SIOPT 2014

Formulations for Surrogate-Based Constrained Blackbox Optimization

Sébastien Le Digabel, École Polytechnique de Montréal
Bastien Talgorn, École Polytechnique de Montréal
Michael Kokkolaras, McGill University

2014–05–20

Presentation outline

Blackbox optimization and the MADS algorithm

Surrogates

Surrogate problem formulations

Results and discussion

Blackbox optimization and the MADS algorithm

Surrogates

Surrogate problem formulations

Results and discussion

Blackbox optimization problems

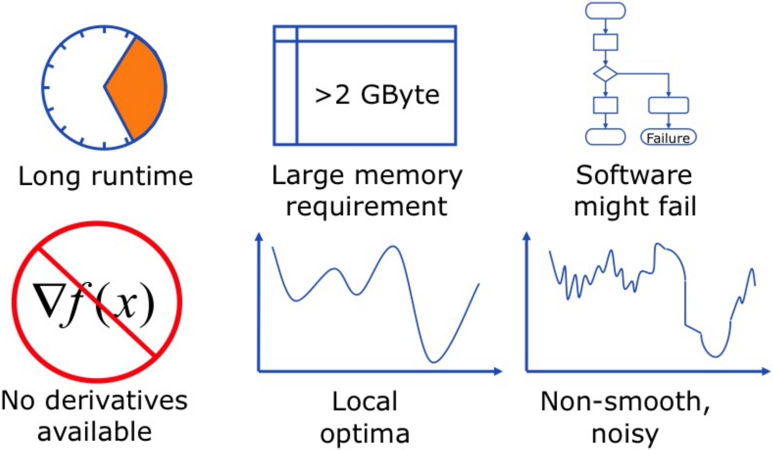
We consider the optimization problem:

$$\min_{x \in \Omega} \quad f(x)$$

with $\Omega = \left\{x \in \mathcal{X} : c_j(x) \leq 0, j \in \{1, 2, \ldots, m\}\right\} \subset \mathbb{R}^n$.

The evaluations of f and of the c_j functions are usually the result of a computer code (a blackbox).

Blackboxes as illustrated by J. Simonis [ISMP 2009]



Mesh Adaptive Direct Search (MADS)

- Audet and Dennis [SIOPT, 2006]
- Iterative algorithm that evaluates the blackbox at some trial points on a spatial discretization called the mesh.
- One iteration = search and poll.
- The search allows trial points generated anywhere on the mesh.
- The poll consists in generating a list of trial points constructed from poll directions. These directions grow dense.
- At the end of the iteration, the mesh size is reduced if no new iterate is found.
- Algorithm is backed by a convergence analysis based on the Clarke Calculus for nonsmooth functions.
- MADS is available via the `NOMAD` free software package at `www.gerad.ca/nomad`.

Blackbox optimization and the MADS algorithm

Surrogates

Surrogate problem formulations

Results and discussion

Static versus dynamic surrogates

- Static surrogate: A cheaper model defined a priori by the user. It is used as a blackbox too. Typically a simplified physics model. Variable precision is not yet considered.

- Dynamic surrogate: Model managed by the algorithm, based on past evaluations. It can be periodically updated.

In the remaining of this presentation, we focus on dynamic surrogates based on the `dynaTree` library.

General framework of MADS + surrogates

Only the additions to the MADS algorithm are reported.

[0] Initializations

[1] Iteration k

- **[1.1]** Model Search
  - select data points from cache
  - construct one model for each output (obj + cstrs)
  - select points for model improvement
  - optimize model to determine oracle points
  - *(→ the subproblem, or surrogate problem)*
  - project candidates to the mesh
  - evaluate candidates opportunistically
- **[1.2]** Poll (if the Search failed)
  - Model Ordering: use models to sort trial points

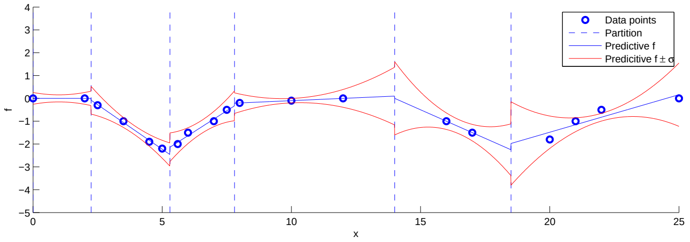
[2] Updates

The dynaTree library

- Developed for parameter-free regression by Taddy, Gramacy, and Polson [J. of the Am. Stat. Association, 2011].
- Based on Bayesian inference.
- R package available on CRAN.

dynaTree **outputs**

- The predictive mean $\hat{f}(x)$.
- The predictive standard deviation $\hat{\sigma}_f(x)$.



