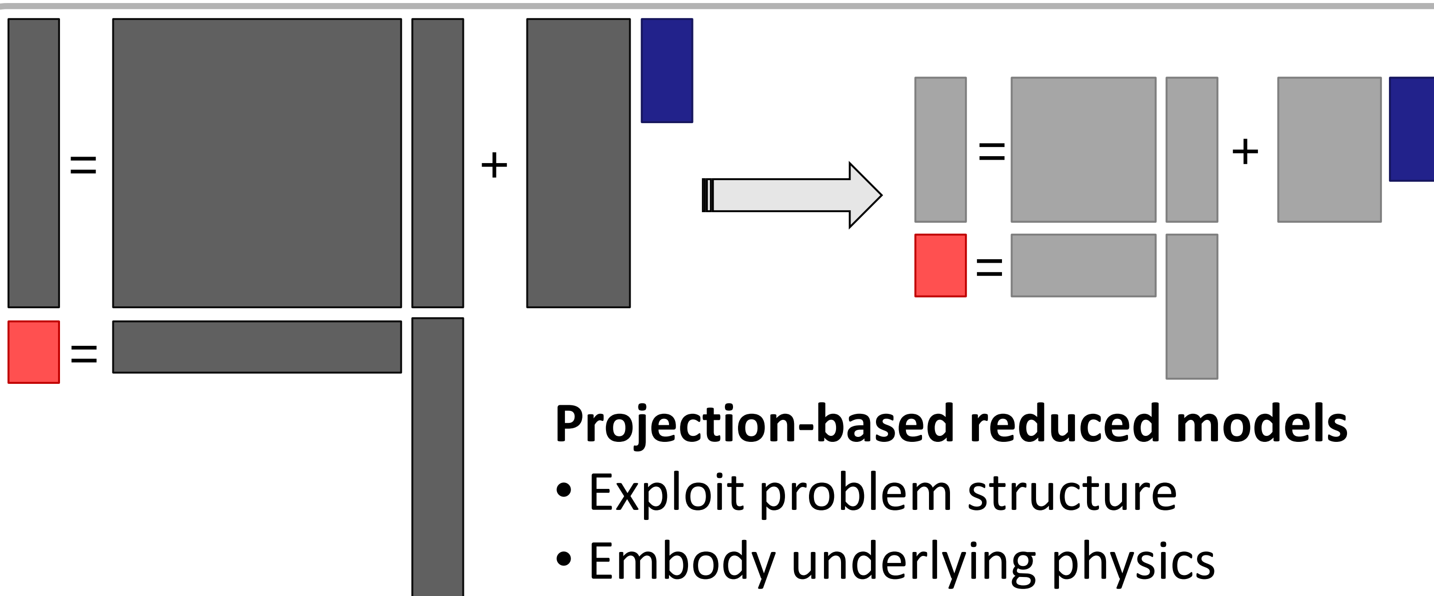


Data-fit models

from Choi et al.



Simplified physics models



Projection-based reduced models

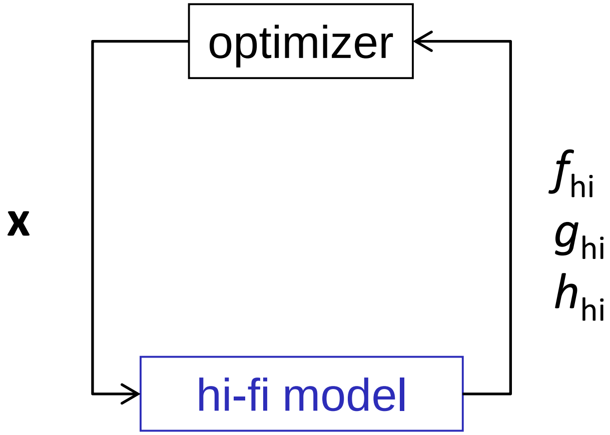
- Exploit problem structure
- Embody underlying physics

