Optimization Using Surrogates for Engineering Design

John Dennis, Mark Abramson, Rice University

Charles Audet, Gilles Couture,
École Polytechnique de Montréal

Andrew Booker, Evin Cramer,
Paul Frank, Joerg Gablonsky, Boeing Phantom Works

Optimization Using Surrogates for Engineering Design

John Dennis, Mark Abramson, Rice University

Charles Audet, Gilles Couture,
École Polytechnique de Montréal

Andrew Booker, Evin Cramer,
Paul Frank, Joerg Gablonsky, Boeing Phantom Works

Thanks to: AFOSR, Boeing, CSRI, LASCI, ExxonMobil, NSF, UTRC

Outline

- The problem $\&$ a one-shot solution approach

Outline

- The problem $\&$ a one-shot solution approach
- The barrier objective and Clarke's tools

Outline

- The problem & a one-shot solution approach
- The barrier objective and Clarke's tools
- The filter approach to the barrier objective

Outline

- The problem & a one-shot solution approach
- The barrier objective and Clarke's tools
- The filter approach to the barrier objective
- The Surrogate Management Framework (SMF)
- Numerical results for the SMF

Outline

- The problem & a one-shot solution approach
- The barrier objective and Clarke's tools
- The filter approach to the barrier objective
- The Surrogate Management Framework (SMF)
- Numerical results for the SMF
- Conclusions, Status, and Plans

The target optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega = \{x \in \Re^n : C(x) \leq 0\}, \end{array}$$

where $f, c_j : \Re^n \rightarrow \Re \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,

The target optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega = \{x \in \Re^n : C(x) \leq 0\}, \end{array}$$

where $f, c_j : \Re^n \longrightarrow \Re \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,
- the functions provide few correct digits and may fail even for $x \in \Omega$

The target optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega = \{x \in \Re^n : C(x) \leq 0\}, \end{array}$$

where $f, c_j : \Re^n \longrightarrow \Re \cup \{\infty\}$ may be discontinuous and:

- the functions are expensive black boxes,
- the functions provide few correct digits and may fail even for $x \in \Omega$
- accurate approximation of derivatives is problematic

Standard one-shot approach

1. Choose surrogates s_f and S_C based on either:

(a) simplified physical models; or

(b) surfaces obtained from f, C values at selected sites

Standard one-shot approach

1. Choose surrogates s_f and S_C based on either:

 (a) simplified physical models; or
 (b) surfaces obtained from f, C values at selected sites

2. Minimize $s_f(x)$ for $S_C(x) \leq 0$ to obtain x_s.
   Every user has their favorite approach for this part

Standard one-shot approach

1. Choose surrogates s_f and S_C based on either:
   (a) simplified physical models; or
   (b) surfaces obtained from f, C values at selected sites

2. Minimize $s_f(x)$ for $S_C(x) \leq 0$ to obtain x_s.
   Every user has their favorite approach for this part

3. Compute $f(x_s), C(x_s)$ to determine if improvement has been made over the best x found to date

Standard one-shot approach

1. Choose surrogates s_f and S_C based on either:

 (a) simplified physical models; or
 (b) surfaces obtained from f, C values at selected sites

2. Minimize $s_f(x)$ for $S_C(x) \leq 0$ to obtain x_s.
   Every user has their favorite approach for this part

3. Compute $f(x_s), C(x_s)$ to determine if improvement has been made over the best x found to date
   But, what if no improvement was found?

Closed constraints

Must be satisfied or else the functions can not be evaluated. Yes/No constraints usually closed - you just have to try...

Some constraints, like bounds, may as well hold at every iterate

To enforce simple closed constraints with a pattern search, replace $f(x)$ by the "barrier objective" $f_B(x)$ which is infinite when x is infeasible

Smoothness is assumed on f at the limit, not on f_B!!

The two phases of GPS algorithms

Global search in the variable space

Flexibility allows for user domain knowledge and heuristics like genetic algorithms or surrogate functions.

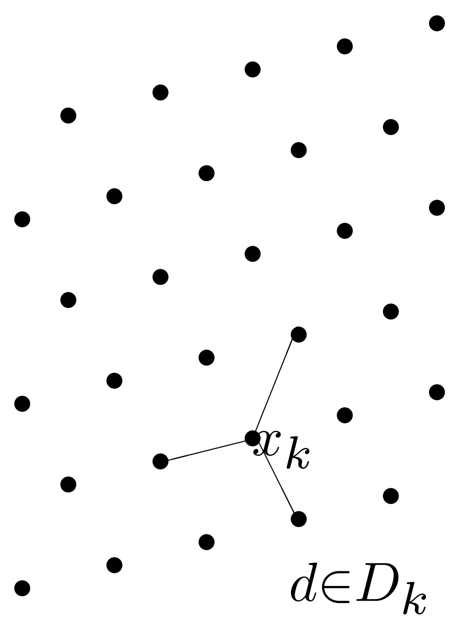
Local poll around the incumbent solution

More rigidly defined.

Ensures first order optimality to match problem assumptions.

The POLL set P_k

$P_k = \{x_k + \Delta_k d : d \in D_k \subset D\}$; points adjacent to x_k in M_k (wrt positive spanning set D_k).



Basic pattern search algorithm for unconstrained optimization

Given Δ_0, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,

Basic pattern search algorithm for unconstrained optimization

Given Δ_0, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,
for $k = 0, 1, \cdots$, do

1. Employ some finite strategy to try to choose $x_{k+1} \in M_k$ such that $f(x_{k+1}) < f(x_k)$ and then set $\Delta_{k+1} = \Delta_k$ or $\Delta_{k+1} = 2\Delta_k$ (x_{k+1} is called an *improved mesh point*);

Basic pattern search algorithm for unconstrained optimization

Given Δ_0, $x_0 \in M_0$ with $f(x_0) < \infty$, and D,
for $k = 0, 1, \cdots$, do

1. Employ some finite strategy to try to choose $x_{k+1} \in M_k$ such that $f(x_{k+1}) < f(x_k)$ and then set $\Delta_{k+1} = \Delta_k$ or $\Delta_{k+1} = 2\Delta_k$ (x_{k+1} is called an *improved mesh point*);

2. Else if x_k minimizes $f(x)$ for $x \in P_k$, then set $x_{k+1} = x_k$ and $\Delta_{k+1} = \Delta_k/2$ (x_k is called a *mesh local optimizer*)

An easy theorem

- Optimizers commonly assume that all the iterates they wish to analyze are in a compact set.

An easy theorem

- Optimizers commonly assume that all the iterates they wish to analyze are in a compact set. We will too

An easy theorem

- Optimizers commonly assume that all the iterates they wish to analyze are in a compact set. We will too
- Sometimes they do this by assuming that the level set of x_0 is compact, but we can't

An easy theorem

- Optimizers commonly assume that all the iterates they wish to analyze are in a compact set. We will too
- Sometimes they do this by assuming that the level set of x_0 is compact, but we can't

Theorem 1. *There is at least one limit point $\hat{x}$ of GPS. If f is lower semicontinuous at $\hat{x}$, then $\lim_{k\to\infty} f(x_k) \geq f(\hat{x})$ exists, and its value is taken on by every limit point of $\{x_k\}$ at which f is continuous.*

Directions are the key

- Let $f(x) = \|x\|_1$ in $\Re^2$, $x_0 = (1,0)$, and $D = \{(1,0),(-1,1),(-1,-1)\}$. Consider GPS with an empty SEARCH. The algorithm never moves from x_0.

Directions are the key

- Let $f(x) = \|x\|_1$ in $\Re^2$, $x_0 = (1,0)$, and $D = \{(1,0),(-1,1),(-1,-1)\}$. Consider GPS with an empty SEARCH. The algorithm never moves from x_0.
- Audet has a C^1 function for which GPS has infinitely many limit points, and one is not a stationary point.

