Derivative Free Optimization Methods for Special Constrained Grey Box Optimization Problems

Pengcheng Xie
xpc@lsec.cc.ac.cn
Supervised by Prof. Ya-xiang Yuan

Institute of Computational Mathematics and Scientific/Engineering Computing
Academy of Mathematics and Systems Science
Chinese Academy of Sciences, China

Talk in Tsinghua University
April , 2021

Outline

- Introduction
- Derivative free methods
- Model update for GBO
- Penalty function and infeasible methods
- Projection and feasible methods
- Conclusion and future work

Introduction

What is derivative free optimization?

Introduction to derivative free optimization

- Most optimization methods rely on information about the derivatives of the problem.
- While in practical problems, information about derivatives may be difficult and expensive to obtain.

Notice that the problems in practice with expensive cost can be divided into two kinds, and we have such methods to solve them

- time-expensive $\Rightarrow$ parallel computing
- computation-expensive $\Rightarrow$ we can only use limited function value

derivative free optimization methods.

Introduction to derivative free optimization

Definition

Derivative-free optimization is also called black box optimization.

- The term black box means that we can give an estimate of $F(x)$ at any $x \in X$.
- It is difficult to obtain other valid information about the objective function $F(x)$.

When the estimation of the function value is very expensive, it becomes meaningful to carefully select the function value estimation points.

Overall goal: Obtain a sequence of iteration points $\{x_k\}$ to reach the optimal point with fewer iteration points.

Application

The derivative-free optimization algorithm has a deep impact on the development of many problems in various fields

- computer parameter tuning problems [Audet and Orban 2006]
- optimal design problems in engineering design [Booker et al. 1998]
- problems related to molecular geometry in biochemistry [Meza and Martinez 1994] and automatic error analysis [Higham 1993]
- dynamic pricing [Levina et al. 2009]

More specific examples include

- locating tsunami detection buoys
- lifting earthquake reconstruction, parameter determination [Zhang 2012; Audet and Hare 2017; Custódio, Scheinberg, and Nunes Vicente 2017]
- hyperparameter tuning of machine learning systems
- tuning learning rate and the number of hidden layers of deep neural networks [Golovin et al. 2017]

Constrained derivative-free optimization problems

Main methods to solve constrained derivative-free optimization problem containing

- active set-methods
- penalty methods
- barrier methods
- primal-dual interior methods
- SQP methods

Some effective solvers for derivative-free optimization problem with constraints

- BOBYQA (Bound Optimization BY Quadratic Approximation)
- COBYLA proposed by M. J. D. Powell based on software NEWUOA

Constrained derivative-free optimization problems

black box optimization with some information

$$\Downarrow$$

grey box optimization

Model-based trust region methods can be modified for grey box optimization, while the mesh adaptive direct search algorithm has already been modified by [Audet et al. 2020].

Optimization problem with orthogonality constraints

Optimization problem with orthogonality constraints, as the following shows, has widely usage, such as:

- electronic structure calculations in materials science [Yang, Meza, and Wang 2006].
- linear eigenvalue problems [Liu, Wen, and Zhang 2013]

The orthogonalization process requires about $O(n^2)$ magnitude computation and it would be more hard to compute when p is big due to the unscalability.

- Optimization on Lie group is also an interesting subject for mathematicians and scientists in Artificial Intelligence.
- Optimization on ellipsoidal surface may become a start of optimizing the problem with quadratic constraints.

NEWUOA

Derivative free optimization solver: NEWUOA

NEWUOA

- M.J.D Powell designed NEWOUA and provided a robust version of NEWUOA after the software UOBYQA.
- It achieved seeking the minimum of function F of several variables when derivatives are not available.
- For N-dimension problems, the main feature of NEWUOA is that quadratic models are updated using only $2N+1$ interpolation points with their conditions.

The key achieving this is to take up the remaining freedom by minimizing the Frobenius norm of the change to the second derivative matrix of the model.

The principle of NEWUOA

The NEWUOA method: Uses the minimum variation property to modify the Hessian $\nabla^2 Q$ of the existing model Q to obtain the new Hessian $\nabla^2 Q$.

