Optimization Using Surrogate Objectives

John Dennis, Mark Abramson, Rice University

Charles Audet, Gilles Couture,
École Polytecnique. de Montréal

Andrew Booker, Mike Carter, Evin Cramer,
Paul Frank, Greg Shubin, Boeing Phantom Works

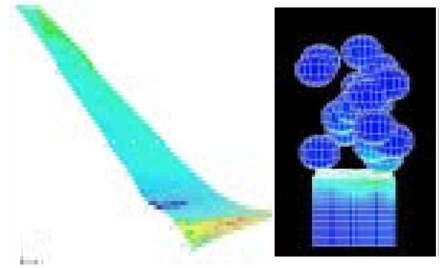
Thanks to: AFOSR, Boeing, DOE, ExxonMobil, NSF

Earlier Collaborators: **Dave Serafini**, Knight Financial
Virginia Torczon, Michael Trosset, William & Mary,

Design Explorer Applications

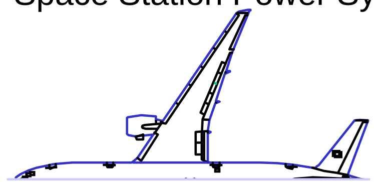
Helicopter Rotor Design

Space Station Power System



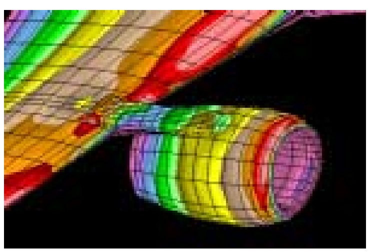
Shot peen forming of wing skins

Aerospike Nozzle



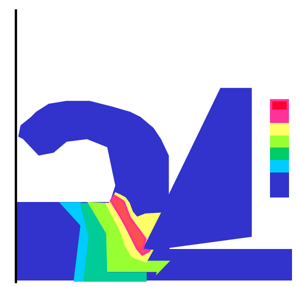
Multidisciplinary wing planform design

3-D Fighter Aerodynamics



Engine Nozzle Performance

777 Engine Duct Seals



Machining,riveting,and drilling database

Outline

- The Problem & A One-Shot Solution Approach
- 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 $f, c_j : \Re^n \longrightarrow \Re \cup \{\infty\}$ may be discontinuous and:

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

Standard one-short 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.
   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.

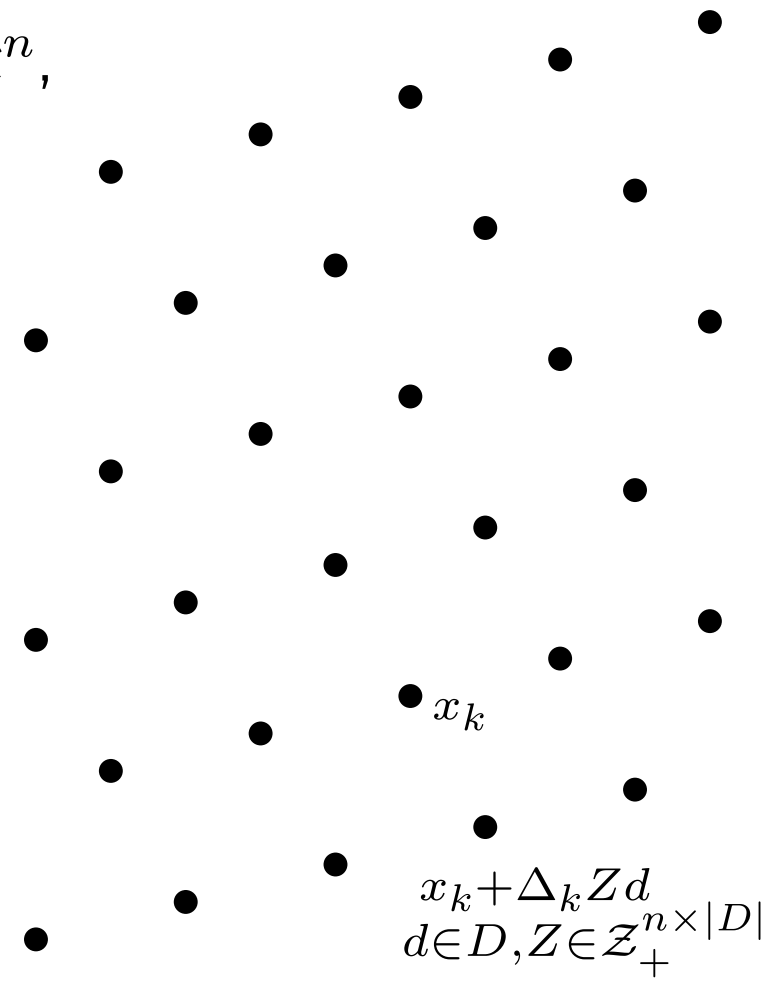
LOCAL POLL around the incumbent solution

More rigidly defined.