Directions are the key

- Let $f(x) = \|x\|_1$ in $\Re^2$, $x_0 = (1,0)$, and $D = \{(1,0),(-1,1),(-1,-1)\}$. Consider GPS with an empty SEARCH. The algorithm never moves from x_0.

- Audet has a C^1 function for which GPS has infinitely many limit points, and one is not a stationary point. Note that since all limit points have the same function value, the nonstationary one is the most interesting.

Some Clarke background

- Let f be Lipschitz near x and v be a vector in $\Re^n$. The *generalized directional derivative* of f at x in the direction v is

$$f^\circ(x;v) \;:=\; \limsup_{y\to x,t\downarrow 0} \frac{f(y+tv)-f(y)}{t}$$

Some Clarke background

- Let f be Lipschitz near x and v be a vector in $\Re^n$. The *generalized directional derivative* of f at x in the direction v is

$$f^\circ(x;v) \;:=\; \limsup_{y\to x,t\downarrow 0} \frac{f(y+tv)-f(y)}{t}$$

- The *generalized gradient* of f at x is defined to be

$$\partial f(x) \;:=\; \{\zeta \in \Re^n : f^\circ(x;v) \geq v^T\zeta \;\forall v \in \Re^n\}$$

- Let f be Lipschitz near a given point x, then there is a neighborhood of x in which f is differentiable except on a subset of Lebesque measure 0

- Let f be Lipschitz near a given point x, then there is a neighborhood of x in which f is differentiable except on a subset of Lebesque measure 0

- $\partial f(x)$ is the convex hull of the limits of all gradients of sequences that converge to x

- $\partial f(x)$ equals the subdifferential for a convex f

- Let f be Lipschitz near a given point x, then there is a neighborhood of x in which f is differentiable except on a subset of Lebesque measure 0

- $\partial f(x)$ is the convex hull of the limits of all gradients of sequences that converge to x

- $\partial f(x)$ equals the subdifferential for a convex f

- If f is Lipschitz near a minimizer x, then $0 \in \partial f(x)$

- Let f be Lipschitz near a given point x, then there is a neighborhood of x in which f is differentiable except on a subset of Lebesque measure 0

- $\partial f(x)$ is the convex hull of the limits of all gradients of sequences that converge to x

- $\partial f(x)$ equals the subdifferential for a convex f

- If f is Lipschitz near a minimizer x, then $0 \in \partial f(x)$

- If f is differentiable at x, then the derivative of f at x is in the generalized gradient $\partial f(x)$

- Let f be Lipschitz near a given point x, then there is a neighborhood of x in which f is differentiable except on a subset of Lebesque measure 0

- $\partial f(x)$ is the convex hull of the limits of all gradients of sequences that converge to x

- $\partial f(x)$ equals the subdifferential for a convex f

- If f is Lipschitz near a minimizer x, then $0 \in \partial f(x)$

- If f is differentiable at x, then the derivative of f at x is in the generalized gradient $\partial f(x)$

- f is *regular* at x if for all v, the one-sided directional derivative exists and equals $f^\circ(x; v)$

- f is *regular* at x if for all v, the one-sided directional derivative exists and equals $f^\circ(x; v)$

- The function f is *strictly differentiable* at x if for all v,

$$\lim_{y \to x, t \downarrow 0} \frac{f(y + tv) - f(y)}{t} = \nabla f(x)^T v$$

- f is *regular* at x if for all v, the one-sided directional derivative exists and equals $f^\circ(x;v)$

- The function f is *strictly differentiable* at x if for all v,

$$\lim_{y \to x, t \downarrow 0} \frac{f(y+tv) - f(y)}{t} = \nabla f(x)^T v$$

- The function f is said to be *Gateaux differentiable* at x if for all v,

$$\lim_{t \downarrow 0} \frac{f(x+tv) - f(x)}{t} = \nabla f(x)^T v$$

- f is *regular* at x if for all v, the one-sided directional derivative exists and equals $f^\circ(x; v)$

- The function f is *strictly differentiable* at x if for all v,

$$\lim_{y \to x, t \downarrow 0} \frac{f(y + tv) - f(y)}{t} = \nabla f(x)^T v$$

- The function f is said to be *Gateaux differentiable* at x if for all v,

$$\lim_{t \downarrow 0} \frac{f(x + tv) - f(x)}{t} = \nabla f(x)^T v$$

- f is Lipschitz near x and differentiable and regular at x iff f is strictly differentiable at x iff $\partial f(x) = \{\nabla f(x)\}$

Convergence theorems

Lemma 1. *The mesh sizes satisfy* $\liminf_{k\to+\infty} \Delta_k = 0$.

Convergence theorems

Lemma 1. *The mesh sizes satisfy* $\liminf_{k \to +\infty} \Delta_k = 0$.

Lemma 2. *There exists a convergent sequence of mesh local optimizers for which the corresponding subsequence of* $\{\Delta_k\}$ *goes to zero.*

Convergence theorems

Lemma 1. *The mesh sizes satisfy* $\liminf_{k \to +\infty} \Delta_k = 0$.

Lemma 2. *There exists a convergent sequence of mesh local optimizers for which the corresponding subsequence of* $\{\Delta_k\}$ *goes to zero. If* f *is Lipschitz near any such limit* $\hat{x}$*, then there is a positive spanning set of columns of* D *on which the generalized directional derivatives of* f *are nonnegative.*

Convergence theorems

Lemma 1. *The mesh sizes satisfy* $\liminf_{k \to +\infty} \Delta_k = 0$.

Lemma 2. *There exists a convergent sequence of mesh local optimizers for which the corresponding subsequence of* $\{\Delta_k\}$ *goes to zero. If* f *is Lipschitz near any such limit* $\hat{x}$, *then there is a positive spanning set of columns of* D *on which the generalized directional derivatives of* f *are nonnegative. Thus, if* f *is strictly differentiable at* $\hat{x}$, *then* $\partial f(\hat{x}) = \{\nabla f(\hat{x})\} = \{0\}$.

Linear constraints

D must be rich enough to contain tangent cone generators for $T_X(y)$ for every $y \in X$, as in:

Definition. *A rule for selecting the positive spanning sets* $D_k = D(k, x_k) \subseteq D$ conforms to X *for some* $\epsilon > 0$, *if at each iteration* k *and for each* y *in the boundary of* X *for which* $\|y - x_k\| < \epsilon$, $T_X(y)$ *is generated by a nonnegative linear combinations of the columns of a subset* D_k^y *of* D_k.

Convergence to KKT points

Theorem. *If f is strictly differentiable at a limit point $\hat{x}$ of a refining subsequence, and if the rule for selecting the positive spanning sets $D_k = D(k, x_k) \subseteq D$ conforms to X for an $\epsilon > 0$, then $\nabla f(\hat{x})^T w \geq 0$ for all $w \in T_X(\hat{x})$, and so $-\nabla f(\hat{x}) \in N_X(\hat{x})$. Thus, $\hat{x}$ is a KKT point.*

Limitations of GPS methods

- They are directional, so the restriction to a finite set of directions is a big limitation, particularly for dealing with nonlinear constraints.

Limitations of GPS methods

- They are directional, so the restriction to a finite set of directions is a big limitation, particularly for dealing with nonlinear constraints.
- They require all trial points be on the mesh, which limits the freedom in the user defined search. This is more important in concept than practice.

Limitations of GPS methods

- They are directional, so the restriction to a finite set of directions is a big limitation, particularly for dealing with nonlinear constraints.
- They require all trial points be on the mesh, which limits the freedom in the user defined search. This is more important in concept than practice.

What next? Frame-based methods are best candidates, but....

Open constraints

Need not be satisfied until convergence.

Open constraints

Need not be satisfied until convergence.

Let $h(x)$ be the aggregate open constraint violation at x.

Ways to handle open constraints:

Open constraints

Need not be satisfied until convergence.

Let $h(x)$ be the aggregate open constraint violation at x.