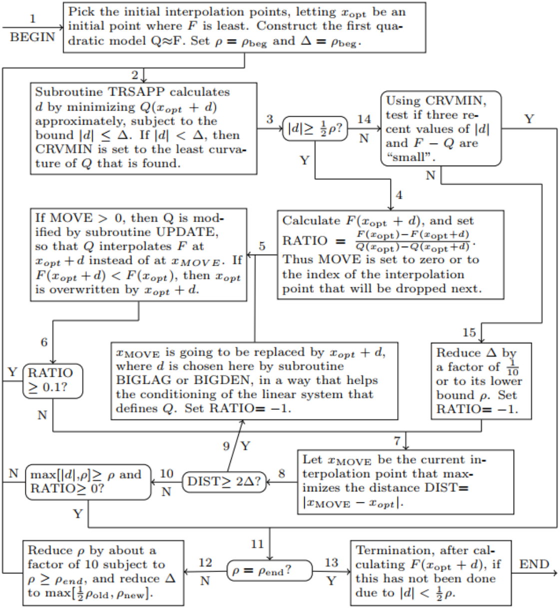
- The NEWUOA method uses less than $m = \frac{(n+1)(n+2)}{2}$ interpolation points.
- The method is to solve the quadratic problem

$$\min \left\| \nabla^2 Q_k - \nabla^2 Q_{k-1} \right\|_F^2, \\ \text{s.t. } Q_k(y) = f(y), y \in I_k.$$

  It draws on the idea of the minimum variation correction in the proposed quasi-Newton method, which is equivalent to symmetric broyden update without derivative.

- Powell recommended us to keep the number of interpolation points in the iterations as 2n+1, so that the computation of each iteration step is $O(n^2)$.

The procedure of the algorithm NEWUOA



Concepts and theorems of NEWUOA

Theorem

(Quadratic Interpolation Is Fully Quadratic) Let $x \in \mathbb{R}^n$ and $f \in \mathscr{C}^{2+}$ on $B_{\bar{\Delta}}(x)$ with constant K. Let $\mathbb{Y} = \{y^0, y^1, \ldots, y^m\} \subset \mathbb{R}^n$ be poised for quadratic interpolation with $y^0 = x$ and $\Delta = \overline{\text{diam}}(\mathbb{Y}) \le \bar{\Delta}$. Let Q_Y be the quadratic interpolation function of f over $\mathbb{Y}$. Then, there exists κ_f, κ_g, and κ_H (based on K, m, and on the "geometry of the interpolation set") such that

$$
\begin{aligned}
|f(y) - Q_Y(y)| &\le \kappa_f \Delta^3 \quad && \text{for all } y \in B_\Delta(x), \\
\|\nabla f(y) - \nabla Q_Y(y)\| &\le \kappa_g \Delta^2 \quad && \text{for all } y \in B_\Delta(x) \\
\text{and } \left\|\nabla^2 f(y) - \nabla^2 Q_Y(y)\right\| &\le \kappa_H \Delta \quad && \text{for all } y \in B_\Delta(x).
\end{aligned}
$$

We define the minimum Frobenius norm model through the solution to the following quadratic optimization problem

$$\begin{aligned} \min_{\alpha_0,\alpha,H} \quad & \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} h_{i,j}^2 \\ & \alpha_0+\alpha^\top y^i+\frac{1}{2}\left(y^i\right)^\top H y^i = f\left(y^i\right) \quad \text{for} \quad i=0,1,2,\ldots m, \\ & H=H^\top. \end{aligned}$$

Definition (Poised for Minimum Frobenius Norm Modelling)

The set $\mathbb{Y}=\left\{y^0,y^1,\ldots,y^m\right\}\subset\mathbb{R}^n$ with $n<m<\frac{1}{2}(n+1)(n+2)-1$, is poised for minimum Frobenius norm modelling if problem (15) has a unique solution (α_0,α,H).