- The predictive cumulative distribution $\mathbb{P}[f(x) \leq f_0]$.
- Likewise for the constraints: $\hat{c}_j(x)$, $\hat{\sigma}_j(x)$, $\mathbb{P}[c_j(x) \leq c_0]$.

Surrogate problem formulations

- At each iteration, find the most promising candidates by solving the surrogate problem.

- The most basic formulation is:

$$\begin{cases} \min_{x \in \mathcal{X}} & \hat{f}(x) \\ \text{s.t.} & \hat{c}_j(x) \leq 0 \;\; \forall j \in J \;. \end{cases}$$

- We tested other formulations in the following submitted manuscript: B. Talgorn, S. Le Digabel, and M. Kokkolaras: *Problem Formulations for Simulation-based Design Optimization using Statistical Surrogates and Direct Search* [Optimization Online].

Diversification term

The standard deviation is added to the blackbox outputs:

$$(F\sigma) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ \text{s.t.} & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \;\; \forall j \in J \end{cases} .$$

$\lambda > 0 \Rightarrow$ diversification: focus more on exploration than on a particular region. The feasible domain is extended and poor values of $\hat{f}$ may be considered if $\hat{\sigma}_f$ is large.

Probability of feasibility of a point

One continuous `dynaTree` model is built for each constraint.

$$\mathbb{P}[x] = \mathbb{P}[x \text{ is feasible}] = \prod_{j \in J} \mathbb{P}[c_j(x) \leq 0] \; .$$

$\Rightarrow$ one scalar statistical measure to handle the constraints.

Chance constraint

$$(F\sigma P) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & \hat{f}(x) - \lambda . \hat{\sigma}_f(x) \\ \text{s.t.} & \mathbb{P}[x \text{ is feasible}] \geq p_c \ . \end{cases}$$

Diversification is possible (with parameter λ), but only candidates which are likely to be feasible are evaluated.

Generally, $p_c = \frac{1}{2}$, but it can be tailored according to the number of constraints.

Improvement

Improvement: $I(x) = \max\{f_{min} - f(x), 0\}$:

- M. Schonlau, D.R. Jones, and W.J. Welch [JOGO, 1998].
- f_{min}: current best known solution value.
- $I(x) > 0$ if x is better than the incumbent solution.
- $I(x) = 0$ otherwise.

Two statistical measurements:

- Probability of improvement: $PI(x) = \mathbb{P}\left[I(x) > 0\right]$.
- Expected improvement: $EI(x) = \mathbb{E}\left[I(x)\right]$.

What to do with constraints?

Probability of feasible improvement

Probabilities on:

- The objective: $PI(x)$
- The feasibility: $\mathbb{P}[x]$

$\Rightarrow$ Probability of Feasible Improvement:

$$PFI(x) = \mathbb{P}[x \text{ is feasible}] \times PI(x)$$

$$(PFI) \quad \left\{ \min_{x \in \mathcal{X}} \quad -PFI(x) \right.$$

Expected improvement subject to constraints

Maximization of the expected improvement under constraints:

$$(EI\sigma) \quad \begin{cases} \min\limits_{x \in \mathcal{X}} & -EI(x) - \lambda.\hat{\sigma}_f(x) \\ \text{s.t.} & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \ \ \forall j \in J \end{cases}$$

Expected feasible improvement (EFI)

Statistical measurement of

- The objective: $EI(x)$
- The feasibility: $\mathbb{P}[x]$

$\Rightarrow$ Expected Feasible Improvement:

$$EFI(x) = \mathbb{P}[x \text{ is feasible}] \times \mathbb{E}\left[I(x)\right]$$

The EFI represents what a candidate will statistically yield, in regard to the optimization problem.