Ways to handle open constraints:

- *Choose ρ large enough* and minimize $f_B(x) + \rho \times h(x)$

Open constraints

Need not be satisfied until convergence.

Let $h(x)$ be the aggregate open constraint violation at x.

Ways to handle open constraints:

- *Choose* ρ *large enough* and minimize $f_B(x) + \rho \times h(x)$
- Use a filter method and forget about ρ

Open constraints

Need not be satisfied until convergence.

Let $h(x)$ be the aggregate open constraint violation at x.

Ways to handle open constraints:

- *Choose ρ large enough* and minimize $f_B(x) + \rho \times h(x)$
- Use a filter method and forget about ρ

Penalty methods may require ρ to become very large and they may not be robust

The filter for open constraints

Define the constraint violation function

$$h(x) = \sum_j \max(0, c_j(x))^2.$$

Note that $h(x) = 0$ iff $x \in \Omega$, and h inherits smoothness from C.

The filter for open constraints

Define the constraint violation function

$$h(x) = \sum_j \max(0, c_j(x))^2.$$

Note that $h(x) = 0$ iff $x \in \Omega$, and h inherits smoothness from C.

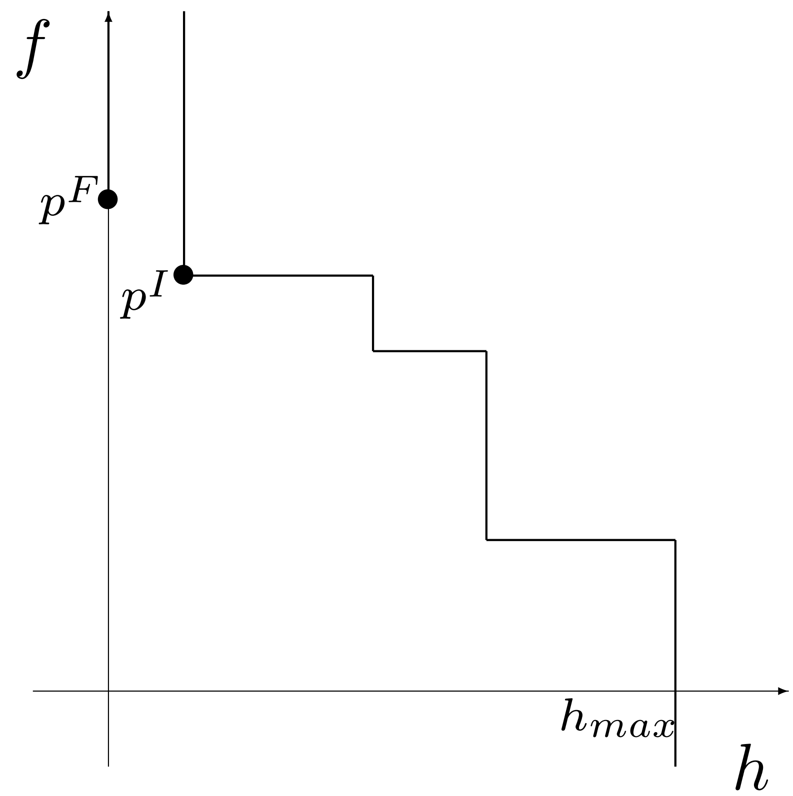
Consider the biobjective optimization problem in f and h where priority is given to the minimization of h over the minimization of f.

Filtered pattern search methods

A trial point x is *unfiltered* if $h(x) < h_{max}$ and no earlier point is at least as optimal and as feasible.

SUCCESSFUL iterations find unfiltered points.

UNSUCCESSFUL iterations don't, even after polling around $p_k \in \{p^F, p^I\}$ the feasible incumbent p^F or the least infeasible point p^I).



Results for constrained optimization: h

For every limit point $\hat{x}$ of any subsequence $\{p_k\}_{k \in K}$ of unsuccessful poll centers where $\{\Delta_k\}_{k \in K} \to 0$, and for the set $\hat{D}$ of associated directions used infinitely many times in this subsequence,

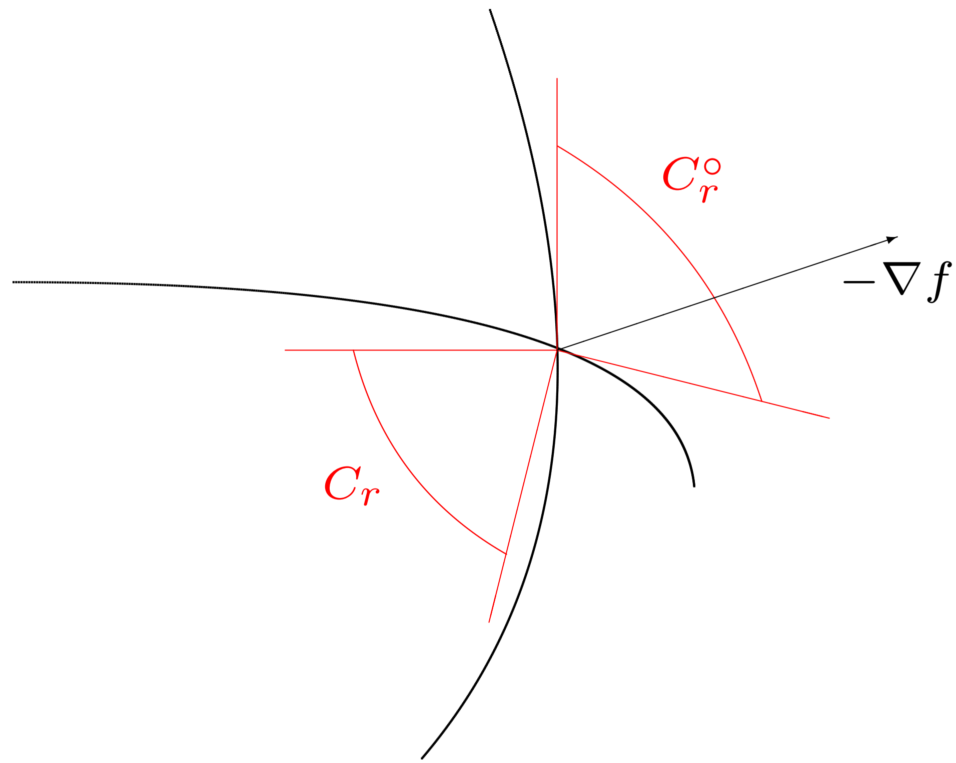
i) h Lipschitz near $\hat{x} \;\Rightarrow h^\circ(\hat{x}; d) \geq 0$ for any $d \in \hat{D}$;

ii) h strictly differentiable at $\hat{x} \Rightarrow \nabla h(\hat{x}) = 0$;

Results for constrained optimization: f

iii) If $h(p_k) = h(p_k + \Delta_k d)$ for a subsequence of unsuccessful iterations, then $f^\circ(\hat{x}; d) \geq 0$.