Concepts and theorems of NEWUOA

Definition ((Minimum Frobenius Norm Model Function)

Let $f : \mathbb{R}^n \mapsto \mathbb{R}$, and $\mathbb{Y} = \{y^0, y^1, \ldots, y^m\} \subset \mathbb{R}^n, n < m < \frac{1}{2}(n+1)(n+2) - 1$, be poised for minimum Frobenius norm modelling. Then the minimum Frobenius norm model function of f over $\mathbb{Y}$ is

$$M_Y(x) := \alpha_0 + \alpha^\top x + \frac{1}{2} x^\top H x$$

where (α_0, α, H) is the unique solution to Equation (15).

Theorem (Minimum Frobenius Norm Modelling Is Fully Linear)

Let $x \in \mathbb{R}^n$ and $f \in \mathscr{C}^{1+}$ on $B_{\bar{\Delta}}(x)$ with constant K. Let $\mathbb{Y} = \{y^0, y^1, \ldots, y^m\} \subset \mathbb{R}^n$ be poised for minimum Frobenius norm modelling with $y^0 = x$ and $\Delta = \overline{\mathrm{diam}}(\mathbb{Y}) \leq \bar{\Delta}$. Let M_Y be the minimum Frobenius norm model function of f over $\mathbb{Y}$. Then, there exists κ_f and κ_g such that

$$\begin{aligned} \|f(y) - M_Y(y)\| &\leq \kappa_f \Delta^2 \quad \textit{for all } y \in B_\Delta(x) \\ \textit{and } \|\nabla f(y) - \nabla M_Y(y)\| &\leq \kappa_g \Delta \quad \textit{for all } y \in B_\Delta(x). \end{aligned}$$

Model update for GBO

Grey box optimization

Definition (Grey-box problem)

Problems with partially known information.

In short, grey-box problems have more information available than black-box problems.

We note that the objective function utilized for interpolation at the old interpolation point $\{x_1, \cdots, x_m\}$ is $f(x) + g(x,k)$,

- $f(x) \Rightarrow$ "black box" function part
- $g(x,k) \Rightarrow$ "white" function part, and k is the number of test points used (iterations)

A quadratic model of the objective function $f(x) + g(x,k)$ satisfying the following interpolation

$$Q_{\text{old}}(x_i) = f(x_i) + g(x_i, k), \quad i = 1, \cdots, m,$$

where m is the number of interpolation points and $m < \frac{1}{2}(n+1)(n+2)$.

Grey box optimization

We want to obtain an update formula that satisfies the following condition

$$Q_{\text{new}}(x_i) = f(x_i) + g(x_i, k+1), \quad i = 1, \cdots, t-1, \text{new}, t+1, \cdots, m,$$

where x_{new} will be stored using the position of the discarded interpolated point x_t, i.e., the newest point added is stored at the position of the t th point. Denote $D(x) = Q_{\text{new}}(x) - Q_{\text{old}}(x)$.

We want $D(x)$ to satisfy the following condition

- If $i \neq t$, then

$$\begin{aligned} D(x_i) &= Q_{\text{new}}(x_i) - Q_{\text{old}}(x_i) \\ &= f(x_i) + g(x_1, k+1) - (f(x_i) + g(x_i, k)) \\ &= g(x_i, k+1) - g(x_i, k). \end{aligned}$$

- If $i = t = \text{new}$, then

$$\begin{aligned} D(x_{\text{new}}) &= Q_{\text{new}}(x_{\text{new}}) - Q_{\text{old}}(x_{\text{new}}) \\ &= f(x_{\text{new}}) + g(x_{\text{new}}, k+1) - Q_{\text{old}}(x_{\text{new}}). \end{aligned}$$

Update method

Considering that the number of interpolation points we have selected is $m < \frac{1}{2}(n+1)(n+2)$, we need to solve the following problem

$$
\begin{aligned}
\min_{D(x)} \quad & \|\nabla^2 D\|_F \\
\text{s.t.} \quad & D(x_i) = g\left(x_i, k+1\right) - g\left(x_i, k\right), \quad i = 1, \cdots, t-1, t+1, \cdots, m, \\
& D(x_i) = f\left(x_{\text{new}}\right) + g\left(x_{\text{new}}, k+1\right) - Q_{\text{old}}\left(x_{\text{new}}\right), \; i = t = \text{new}.
\end{aligned}
$$