Multifidelity optimization

Example: Deterministic Design Optimization

$$\min_{x} f(x)$$
$$\text{s.t.} \quad g(x) \leq 0$$
$$h(x) = 0$$

Design variables x
Objective $f(x)$
Constraints $g(x)$, $h(x)$

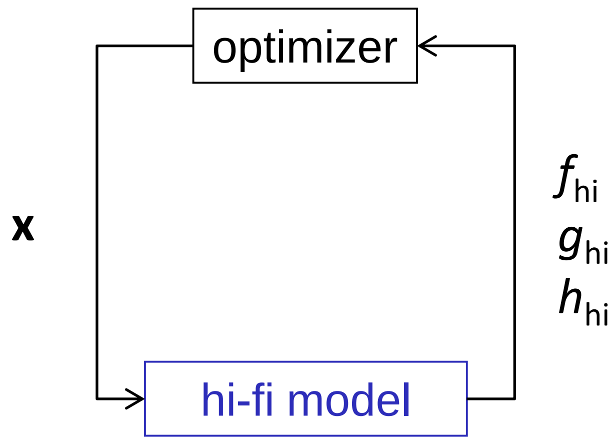


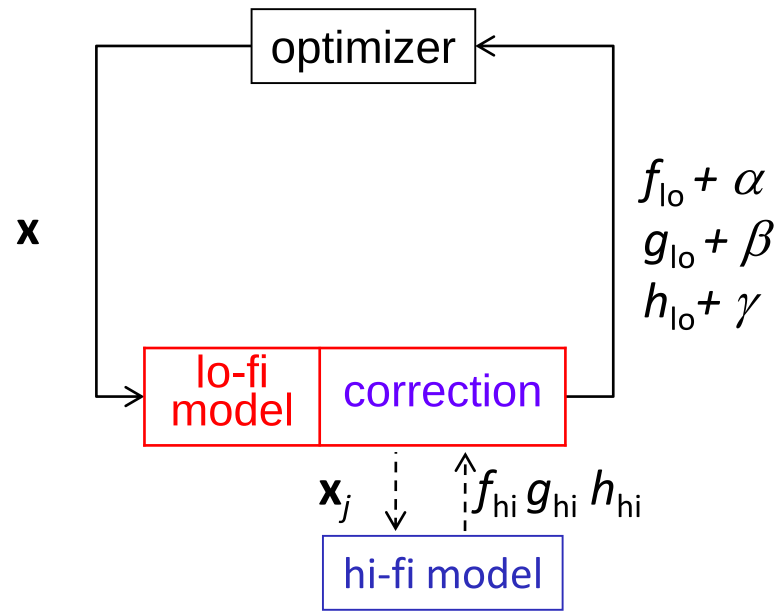
Multifidelity optimization: Use cheap models as much as possible; use adaptation of low-fidelity models

Example: Deterministic Design Optimization

$$
\begin{aligned}
&\min_x f(x) \\
&\text{s.t.} \quad g(x) \le 0 \\
&\qquad\quad h(x) = 0
\end{aligned}
$$

Design variables	x
Objective	$f(x)$
Constraints	$g(x)$, $h(x)$





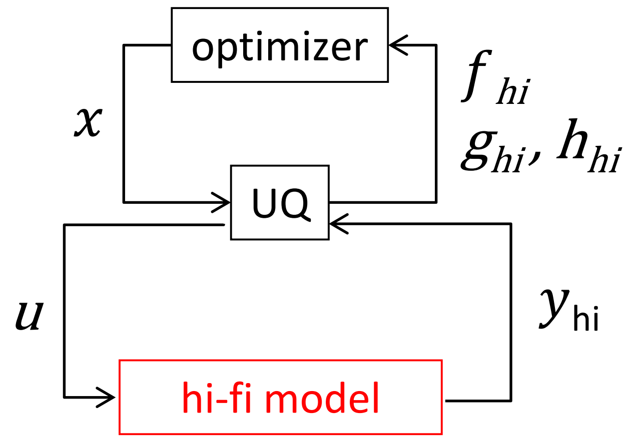
Adaptive corrections: Exploit model local accuracy

- Computed using occasional recourse to the high-fidelity model
- Constructed so that surrogate has desirable properties (e.g., for convergence)
- Managed using e.g. trust regions (Alexandrov and Lewis, 2001; Conn et al., 2009)

The challenge of optimization under uncertainty (OUU)

$$\min_{x} f\big(x, s(x)\big)$$
$$\text{s.t.} \quad g\big(x, s(x)\big) \leq 0$$
$$h\big(x, s(x)\big) = 0$$

Design variables	x
Uncertain parameters	u
Model outputs	$y(x,u)$
Statistics of model	$s(x)$