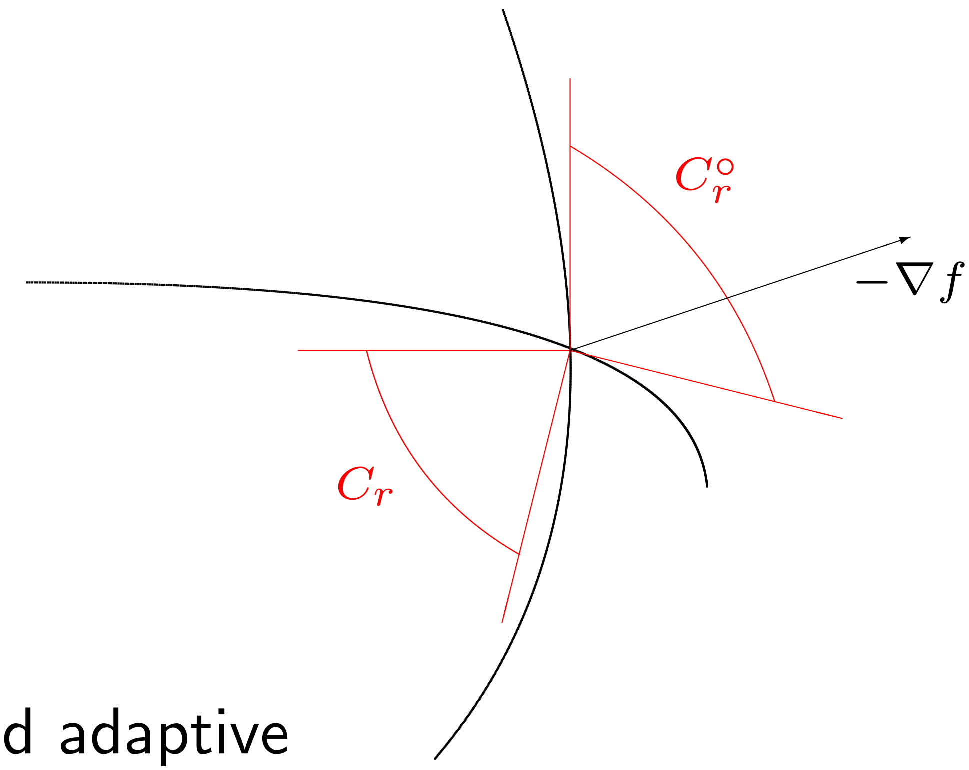
iv) Let C_r be the cone generated by all such directions of $\hat{D}$. If f is strictly differentiable, then $-\nabla f(\hat{x})$ is in the polar C_r°, which contains the normal $N_\Omega(\hat{x})$.



Results for constrained optimization: f

iii) If $h(p_k) = h(p_k + \Delta_k d)$ for a subsequence of unsuccessful iterations, then $f^\circ(\hat{x}; d) \geq 0$.

iv) Let C_r be the cone generated by all such directions of $\hat{D}$. If f is strictly differentiable, then $-\nabla f(\hat{x})$ is in the polar C_r°, which contains the normal $N_\Omega(\hat{x})$.



$Future$: We are working to find adaptive choices of directions to close this KKT gap.

Optimization using surrogate functions

Poll step: Polling is done on the actual (barrier) functions, which guarantees convergence of the algorithm (through the sequence of unsuccessful poll steps).

Search step: Use surrogates s_f, S_C to find some promising candidates where f, C will be evaluated.

Update s_f, S_C with the new evaluations of f, C.

Surrogate Management Framework

Given initial surrogates s_f, s_C and $p_0 \in M_0$, a grid on $\Re^n$, let $P_0 \subset M_0$ be p_0 and the points of M_0 adjacent to x_0.

Surrogate Management Framework

Given initial surrogates s_f, s_C and $p_0 \in M_0$, a grid on $\Re^n$, let $P_0 \subset M_0$ be p_0 and the points of M_0 adjacent to x_0.

For $k = 0, 1, \cdots$, do

Surrogate Management Framework

Given initial surrogates s_f, s_C and $p_0 \in M_0$, a grid on $\Re^n$, let $P_0 \subset M_0$ be p_0 and the points of M_0 adjacent to x_0.

For $k = 0, 1, \cdots$, do

1. Search on s_f, s_C to find an unfiltered $x_{k+1} \in M_k$ and then set $M_{k+1} = M_k$ and update the surrogates;

Surrogate Management Framework

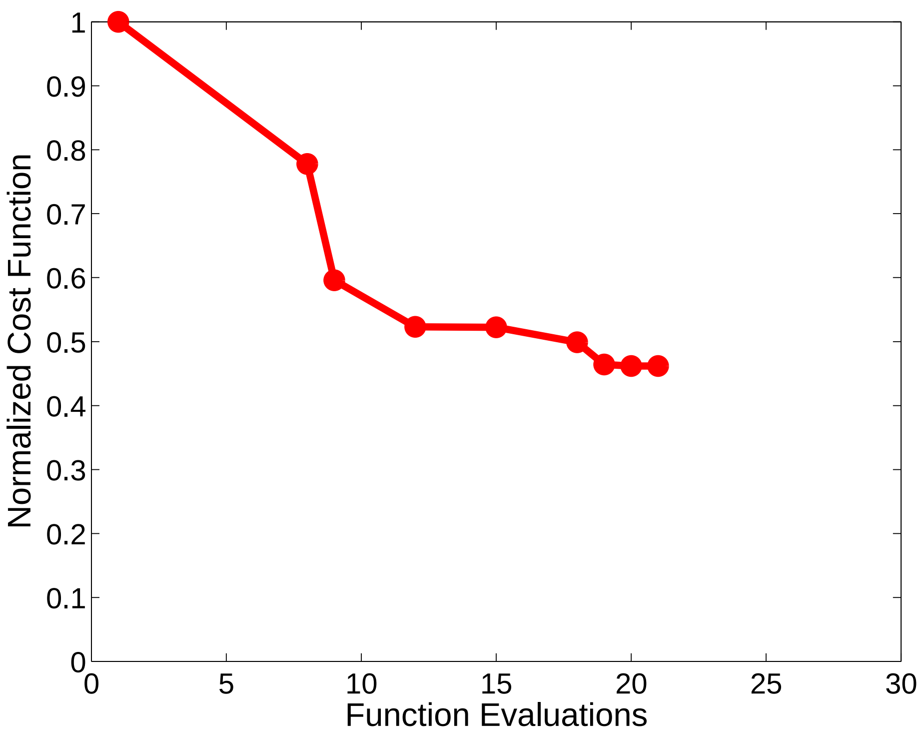
Given initial surrogates s_f, s_C and $p_0 \in M_0$, a grid on $\Re^n$, let $P_0 \subset M_0$ be p_0 and the points of M_0 adjacent to x_0.

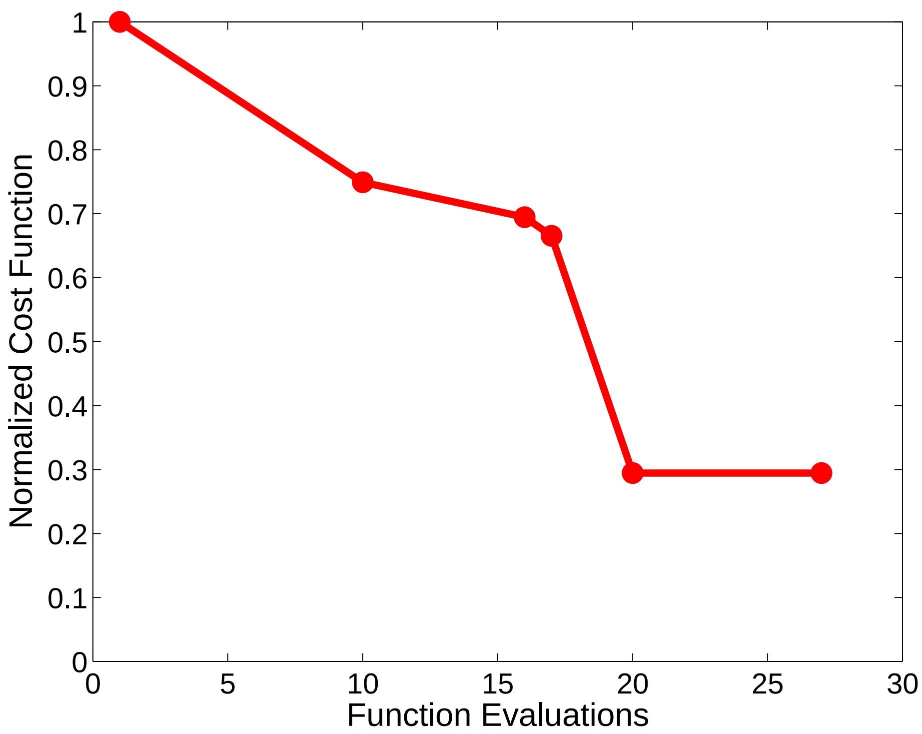
For $k = 0, 1, \cdots,$ do

1. Search on s_f, s_C to find an unfiltered $x_{k+1} \in M_k$ and then set $M_{k+1} = M_k$ and update the surrogates;
2. Else if p_k is the only unfiltered point in P_k; Then set $x_{k+1} = x_k$ and $M_{k+1} = M_k/2$ and update the surrogates; Else return to 1.