While we let λ_j, $j = 1, 2, \cdots, m$ serve as Lagrange multipliers for the KKT conditions of this problem, which, as Powell pointed out in [Powell 2004b], have the following properties

$$
\sum_{j=1}^{m} \lambda_j = 0, \quad \sum_{j=1}^{m} \lambda_j \left(x_j - x_0\right) = 0. \tag{1}
$$

and the Hessian matrix of the quadratic polynomial $D(x)$ possesses the following form.

$$
\nabla^2 D = \sum_{j=1}^{m} \lambda_j x_j x_j^\top = \sum_{j=1}^{m} \lambda_j \left(x_j - x_0\right)\left(x_j - x_0\right)^\top. \tag{2}
$$

Update method

The second equal sign of the above expression (2) is obtained from the expression (1), and this form of $\nabla^2 D$ allows $D(x)$ to be written in the following functional form

$$D(x) = c + (x - x_0)^\top g + \frac{1}{2}\sum_{j=1}^{m} \lambda_j \{(x - x_0)^\top (x_j - x_0)\}^2, \quad x \in \mathbb{R}^n.$$

By determining the values of the parameters $c \in \mathbb{R}^n, g \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^m$, we can uniquely determine $D(x)$ and thus update to obtain $Q_{\text{new}}(x)$.

From the conditions given earlier, we can obtain the following system of linear equations

$$\begin{pmatrix} A & X^\top \\ X & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \\ g \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix},$$

Update method

where the elements of the matrix A are

$$A_{ij} = \frac{1}{2}\left\{(x_i - x_0)^\top (x_j - x_0)\right\}^2, \quad 1 \leq i,j \leq n.$$

the elements of the matrix H are

$$X = \begin{pmatrix} 1 & \cdots & 1 \\ x_1 - x_0 & \cdots & x_m - x_0 \end{pmatrix}.$$

The vector r has the form

$$r = \begin{pmatrix} g(x_1, k+1) - g(x_1, k) \\ \vdots \\ g(x_{t-1}, k+1) - g(x_{t-1}, k) \\ f(x_{\text{new}}) + g(x_{\text{new}}, k+1) - Q_{\text{old}}(x_{\text{new}}) \\ g(x_{t+1}, k+1) - g(x_{t+1}, k) \\ \vdots \\ g(x_m, k+1) - g(x_m, k) \end{pmatrix}. \tag{3}$$

Update method

Let W and H be the following matrices:

$$W = \begin{pmatrix} A & X^\top \\ X & 0 \end{pmatrix}, \quad H = W^{-1} = \begin{pmatrix} \Omega & \Xi^\top \\ \Xi & \gamma \end{pmatrix}.$$

We know that we can derive the elements of the matrix W directly from the vector $x_i, i = 1, 2, \cdots, m$. In updating the matrix H, the following formula given by Powell in [Powell 2004a] can be used directly.

$$\begin{aligned} H_{\text{new}} =& H + \sigma^{-1} \Big[\alpha \left(e_t - Hw\right)\left(e_t - Hw\right)^T - \beta H e_t e_t^\top H \\ & + \tau \left\{ H e_t \left(e_t - Hw^\top\right) + \left(e_t - Hw\right) e_t^\top H \right\} \Big], \end{aligned} \tag{4}$$

where the expressions for each parameter are.

$$\begin{aligned} \alpha =& e_t^T H e_t, \quad \beta = \frac{1}{2}\|x_{\text{new}} - x_0\|^4 - w^T H w, \\ \tau =& e_t^T H w, \quad \sigma = \alpha\beta + \tau^2. \end{aligned}$$

Grey box optimization

Finally, the update formula for $D(x)$ is

$$D(x) = c + (x - x_0)^\top g + \frac{1}{2} \sum_{j=1}^{m} \lambda_j \left\{ (x - x_0)^\top (x_j - x_0) \right\}^2, x \in \mathbb{R}^n,$$

the parameter of it is the vector

$$\begin{pmatrix} \lambda \\ c \\ g \end{pmatrix} = H \begin{pmatrix} r \\ 0 \end{pmatrix},$$

where the matrix H is obtained by updating the formula (4) and r is of the form (3).