Ensures first order optimality to match problem assumptions.

Basic pattern search algorithm for unconstrained optimization

minimize $f(x)$ subject to $x \in \Omega \equiv \Re^n$,
Given Δ_0, x_0 in initial mesh M_0,
repeat:

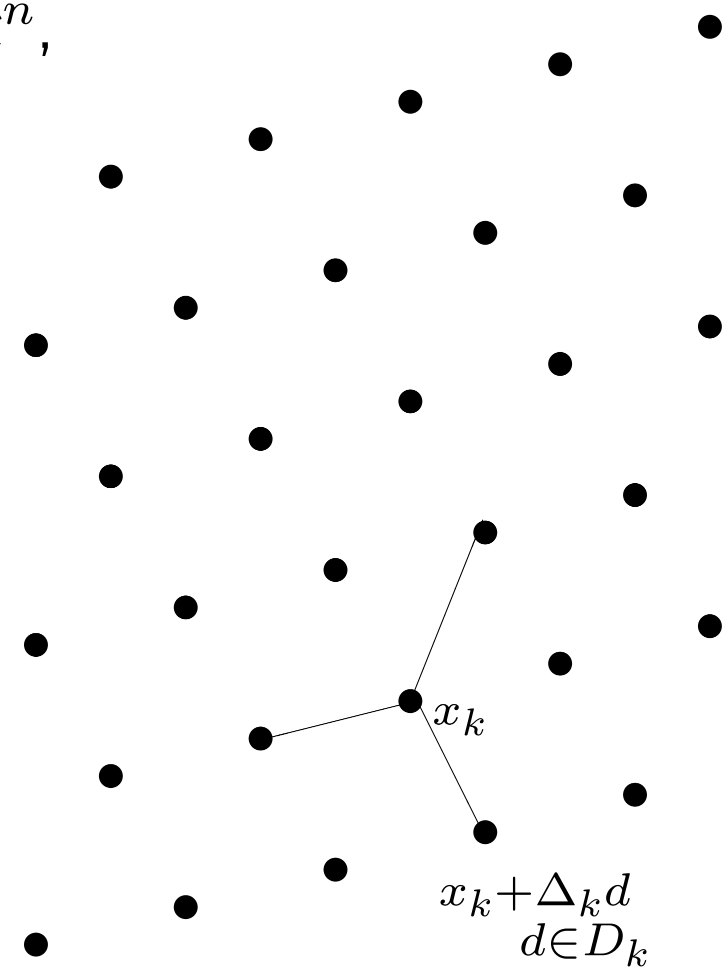


Unconstrained GPS

minimize $f(x)$ subject to $x \in \Omega \equiv \Re^n$,

Given Δ_0, x_0 in initial mesh M_0,
repeat:
– SEARCH anywhere on M_k
– POLL around x_k on M_k

Each iteration is either
SUCCESSFUL:
$f(x_{k+1}) < f(x_k)$ and $\Delta_{k+1} \geq \Delta_k$
UNSUCCESSFUL:
$x_{k+1} = x_k$ and $\Delta_{k+1} < \Delta_k$



Convergence results on barrier approach

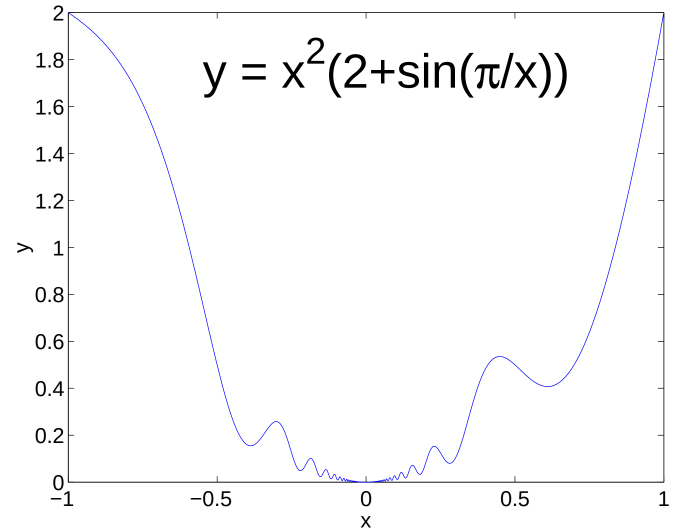
- If all iterates are in a compact set, then the mesh gets infinitely fine ($\liminf_k \Delta_k = 0$) Torczon‘97
- For every limit $\hat{x}$ of any subsequence $\{x_k\}_{k\in K}$ of unsuccessful iterates where $\{\Delta_k\}_{k\in K} \to 0$, and for the set $\hat{D}$ of directions used infinitely often in $\{x_k\}_{k\in K}$,

If f is Lipschitz near $\hat{x}$, then $f^\circ(\hat{x}; d) \geq 0, \ \forall d \in \hat{D}$.