Trailing edge design - before and after IMA

Optimize shape for total acoustic noise reduction. Work of Stanford grad student Alison Marsden, an IMA shortcourse participant.

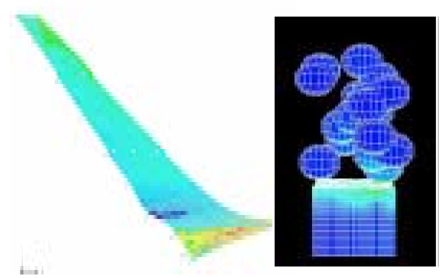




Design Explorer Applications

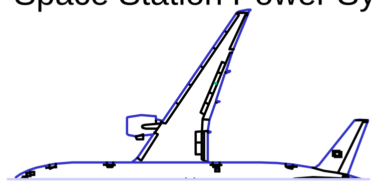
Helicopter Rotor Design

Space Station Power System



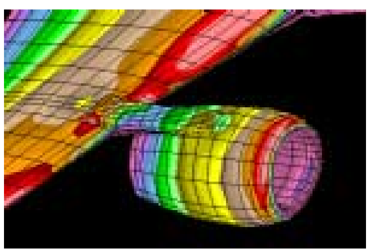
Shot peen forming of wing skins

Aerospike Nozzle



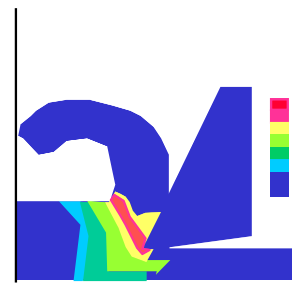