High-fidelity model embedded in a UQ loop within an optimization loop

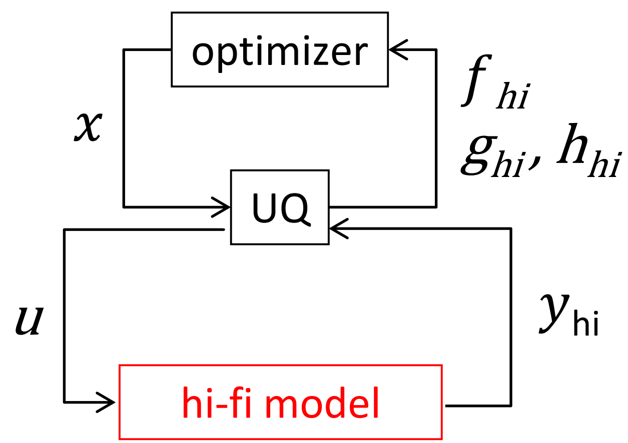
- Large computational cost
- Need an optimizer that is tolerant to noisy estimates of statistics

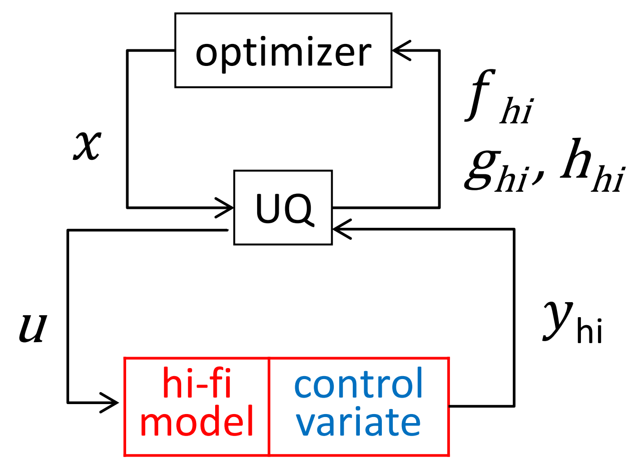
Multifidelity OUU approach: Control variates

Leo Ng
PhD 2013

$$\min_{x} f\big(x, s(x)\big)$$
$$\text{s.t.}\quad g\big(x, s(x)\big) \leq 0$$
$$h\big(x, s(x)\big) = 0$$

Design variables	x
Uncertain parameters	u
Model outputs	$y(x,u)$
Statistics of model	$s(x)$

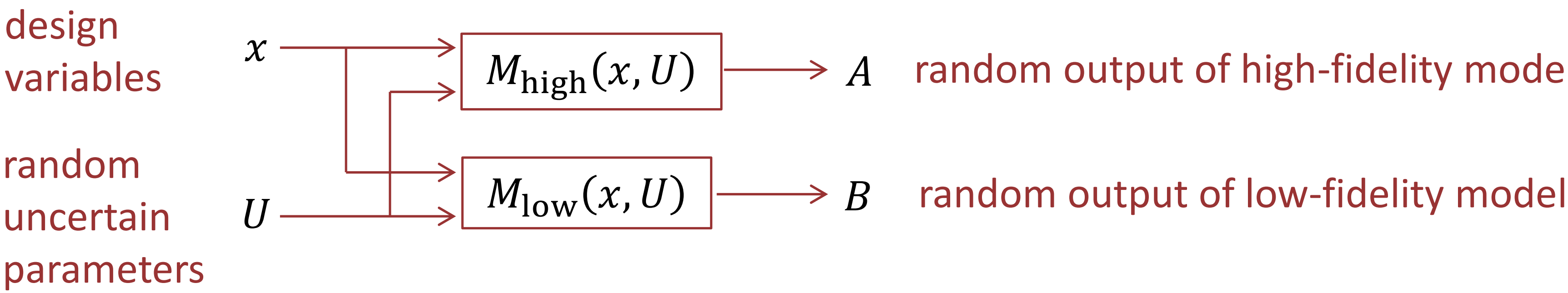




Control variates: Exploit model correlation

- Estimate correlation between high- and low-fidelity models

Problem setup



s_A = statistics of A (e.g., mean, variance)
$\hat{s}_A$ = estimator of s_A

$$\min_x f\big(x, s_A(x)\big) \quad \text{s.t.} \quad g\big(x, s_A(x)\big) \le 0$$

approximated by

$$\min_x f(x, \hat{s}_A) \quad \text{s.t.} \quad g\big(x, \hat{s}_A(x)\big) \le 0$$