PROOF:
$$\begin{aligned} f^\circ(\hat{x}; d) &:= \limsup_{y\to\hat{x},\ t\downarrow 0} \frac{f(y+td) - f(y)}{t} \\ &\geq \limsup_{k\in K} \frac{f(x_k + \Delta_k d) - f(x_k)}{\Delta_k} \ \geq \ 0. \end{aligned}$$

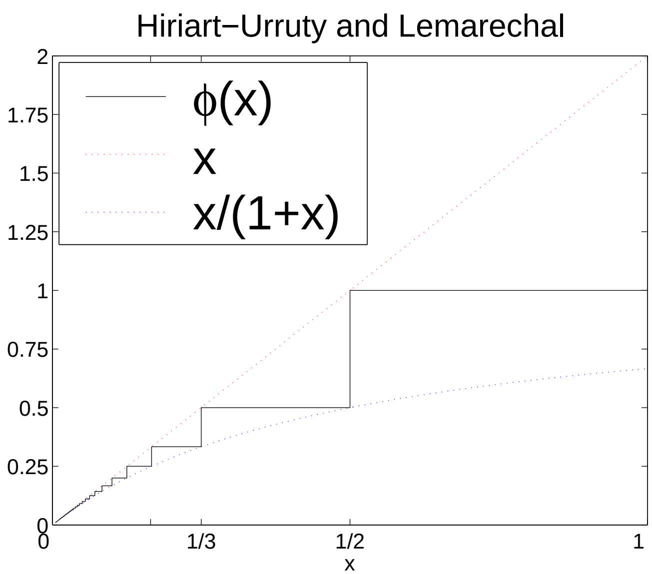
Convergence results on barrier approach

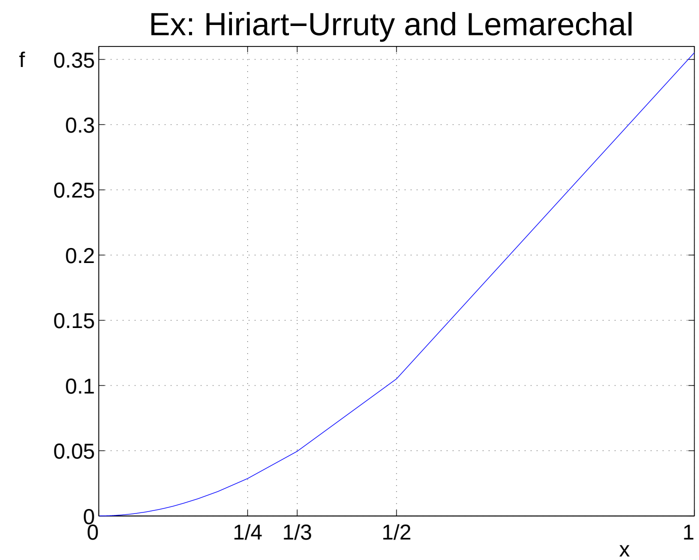
Any GPS algorithm applied to this f will produce a local minimizer or $\hat{x} = 0$, the only point with $f^{\circ}(\hat{x}; \pm 1) = 1 \geq 0$.



♦ *If f is regular at $\hat{x}$, then $f'(\hat{x}; d) \geq 0$ for every $d \in \hat{D}$.*

Convergence results on barrier approach





Consider the convex function $f : \Re \longrightarrow \Re$, $f(x) = \int_0^x \phi(u)du$,

where $\phi(u) = \begin{cases} u & \text{if } u \leq 0 \\ \frac{1}{1+\kappa} & \text{if } \kappa + 1 > \frac{1}{u} \geq \kappa \in \mathcal{Z}_+. \end{cases}$

Convergence results on barrier approach

f has kinks at $\frac{1}{\kappa}$ with $\partial f(\frac{1}{\kappa}) = [\frac{1}{\kappa+1}, \frac{1}{\kappa}]$, $\kappa = 1, 2, \ldots$
so f is not continuously differentiable near $\hat{x}$.