Multidisciplinary wing planform design

3-D Fighter Aerodynamics



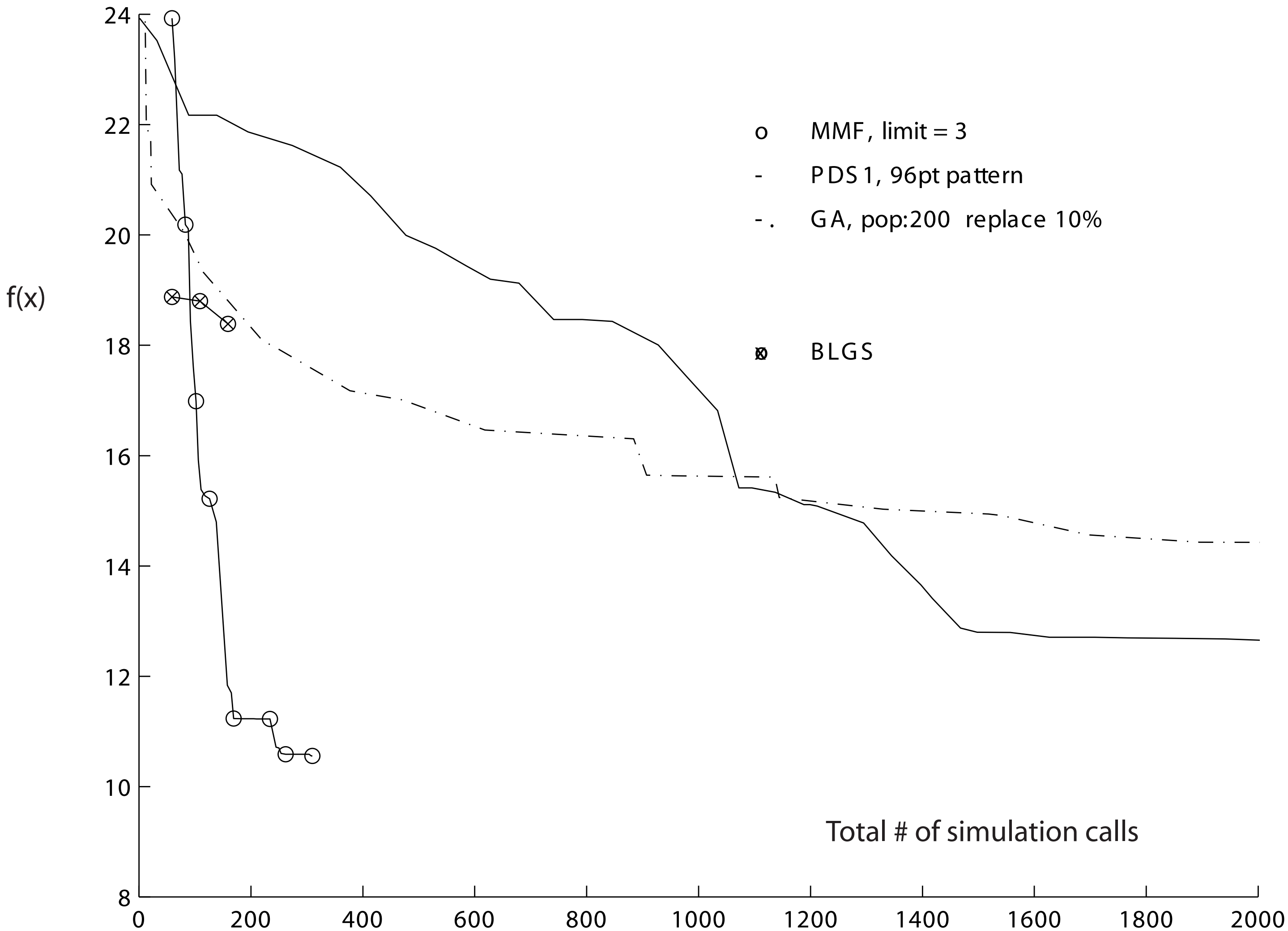
Engine Nozzle Performance

777 Engine Duct Seals



Machining,riveting,and drilling database

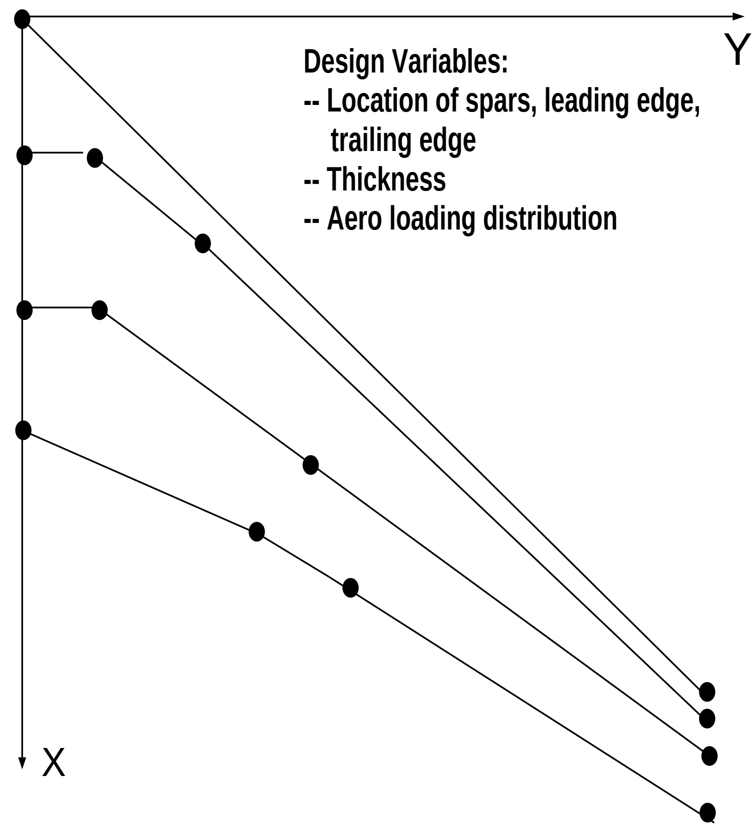
Comparisons on 31d helicopter example



Boeing wing planform design

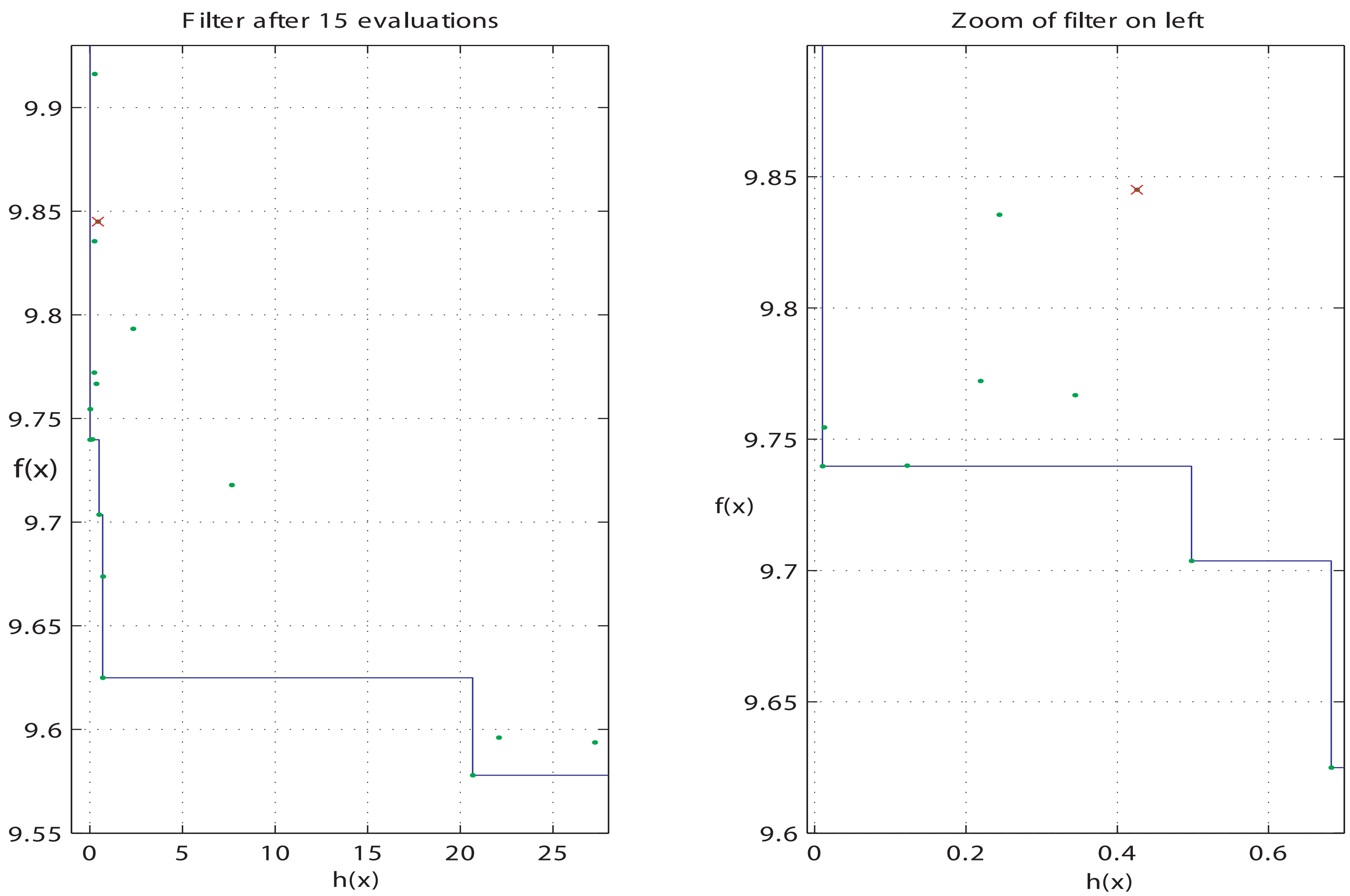
Infeasible baseline design.

	n	# of ctrs	# of fevals
A	15	11	304
B	15	11	292

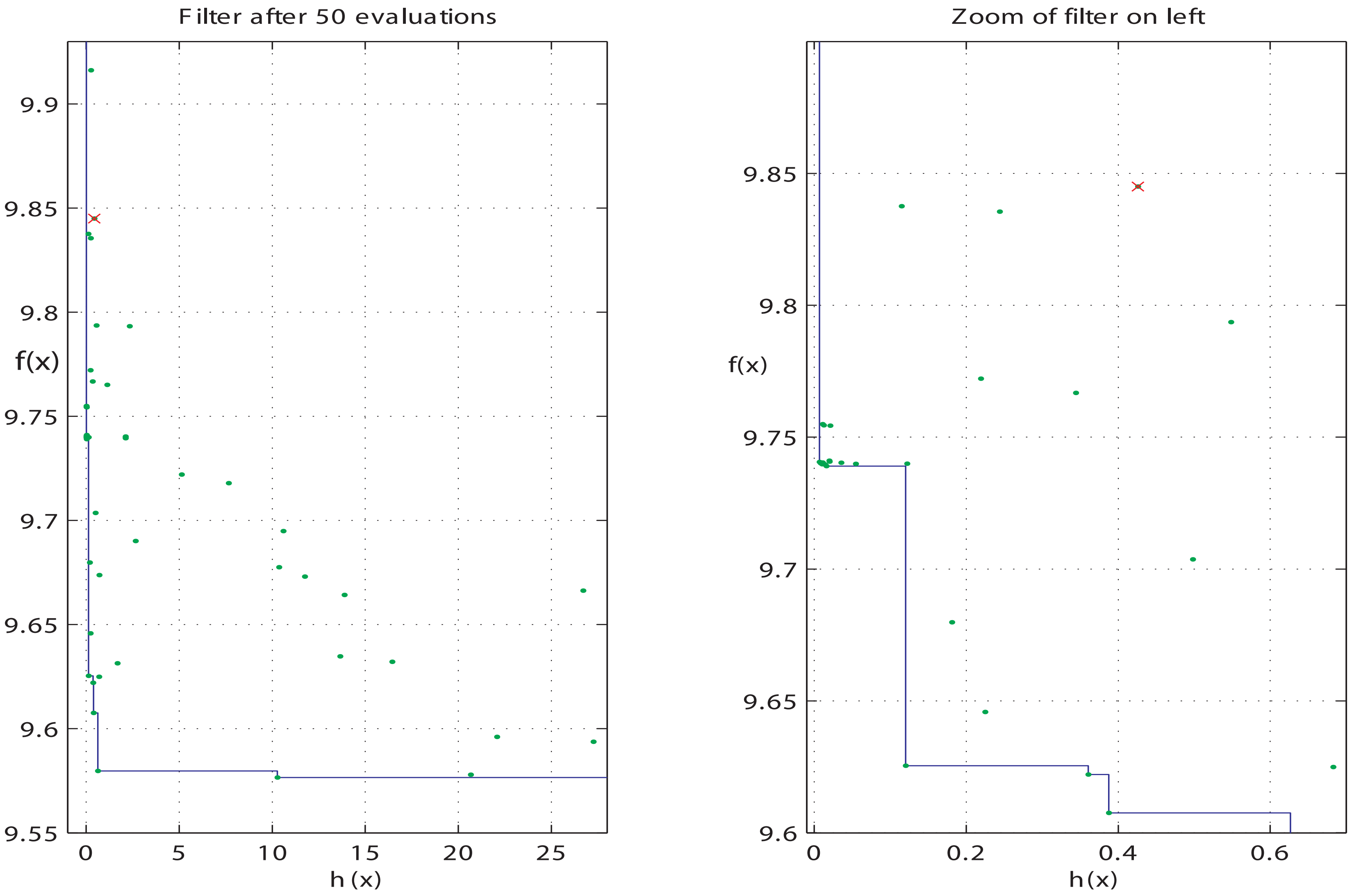


SonicCruiser planform (cancelled)