Expected feasible improvement (EFI)

Maximization of the expected feasible improvement:

$$(EFI) \left\{ \min_{x \in \mathcal{X}} \ -EFI(x) \right.$$

EFI with a diversification term:

$$(EFI\sigma) \left\{ \min_{x \in \mathcal{X}} \ -EFI(x) - \lambda.\hat{\sigma}_f(x) \right.$$

Expected feasible improvement with μ

$$\mu(x) = 4\mathbb{P}[x](1 - \mathbb{P}[x])$$

$\mu(x)$ represents the uncertainty on the feasibility (variance of a Bernoulli law of probability $\mathbb{P}[x]$).

$$(EFI\mu) \quad \left\{ \min_{x \in \mathcal{X}} \quad -EFI(x) - \lambda.\hat{\sigma}_f(x).\mu(x) \right.$$

$\Rightarrow$ diversification favours a candidate with uncertainties on the objective AND the feasibility.

Crossed diversification terms

Search for candidates with:
promising objective (high EI) AND uncertain feasibility (high μ)
OR
uncertain objective (high $\hat{\sigma}_f$) AND promising feasibility (high $\mathbb{P}[x]$)

$\Rightarrow$ Diversification term : $EI(x)\mu(x) + \hat{\sigma}_f(x)\mathbb{P}[x]$

$$(EFIC) \quad \left\{ \min_{x \in \mathcal{X}} \; -EFI(x) - \lambda.\Big(EI(x).\mu(x) + \hat{\sigma}_f(x)\mathbb{P}[x]\Big) \right.$$

List of surrogate problem formulations

Formulation	Problem	λ
$(F\sigma)$	$\begin{cases} \min_{x\in\mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \end{cases}$	$\lambda \in \{0, \frac{1}{100}, \frac{1}{10}, 1\}$
$(F\sigma P)$	$\begin{cases} \min_{x\in\mathcal{X}} & \hat{f}(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \mathbb{P}[x \text{ is feasible}] \geq p_c \end{cases}$	
$(EI\sigma)$	$\begin{cases} \min_{x\in\mathcal{X}} & EI(x) - \lambda.\hat{\sigma}_f(x) \\ st: & \hat{c}_j(x) - \lambda.\hat{\sigma}_j(x) \leq 0 \end{cases}$	
$(EFI\sigma)$	$\min_{x\in\mathcal{X}} -EFI(x) - \lambda.\hat{\sigma}_f(x)$	
$(EFI\mu)$	$\min_{x\in\mathcal{X}} -EFI(x) - \lambda.\hat{\sigma}_f(x).\mu(x)$	
$(EFIC)$	$\min_{x\in\mathcal{X}} -EFI(x) - \lambda.(EI(x).\mu(x) + \hat{\sigma}_f(x)\mathbb{P}[x])$	
(PFI)	$\min_{x\in\mathcal{X}} -PFI(x)$	

Blackbox optimization and the MADS algorithm

Surrogates

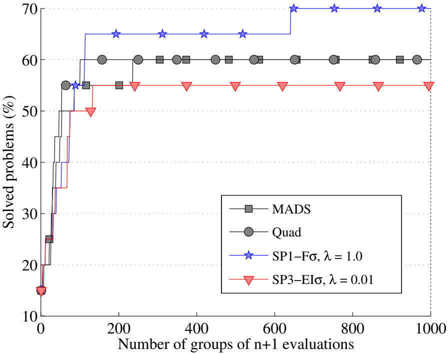
Surrogate problem formulations

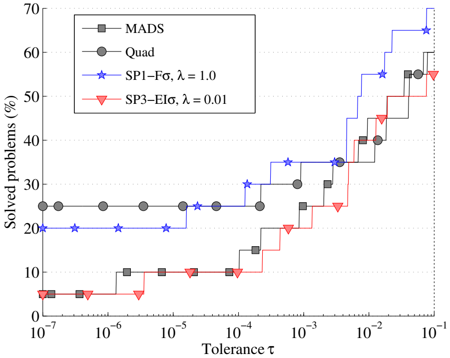
Results and discussion

Set of analytic problems

Name	n	m	Bounds	Smooth	f^*
G2	10	3	yes	no	-0.740466
MAD6	5	8	no	no	0.101831
PENTAGON	6	16	no	no	-1.85962
SNAKE	2	3	no	yes	0
HS24	2	4	no	yes	-1
HS34	3	3	yes	yes	-0.833795
HS36	3	2	yes	yes	-3300
HS37	3	3	yes	yes	-3455.51
HS64	3	2	no	yes	6299.94
HS66	3	3	yes	yes	0.532397
HS72	4	3	yes	yes	727.701
HS73	3	4	no	no	29.8944
HS86	5	11	no	yes	-32.2879
HS93	6	3	no	yes	135.075961
HS101	7	7	yes	yes	1809.76
HS102	7	7	yes	yes	911.88
HS103	7	7	yes	yes	543.67
HS104	8	7	yes	yes	4.02305
HS105	8	4	yes	yes	1136.36
HS114	9	7	yes	no	-1192.28

Profiles for the analytic problems (20 instances)





Left: *Data profile* with $\tau = 10^{-1}$. Right: *Performance profile* after $1000(n+1)$ evaluations.