$\partial f(0)$ reduces to the singleton $\{0\} \Rightarrow f$ is strictly differentiable at $\hat{x}$.

Any pattern search algorithm finds the optimal solution.

♦ *If f is strictly differentiable at $\hat{x}$, then $\nabla f(\hat{x}) = 0$.*

Positive spanning sets

- A richer set of D_k increases the number of directions in $D_k \to \hat{D}$, and thus the directions for which $f^\circ(\hat{x}; d) \geq 0$.

- User's domain specific knowledge can help choose D_k.

- Theory limited to a finite number of different D_k, so the *barrier* approach (reject infeasible points) does not apply to general constraints because a finite number can not conform to $\partial\Omega$.

Open constraints

Need not be satisfied at every iteration.

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. They are so common that we forget their inelegance*.

The filter for open constraints

Define the nonnegative constraint violation function

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

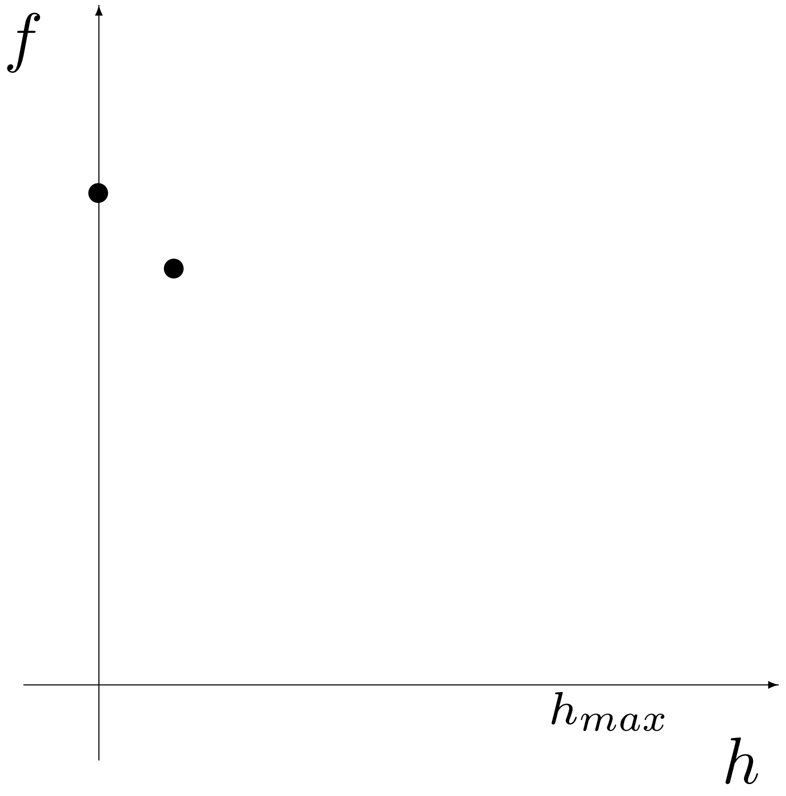
Note that $h(x) = 0$ iff $x \in \Omega$, and h inherits smoothness from C.

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

Combining pattern search and filter methods

Each time a trial point x is considered, $h(x)$ and $f(x)$ are plotted the bi-loss map.

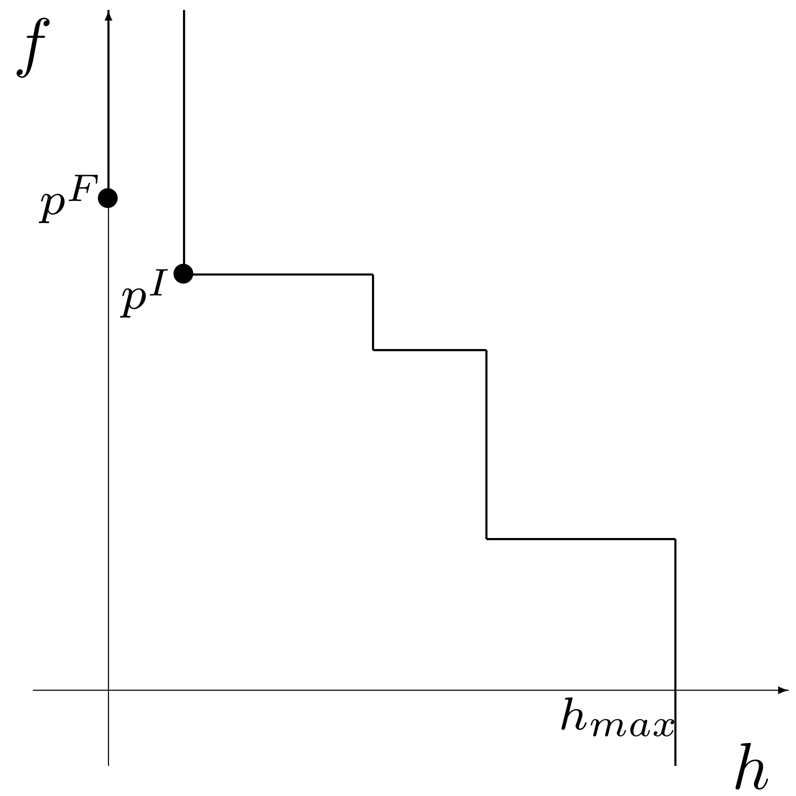
Only trial points with $h < h_{max}$ are considered.