Assumption

Assumption 1. Function evaluation $g(x,k)$ at the k-th step has noises, which means $g(x,k)$ is not exactly $g(x)$. There exists a sequence of $\{g(x,k)\}_{k=1}^{\infty}$, satisfies that $\lim_{k\to\infty}\|g(x,k)-g(x)\|=0$.

Assumption 2. The change of the function evaluation of the continuous step is not dramatic. In detail, the assumption is that there exists an upper bound $\varepsilon(k)$ at the k-th evaluation, which satisfies that

$$\|g(x,k+1)-g(x,k)\|_{\infty}=0<\varepsilon(k).$$

and $\lim_{k\to\infty}\varepsilon(k)=0$.

The challenges

It would be reckless to try to simply and directly transform the parameters, penalty factors, in the penalty function.

Why?

This is because:

- Each test point that obtains a function value contributes to a smaller and smaller function value for the set of interpolated points
- A model point helps the model obtained by interpolation to better describe the original objective function locally.

Penalty function

and infeasible methods

Derivative-free optimization algorithm for ellipsoidal constrained problem

For derivative-free optimization problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & F(x) \\ \text{s.t.} \quad & x^T A x + 1 = 0, \end{aligned} \tag{5}$$

where $A = \mathrm{diag}\{a_1, a_2, \cdots, a_n\}, a_i > 0, i = 1, \cdots, n$.

We have developed two improvements of the NEWUOA algorithm

- Feasible-free method
- Feasible method

Infeasible method and feasible method

One is feasible-free method, which does not force each iteration point to stay in the feasible domain. We add penalty function to the objective function

- Courant penalty function
- augmented Lagrangian penalty function

The other method is feasible method

- The iterative points calculated by the truncated conjugate gradient method.
- The interpolation points in the set of interpolation points after projection.

We also designed projection method.

Courant penalty method

Definition

We know that for situation in $\mathbb{R}^2$, the 2-norm is

$$\|x\|_2 = (x_1^2 + x_2^2)^{\frac{1}{2}}, \quad x = (x_1, x_2).$$

Now we want to define that in ellipsoidal constraint, it can be defined like

$$\|x\|_2 = (x^T A x)^{\frac{1}{2}}, \quad x = (x_1, \cdots, x_n).$$

For simplicity of expression, the mark $\|x\|_2$ denotes the 2-norm in ellipsoidal constraint in the content below, unless we state otherwise.

The purpose of setting penalty function: Force the iteration points fall on the feasible field as much as possible.

- If the penalty factor σ is too small $\Rightarrow$ the influence of the penalty item would be insignificant.
- If the penalty factor σ is too large $\Rightarrow$some points with low function value which is just about the boundary of the feasible field, would be dropped by mistake.

Courant penalty method

For the ellipsoidal constraint

$$x^T A x + 1 = 0, \tag{6}$$

where $A = \mathrm{diag}\{\mathrm{a}_1, \mathrm{a}_2, \cdots, \mathrm{a}_\mathrm{n}\}$, and where we denote the constraint as

$$c(x) = x^T A x + 1. \tag{7}$$

The corresponding Courant penalty function is

$$P(x, \sigma) = F(x) + \sigma (c(x))^2, \tag{8}$$

where $c(x)$ is definited in (7).

What we can use: The function value at interpolation points.

Difficulties: How to select a suitable penalty parameter σ which is systematical and widely applicable.

Courant penalty function

We take the variable penalty factor Courant penalty function as an example to introduce, in essence, any other penalty function involving variable coefficients is the same.