Realistic MDO application

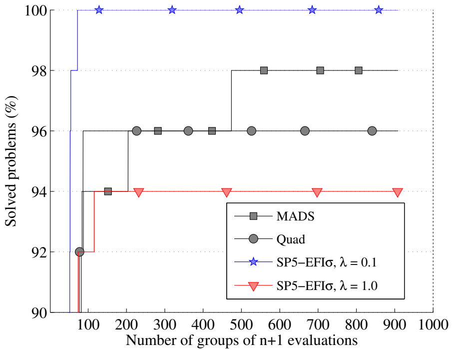
- NASA Aircraft Range problem with $n = m = 10$.
- Supersonic business jet with 3 disciplines: aerodynamics, structure, and propulsion.
- The problem can be summarized as

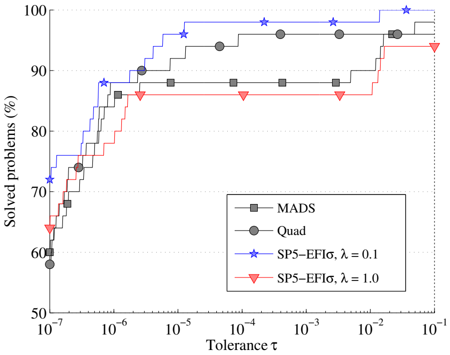
$$
\begin{array}{rl}
\text{max} & \text{aircraft range} \\
\text{subject to} & \text{normalized stress} \leq 1.09 \text{ (5 constraints)} \\
& \text{pressure gradient} \leq 1.04 \text{ Pa.m}^{-1} \\
& 0.5 \leq \text{eng. scale factor} \leq 1.5 \\
& \text{normalized engine temperature} \leq 1.02 \\
& \text{throttle setting} \leq \text{max throttle}
\end{array}
$$

Realistic MDO application: variables

Variables	Bounds Lower	Bounds Upper	x^*
Taper ratio	0.1	0.4	0.4
Wingbox cross-section	0.75	1.25	0.75
Skin friction coeff.	0.75	1.25	0.75
Throttle	0.1	1.0	0.156
Thickness/chord	0.01	0.09	0.06
Altitude	30000	60000	60000
Mach number	1.4	1.8	1.4
Aspect ratio	2.5	8.5	2.5
Wing sweep	40	70	70
Wing surface area	50	1500	1500

Profiles for the MDO problem (50 instances)





Left: *Data profile* with $\tau = 10^{-1}$. Right: *Performance profile* after 10,000 evaluations.

Current conclusions

- Set of 20 analytic problems: $(F\sigma)$ with $\lambda = 1$ performs better.
- For two realistic MDO applications, $(EI\sigma)$ with $\lambda = 1/100$ and 1, and $(EFI\sigma)$ with $\lambda = 1/10$, gave the best results.
- `NOMAD` with `dynaTree` should be available in a future `NOMAD` release.

- C. Audet and J. E. Dennis, Jr.: *Mesh adaptive direct search algorithms for constrained optimization.* SIAM Journal on Optimization, 2006.
- A.R. Conn and S. Le Digabel: *Use of quadratic models with mesh-adaptive direct search for constrained black box optimization.* Optimization Methods and Software, 2013.
- S. Kodiyalam: *Multidisciplinary aerospace systems optimization.* NASA/Lockheed Martin Technical Report, 2001.
- S. Le Digabel: *Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm.* ACM Transactions on Mathematical Software, 2011.
- M.A. Taddy, R.B. Gramacy, and N.G. Polson: *Dynamic trees for learning and design.* Journal of the American Statistical Association, 2011.
- B. Talgorn, S. Le Digabel, and M. Kokkolaras: *Problem Formulations for Simulation-based Design Optimization using Statistical Surrogates and Direct Search.* Technical report, Les Cahiers du GERAD G-2014-04.