Polling centers

SUCCESSFUL iterations find unfiltered points.

UNSUCCESSFUL iterations don't, even after polling around p_k (chosen as either the feasible incumbent p^F or the most feasible 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,

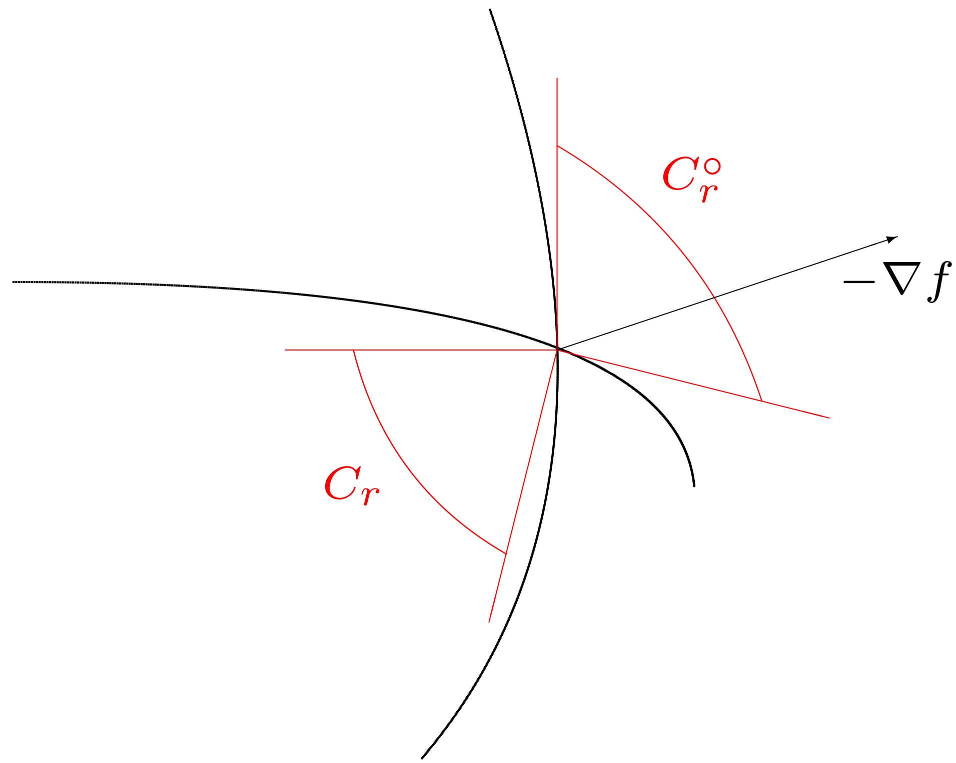
i) h Lipschitz near $\hat{x} \;\Rightarrow h^\circ(\hat{x}; d) \geq 0$ for any $d \in \hat{D}$;

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 $N_\Omega(\hat{x})$.



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.