Variance reduction with control variate

- Regular MC estimator for $s_A = \mathbb{E}[A]$ using n samples of A:

$$\bar{a}_n = \frac{1}{n}\sum_{i=1}^{n} a_i \qquad \mathrm{Var}[\bar{a}_n] = \frac{\sigma_A^2}{n}$$

Definitions:

$$\sigma_A^2 = \mathrm{Var}[A]$$

$$\sigma_B^2 = \mathrm{Var}[B]$$

$$\rho_{AB} = \mathrm{Corr}[A, B]$$

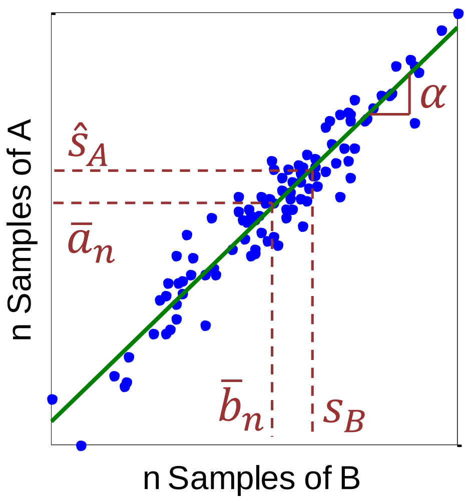
- Control variate (CV) estimator of s_A:
  - Additional random variable B with known $s_B = \mathbb{E}[B]$

$$\hat{s}_A = \bar{a}_n + \alpha\left(s_B - \bar{b}_n\right)$$

$$\mathrm{Var}[\hat{s}_A] = \frac{\sigma_A^2 + \alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B}{n}$$

- Minimize $\mathrm{Var}[\hat{s}_A]$ with respect to α

$$\mathrm{Var}[\hat{s}_A^*] = \underbrace{(1 - \rho_{AB}^2)}_{\leq 1}\frac{\sigma_A^2}{n}$$



Low-fidelity model as control variate

- Multifidelity estimator of s_A based on control variate method:
  - A = random output of high-fidelity model
  - B = random output of low-fidelity model (s_B unknown)

Definitions:

$$\sigma_A^2 = \mathrm{Var}[A]$$

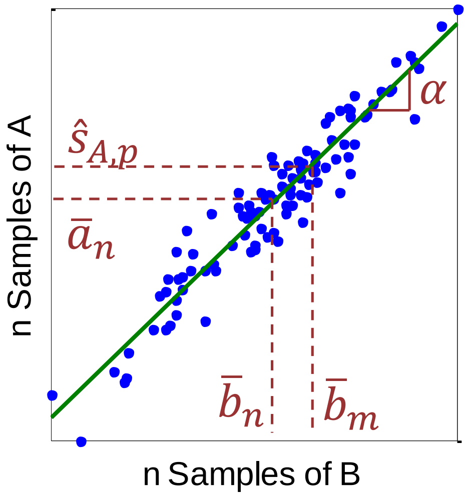
$$\sigma_B^2 = \mathrm{Var}[B]$$

$$\rho_{AB} = \mathrm{Corr}[A,B]$$

$$\hat{s}_{A,p} = \bar{a}_n + \alpha\left(\bar{b}_m - \bar{b}_n\right) \quad \text{with} \quad m \gg n$$