Letting σ_k for the kth iteration be σ_{old} and σ_{k+1} for the $k+1$th iteration be σ_{new}, we know that

$$\begin{cases} Q_{\text{old}}(x_i) = f(x_i) + \sigma_{\text{old}}(c(x_i))^2, & i = 1, \cdots, m, \\ Q_{\text{new}}(x_i) = f(x_i) + \sigma_{\text{new}}(c(x_i))^2, & i = 1, \cdots, t-1, \text{new}, t+1, \cdots, m. \end{cases}$$

Courant penalty function

Algorithm 1 Courant penalty method with constant parameter with adjustable penalty factor

1: Given $x_1 \in \mathbb{R}^n, \sigma_1 > 0, k := 1, \varepsilon \geq 0$.

2: Use initial value x_k and NEWUOA algorithm to solve

$$\min_x F(x) + \sigma_k \|c(x)\|_2^2, \tag{9}$$

get the solution $x(\sigma_k)$.

3: If $\|c(x(\sigma_k))\|_2 \leq \varepsilon$, then stop; $x_{k+1} = x(\sigma_k), \sigma_{k+1} = 10\sigma_k$, $k = k+1$; Go to step 2.

The new update method

Letting σ_k for the kth iteration be σ_{old} and σ_{k+1} for the $k+1$th iteration be σ_{new}, we know that

$$\begin{cases} Q_{\text{old}}(x_i) = f(x_i) + \sigma_{\text{old}}(c(x_i))^2, & i = 1, \cdots, m, \\ Q_{\text{new}}(x_i) = f(x_i) + \sigma_{\text{new}}(c(x_i))^2, & i = 1, \cdots, t-1, \text{new}, t+1, \cdots, m. \end{cases}$$

Construct a quadratic function $D(x)$ satisfying the following conditions such that $\|\nabla^2 D\|_F$ is minimum under the constraint.

$$\begin{cases} D(x_i) = (\sigma_{\text{new}} - \sigma_{\text{old}})(c(x_i))^2, & i = 1, \cdots, t-1, t+1, \cdots, m, \\ D(x_i) = f(x_{\text{new}}) + \sigma_{\text{new}}(c(x_{\text{new}}))^2 - Q_{\text{old}}(x_{\text{new}}), & i = t = \text{new}. \end{cases}$$

In the following we can determine the values of the parameters c, g, λ using the update formula for the matrix H given earlier and $(r, 0)^T$ satisfying the following equation.

$$r_i = \{f(x_i) + \sigma_{\text{new}}(c(x_i)^2 - Q_{\text{old}}(x_i)\} \cdot \delta_{it} + \{(\sigma_{\text{new}} - \sigma_{\text{old}})(c(x_{new}))^2\} \cdot (1 - \delta_{it}),$$

where $r = (r_1, \cdots, r_m)$.

Projection and feasible methods

Projection

Proposition (Standard stereographic projection)

The function which sends a point $x \in S^n \backslash p \subset \mathbb{R}^{n+1}$ *to the intersection of the line through* p *and* x *with the equatorial hyperplane is a homeomorphism which is given in terms of ambient coordinates by*

$$\begin{aligned} \mathbb{R}^{n+1} \supset \quad S^n \backslash (1,0,\cdots,0) \quad &\longrightarrow \quad \mathbb{R}^n \subset \mathbb{R}^{n+1} \\ (x_1, x_2, \cdots, x_{n+1}) &\longmapsto \frac{1}{1-x_1}(0, x_2, \cdots, x_{n+1}). \end{aligned}$$

In particular, therefore also an inverse function to the stereographic projection exists and is a rational function, hence continuous. So we have exhibited a homeomorphism as required.