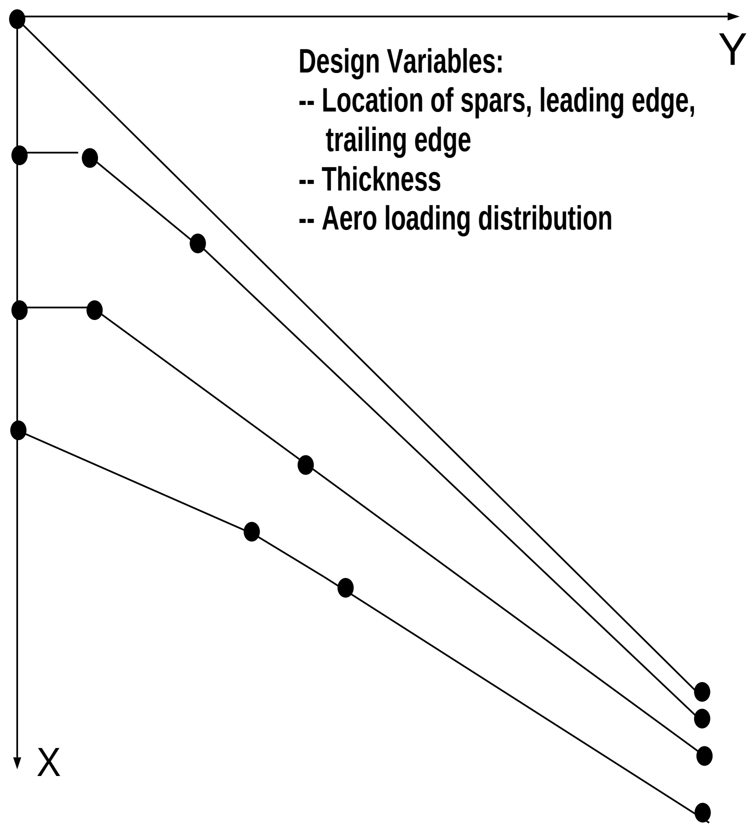
For $k = 0, 1, \cdots$, do

1. Search on s_f, s_C to find and 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.

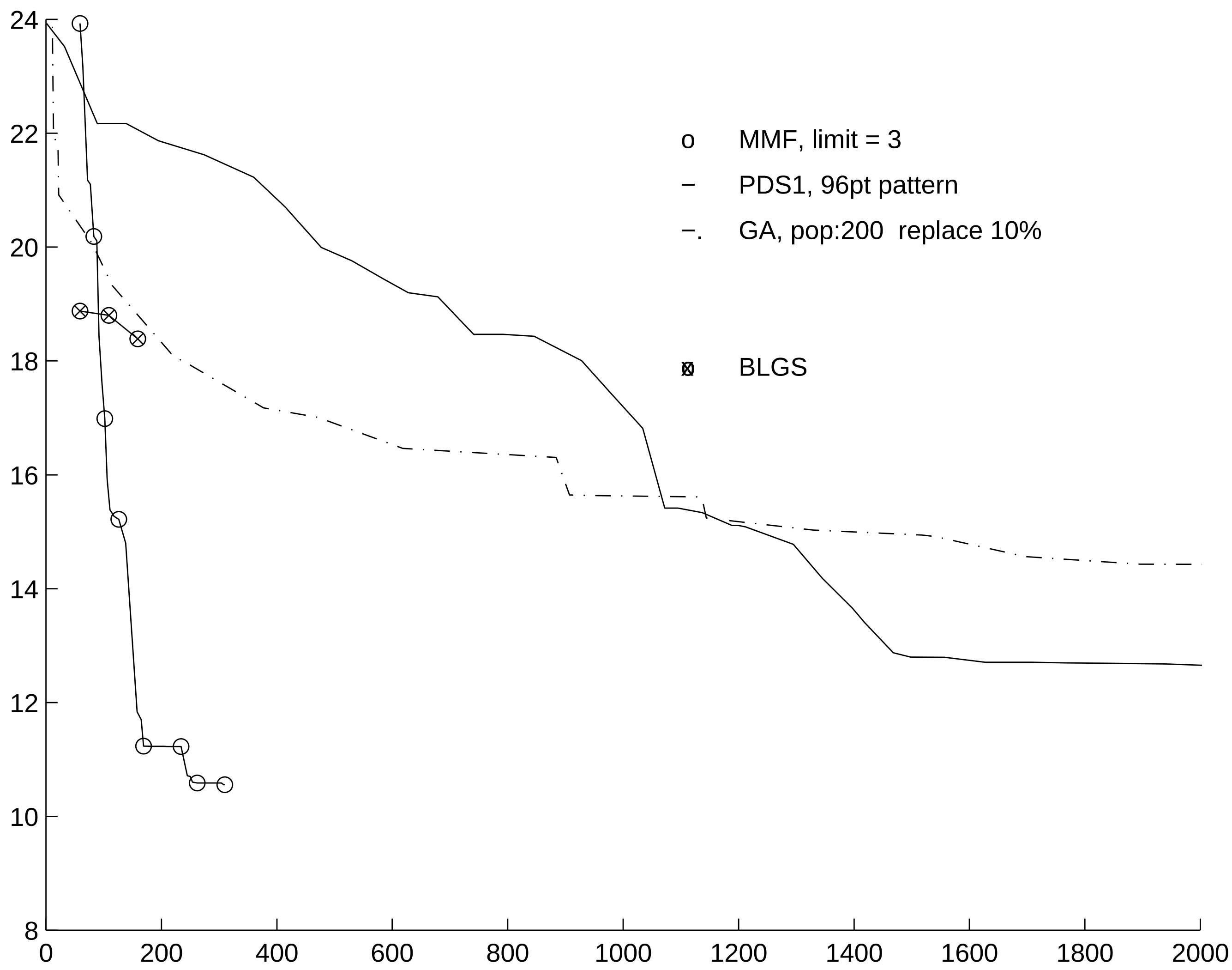
Boeing wing planform design

Infeasible baseline design.