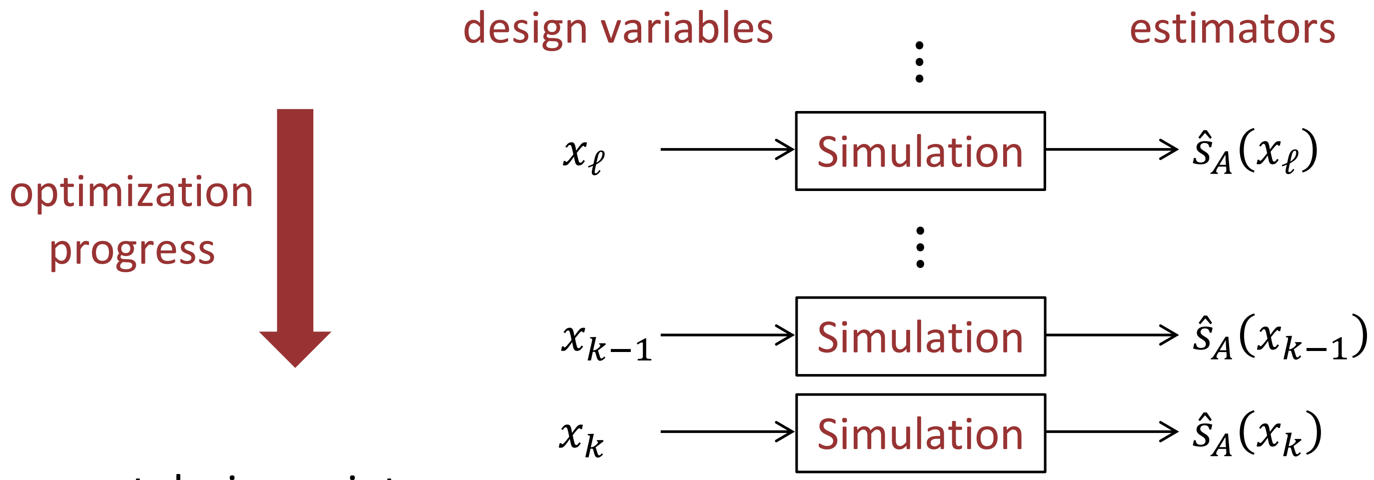
$$\mathrm{Var}\left[\hat{s}_{A,p}\right] = \frac{\sigma_A^2 + \alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B}{n} - \frac{\alpha^2\sigma_B^2 - 2\alpha\rho_{AB}\sigma_A\sigma_B}{m}$$

- Using difference $\left(\bar{b}_m - \bar{b}_n\right)$ as correction to $\bar{a}_n$
- Leveraging correlation between A and B
  - Correlation captured in α



Model correlation over design space

- What if low-fidelity model unavailable?
  - Use $M_{\text{high}}(x + \Delta x, U)$ as **surrogate** for $M_{\text{high}}(x, U)$



- At current design point x_k
  - Define $A = M_{\text{high}}(x_k, U)$
  - Want to compute $\hat{s}_A$ as estimator of $s_A = \mathbb{E}[A]$

Information Reuse Estimator

- Previously visited design point x_ℓ where $\ell < k$
  - Define surrogate as $C = M_{\text{high}}(x_\ell, U)$
  - Reuse available data: $\hat{s}_C$ as estimator of $s_C = \mathbb{E}[C]$ with error $\text{Var}[\hat{s}_C]$

Aircraft conceptual design under uncertainty

- Conceptual design of fuel efficient and quiet aircraft with 2030-2035 technologies
- Model = TASOPT aircraft sizing and mission performance analysis code (*Drela 2010*)
  - Includes aerodynamics, structures, weights, propulsion, stability, control, trajectory simulation

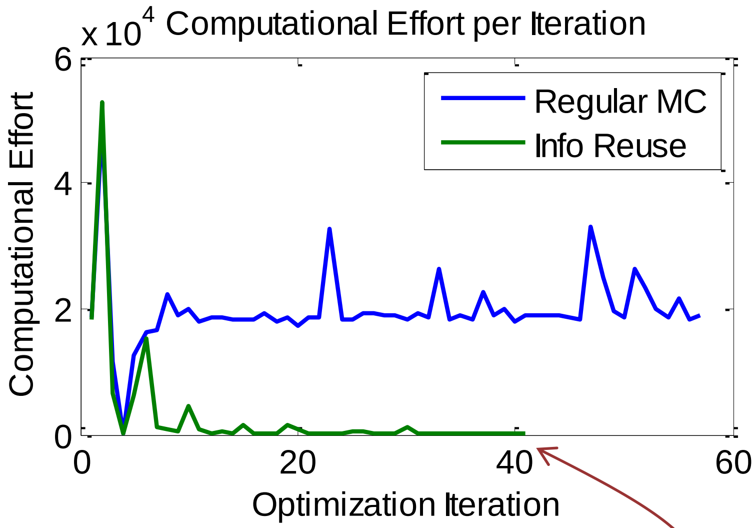
D8 aircraft concept
(*Greitzer et al. 2010*)