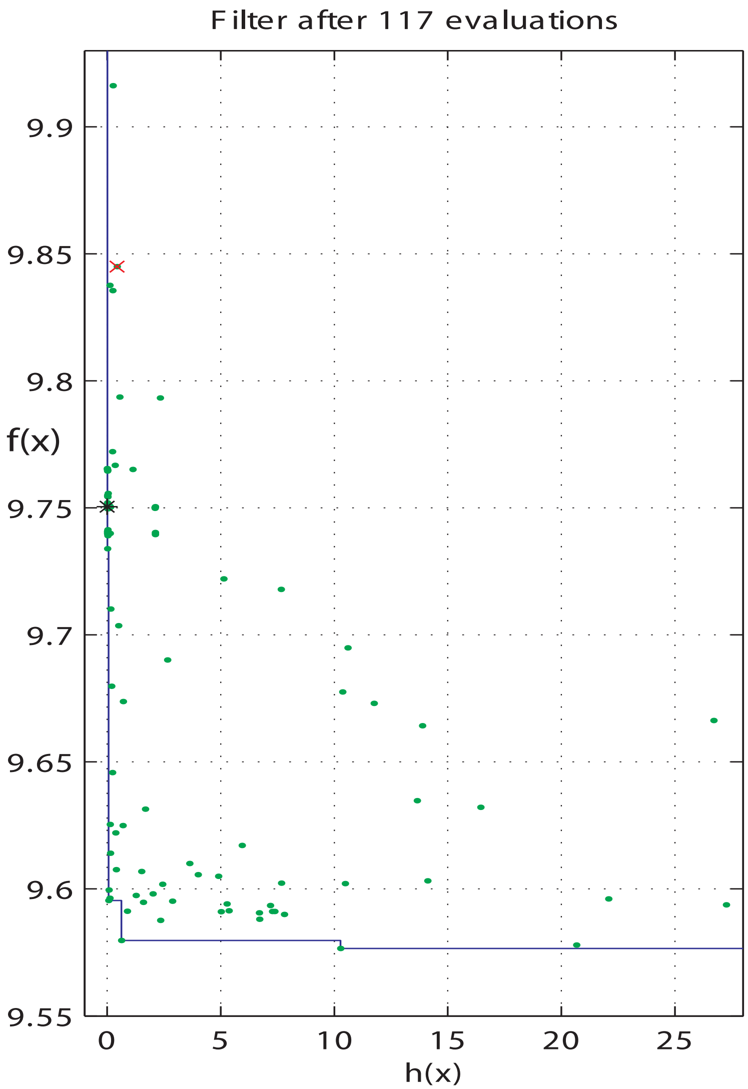
A planform filter

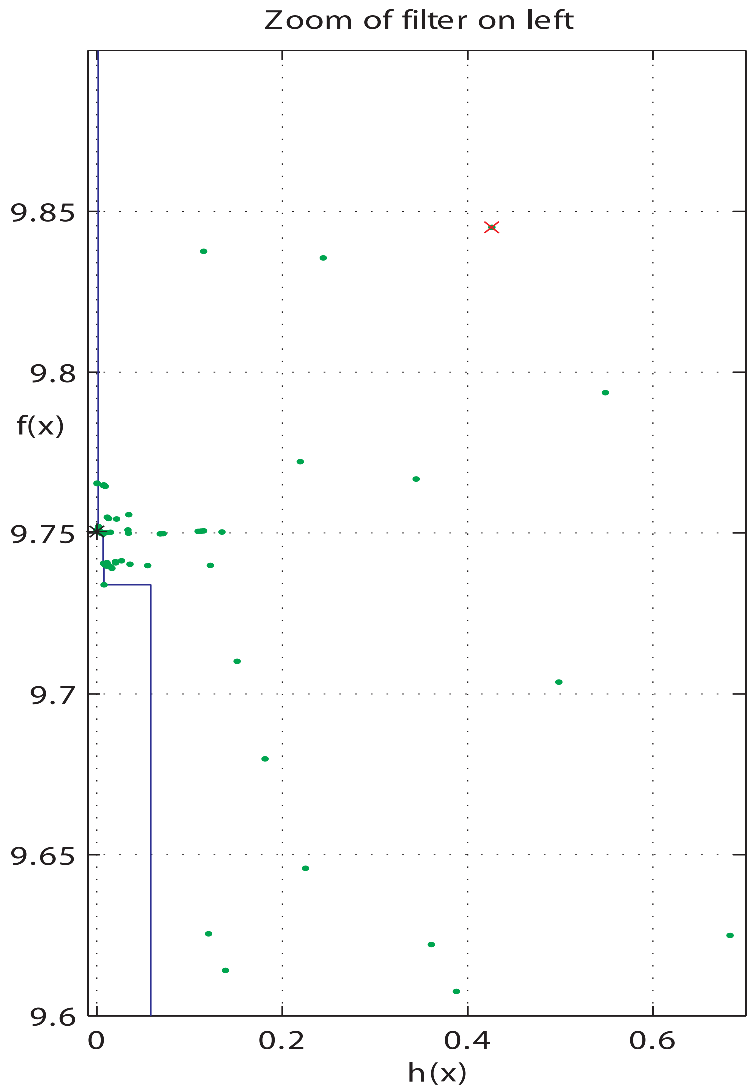


A planform filter

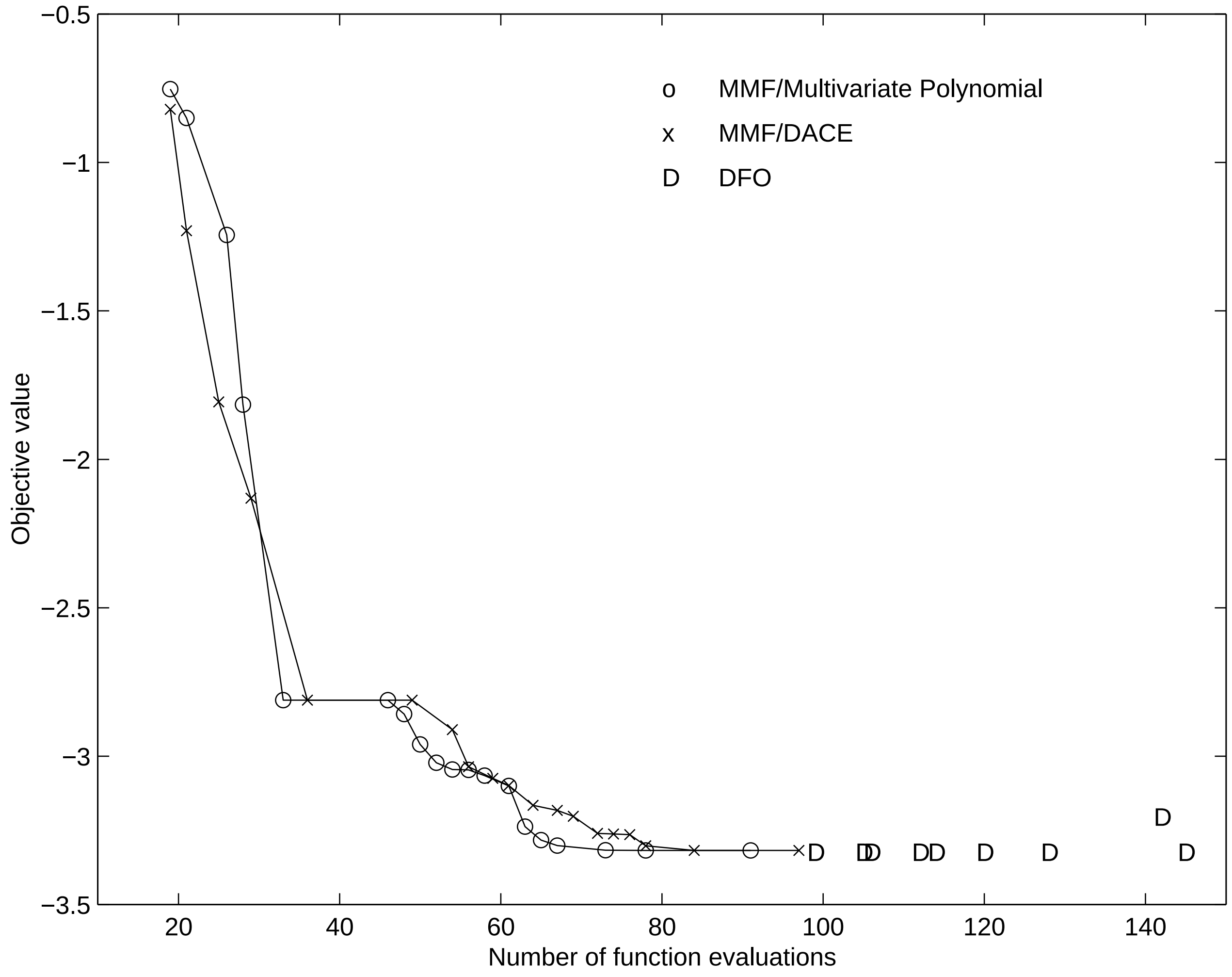


A planform filter

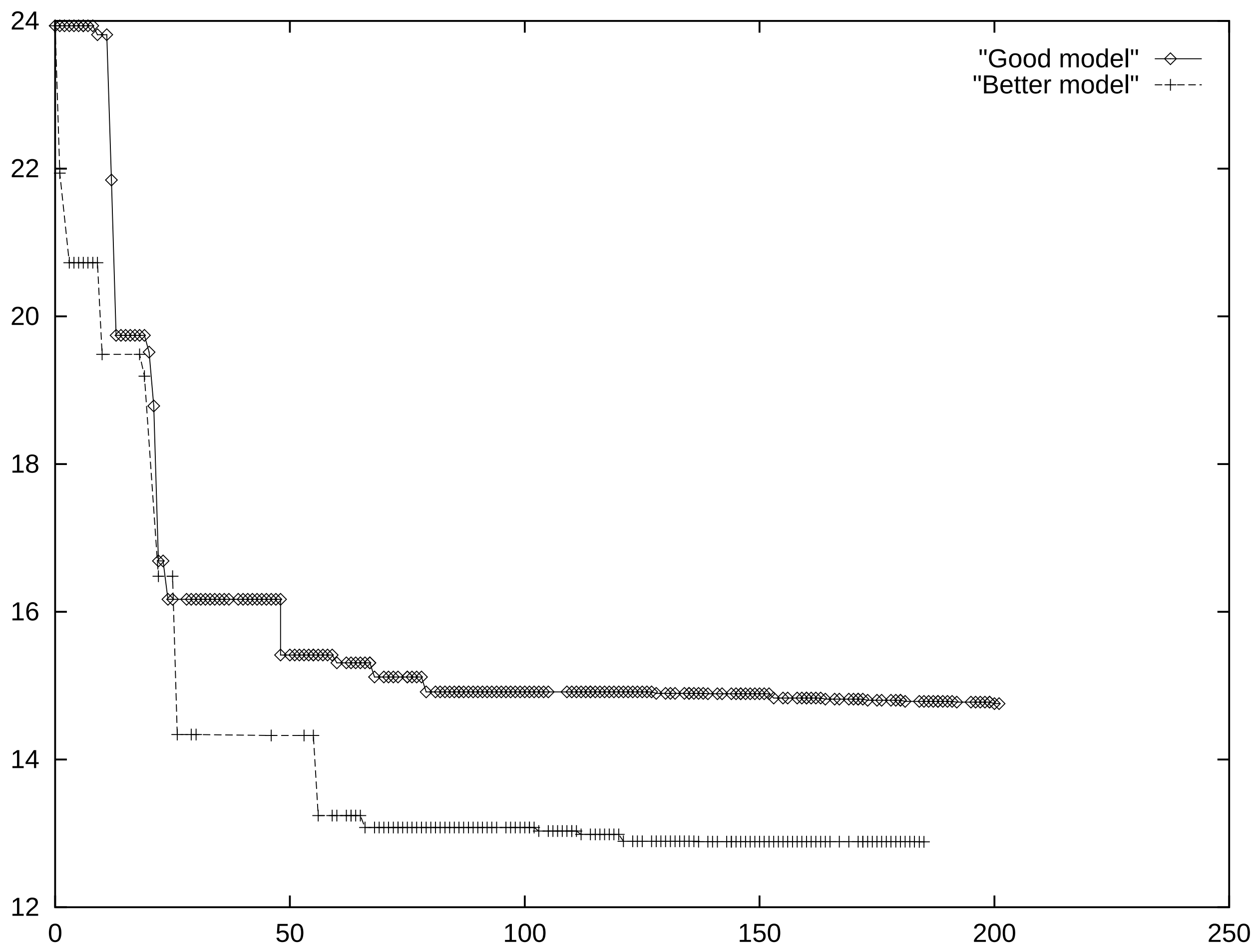




Polynomial vs DACE surrogates



Recovering from a bad surrogate



Software

- Boeing Design Explorer uses our SMF with proprietary surrogates and a specialized search. Commercialization is underway

Software

- Boeing Design Explorer uses our SMF with proprietary surrogates and a specialized search. Commercialization is underway

- NOMADm is a downloadable MatLab implementation in production use by BMW for designing diesel engine control laws.

Software

- Boeing Design Explorer uses our SMF with proprietary surrogates and a specialized search. Commercialization is underway

- NOMADm is a downloadable MatLab implementation in production use by BMW for designing diesel engine control laws.

- ExxonMobil and UTRC licensing NOMAD - C++

Conclusions

- Clarke's nonsmooth analysis tools simplify, shorten, and strengthen the pattern search analysis
- Filters handle nonlinear constraints without using derivatives, Lagrange multipliers or penalty parameters
- SMF seamlessly incorporates user heuristic SEARCHES, and POLLING provides robustness and rigor
- Designers want more accurate surrogates than SMF needs

Plans

- Need for higher dimension than present interpolatory surrogates can handle. Requires rethinking the surrogate/optimization interface. High payoff, high risk