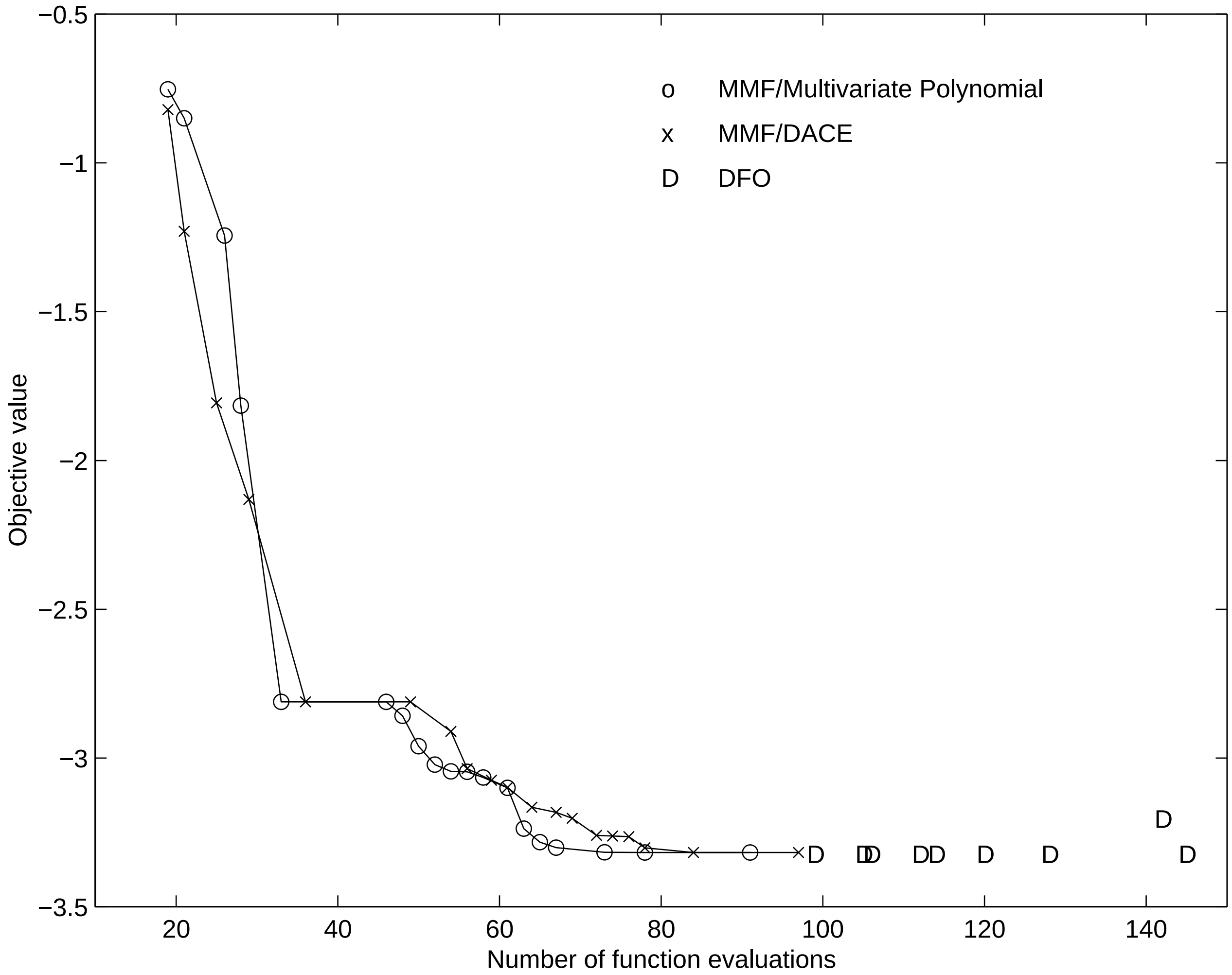
	n	# of ctrs	# of fevals
A	15	11	304
B	15	11	292