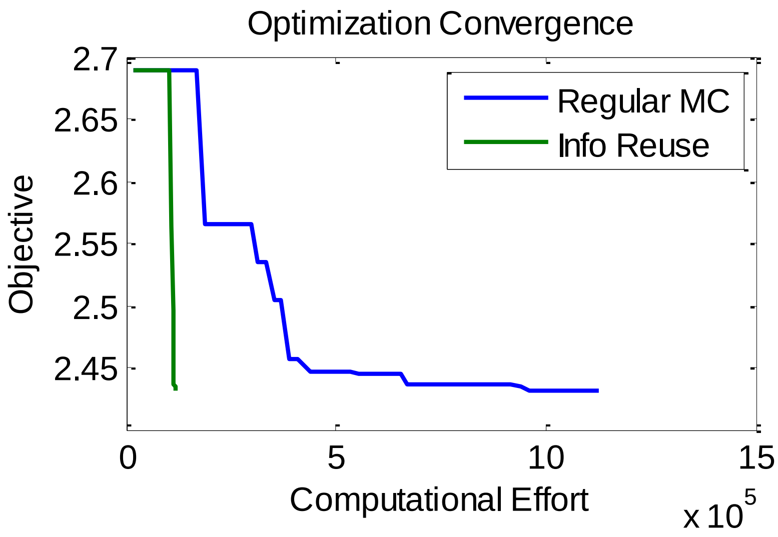
Aircraft conceptual design under uncertainty

- 8 design variables:
  - Wing geometry (aspect ratio, sweep, thicknesses), cruise lift coefficient, cruise lift distribution fractions, begin cruise altitude
- 19 random inputs representing uncertainties in technologies:
  - Material properties, boundary layer ingestion, secondary weights, engine cycle, etc.
- Objective (formulated as mean)
  - Payload fuel energy intensity (PFEI) [kJ/(kg km)]
- 4 constraints (formulated as mean + std ≤ 0)
  - Field length, fuel volume, span length, top-of-climb angle

- Optimization loop: COBYLA constrained derivative-free solver (*Powell 1994*)
- Simulation loop: Fixed RMSE for estimators specified, number of samples allowed to vary

Aircraft conceptual design under uncertainty



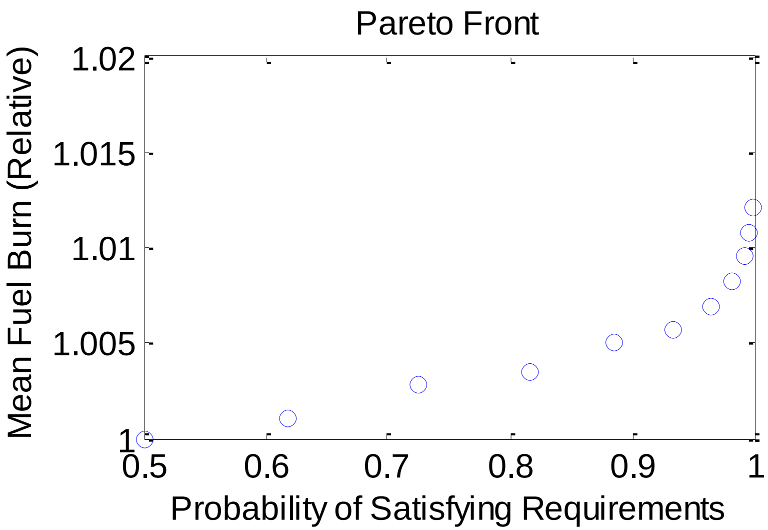


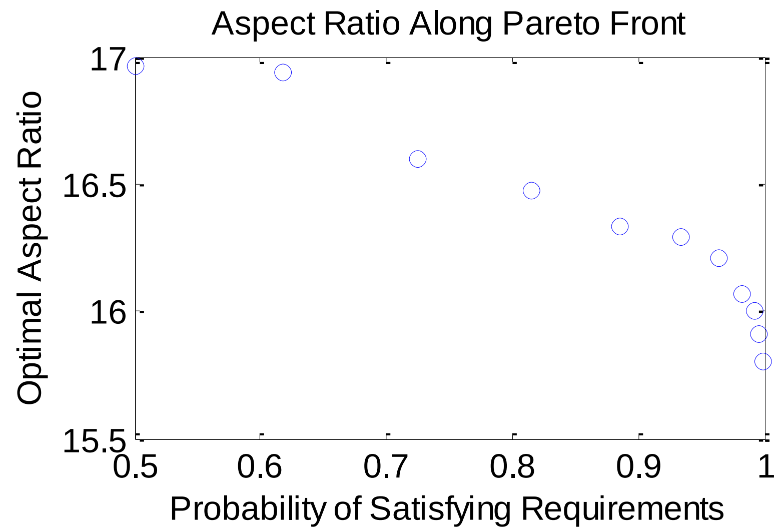
Variance reduction means significantly fewer samples for desired RMSE