Plans

- Need for higher dimension than present interpolatory surrogates can handle. Requires rethinking the surrogate/optimization interface. High payoff, high risk
- Put filter approach to categorical variables from NOMADm into NOMAD and test on pipe bending problems. Usability issues

Plans

- Need for higher dimension than present interpolatory surrogates can handle. Requires rethinking the surrogate/optimization interface. High payoff, high risk
- Put filter approach to categorical variables from NOMADm into NOMAD and test on pipe bending problems. Usability issues
- Use crude/inaccurate derivatives to increase efficiency. Low risk

Plans

- Put NOMAD into Boeing Design Explorer. High payoff, low risk

Plans

- Put NOMAD into Boeing Design Explorer. High payoff, low risk
- Parallelize NOMAD. Open ended , high payoff, low risk

Plans

- Put NOMAD into Boeing Design Explorer. High payoff, low risk
- Parallelize NOMAD. Open ended , high payoff, low risk
- Integrate Ferris-Mangasarian decomposition with filters. High risk, high payoff

Plans

- Put NOMAD into Boeing Design Explorer. High payoff, low risk
- Parallelize NOMAD. Open ended , high payoff, low risk
- Integrate Ferris-Mangasarian decomposition with filters. High risk, high payoff
- Implement Coope & Price "bent frames". High risk, theoretical payoff is high

Expanding the Realm of Applications for Response Modeling