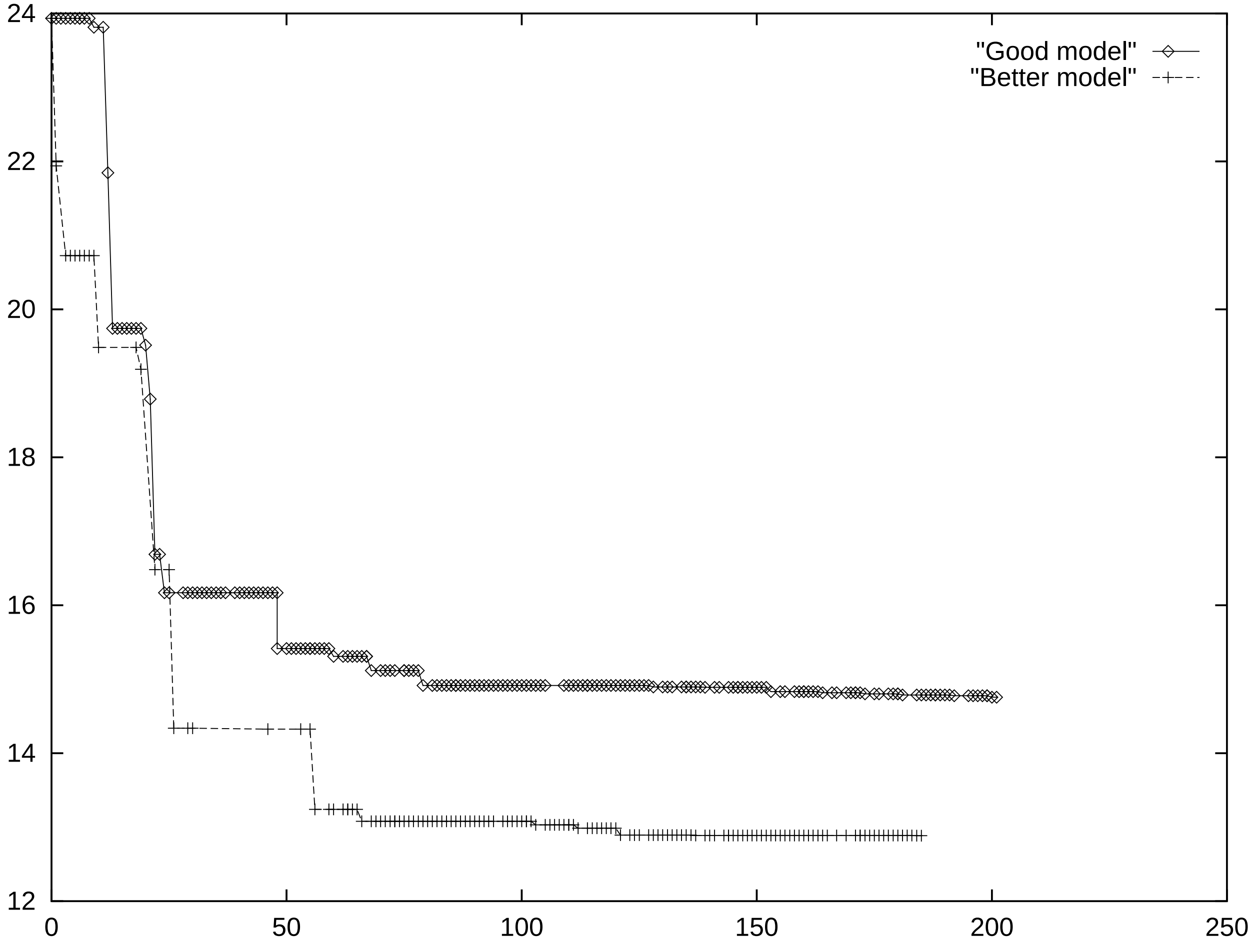
Comparisons on 31d helicopter example



Polynomial vs DACE surrogates



Recovering from a bad surrogate



Conclusions

- Clarke's tools simplify, shorten, and strengthen the pattern search analysis
- Filters handle nonlinear constraints without using derivatives, Lagrange multipliers or penalty parameters.
- FPS can be used directly on $f\&C$, but may use a lot of function values, especially near the optimum
- SMF seamlessly incorporates user heuristic SEARCHES, and POLLING provides robustness and rigor

Conclusions

- Software incorporates user heuristics into SEARCH:
  – MatLab by Mark Abramson and C++ by Gilles Couture
  Polling ensures robustness and provides rigor.

- SMF needs less accurate surrogates than designers want anyway, and it alleviates some dimensional effects. We need more adaptable surrogates

- Mixed variable filter pattern search algorithm is subject of Mark Abramson's dissertation.