Maintained at least 32 samples (computational effort = 64) to estimate ρ_{AC} and γ

	Total Computational Effort
Regular MC	1.1×10^6
Info Reuse	0.12×10^6 (-90%)

Aircraft conceptual design under uncertainty





- Study risk-performance trade-off by varying the weights on mean and std. dev. in the constraints and resolve optimization problem

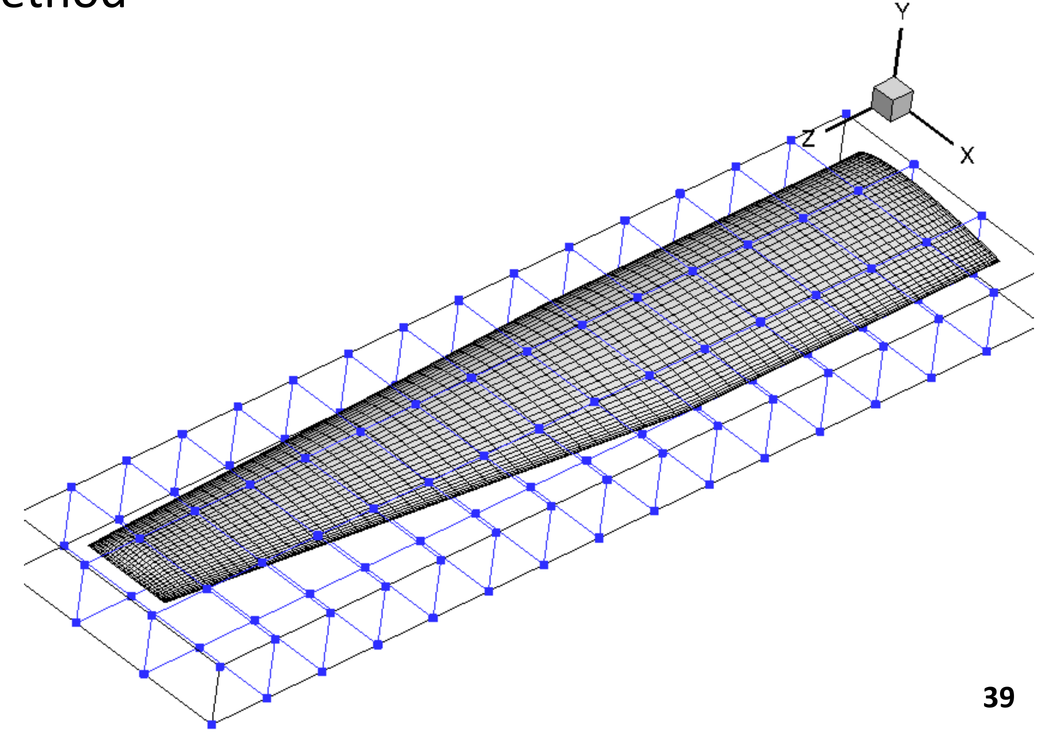
- Information reuse estimator is advantageous when re-solving the optimization problem
  - Reuse data from design points visited in previous optimization problem

High-fidelity wing optimization

- Shape optimization of (roughly) Bombardier Q400 wing
  - Free-form deformation geometry control (*Kenway et al. 2010*)

- Coupled aerostructural solver (*Kennedy & Martins 2010*)
  - Aerodynamics: TriPan panel method
  - Structures: Toolkit for the Analysis of Composite Structures (TACS) finite element method

	Coarse	Fine
Aerodynamic Panels	1000	2960
Structural d.o.f.	5624	14,288
Eval time	6 s	24 s



High-fidelity wing optimization

- 46 design variables:
  - 8 wing twist angles, 19 forward spar thicknesses, 19 aft spar thicknesses

- 7 random inputs:
  - Take-off weight, Mach number, material properties (density, elastic modulus, Poisson ratio, yield stress), wing weight fraction

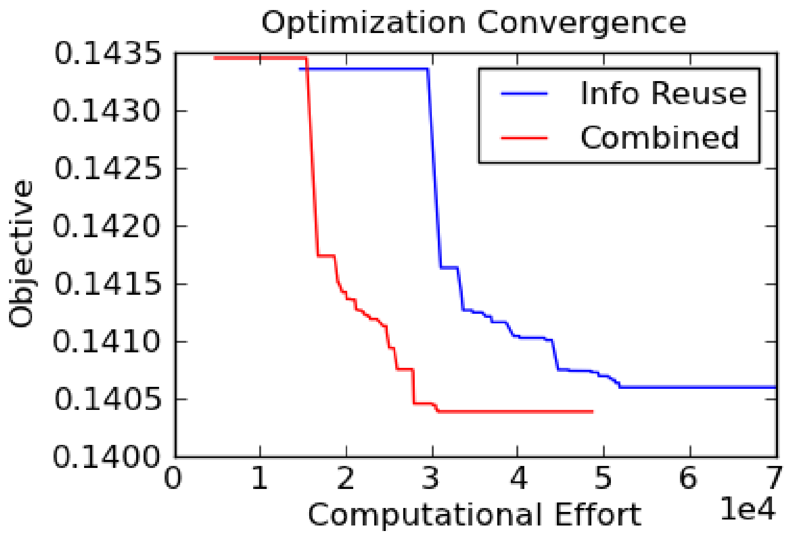
- Objective = drag (formulated as mean + 2 std)

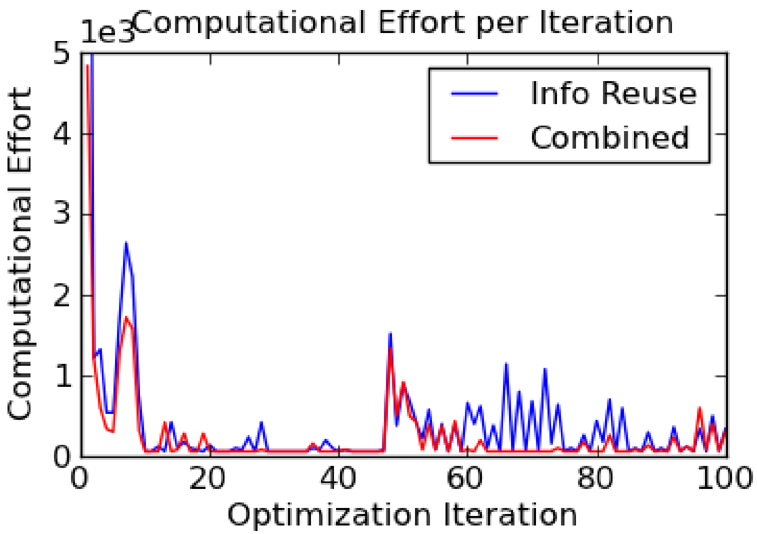
- 4 nonlinear stress constraints (formulated as mean + 2 std ≤ 0)

- 36 linear geometry constraints (deterministic)

- Optimization loop: COBYLA constrained derivative-free solver (*Powell 1994*)
- Simulation loop: Fixed RMSE for estimators specified, number of samples allowed to vary

High-fidelity wing optimization





- Solved on 16-processor desktop machine
- Combined estimator enable OUU solution in reasonable turnaround time
- Regular Monte Carlo estimator would take about 3.2 months

	Computational Effort	Total Time (days)
Regular MC	--	--
Info Reuse	7×10^4	13.4
Combined	5×10^4	9.7

Conclusions

Statistical and learning methods can improve our effectiveness in solving optimization problems, by helping us exploit the availability of multiple models and multiple optimization solvers

- Automatically tailoring optimization solvers to problem structure
- Recognizing when a problem is similar to problems solved before
- Using multiple sources of information (including surrogate models, previously evaluated designs) to accelerate optimization problem solution

Acknowledgements

- This work was supported by
  - NASA STTR Award Number NNX14CP65P
  - AFOSR Computational Mathematics Program, AFOSR MURI on Uncertainty Quantification (F. Fahroo)