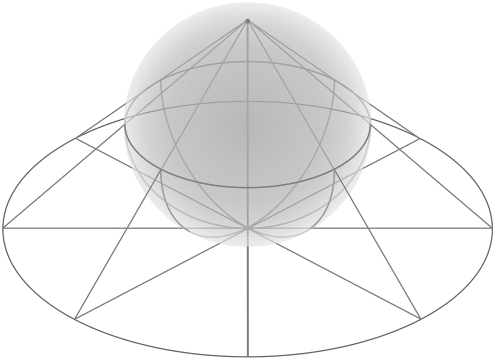


Figure 1: Stereographic Projection

Projection

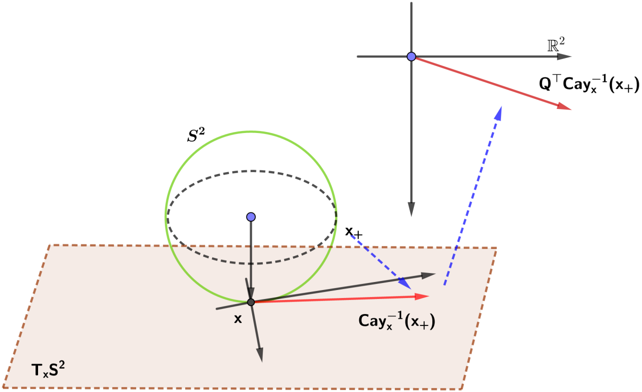


Figure 2: Cayley Transform

Projection

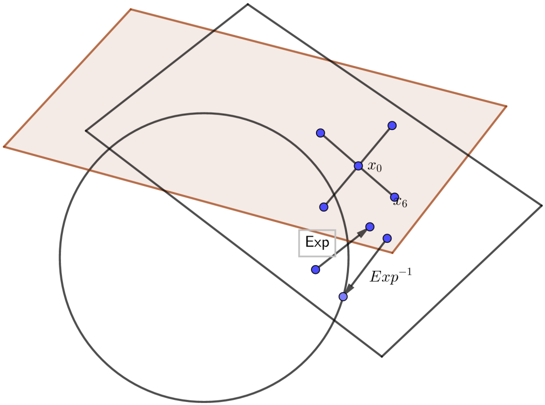


Figure 3: Exponential Projection

Profile

Definition

$$f_{\text{acc}}^N = \frac{f\left(x^N\right) - f\left(x^0\right)}{f\left(x^*\right) - f\left(x^0\right)} \in [0,1]. \tag{10}$$

where x^N is the best point found by the algorithm after N function evaluations, x^0 is the initial point, and x^* is the best known solution.

Given the "tolerance" $\tau \in [0,1]$, we definite

Definition

$$T_{a,p} = \begin{cases} 1, & \text{if} \quad f_{\text{acc}}^N \geq 1-\tau \quad \text{for some } N, \\ 0, & \text{otherwise.} \end{cases}$$

$$N_{a,p} = \min\{n \in N, f_{\text{acc}}^N \geq 1-\tau\}.$$

$$r_{a,p} = \begin{cases} \frac{N_{a,p}}{\min\left\{N_{\tilde{a},p} : \tilde{a} \in \mathscr{A}, T_{\tilde{a},p}=1\right\}} & \text{if} \quad T_{a,p} = 1, \\ \infty, & \text{if} \quad T_{a,p} = 0. \end{cases}$$

Profile

Performance profile:

$$\rho_a : [1, \infty) \mapsto [0,1],$$
$$\rho_a(\alpha) = \frac{1}{|\mathscr{P}|} \left| \{ p \in \mathscr{P} : r_{a,p} \leq \alpha \} \right|.$$

where $|\cdot|$ counts the number of elements in a set.

Data profile:

$$d_a(k) = \frac{1}{|\mathscr{P}|} \left| \{ p \in \mathscr{P} : N_{a,p} \leq k\,(n_p + 1)\, T_{a,p} \} \right|,$$

where the n_p means the dimension of the problem p.

Accuracy profile:

$$-\log_{10}\left(1 - f_{\text{acc}}^{N_{a,p}^{tot}}\right) \geq d \quad \Leftrightarrow \quad f_{\text{acc}}^{N_{a,p}^{tot}} \geq 1 - 10^{-d},$$
$$r_a(d) = \frac{1}{|\mathscr{P}|} \left| \left\{ p \in \mathscr{P} : -\log_{10}\left(1 - f_{\text{acc}}^{N_{a,p}^{tot}}\right) \geq d \right\} \right|.$$

Profile

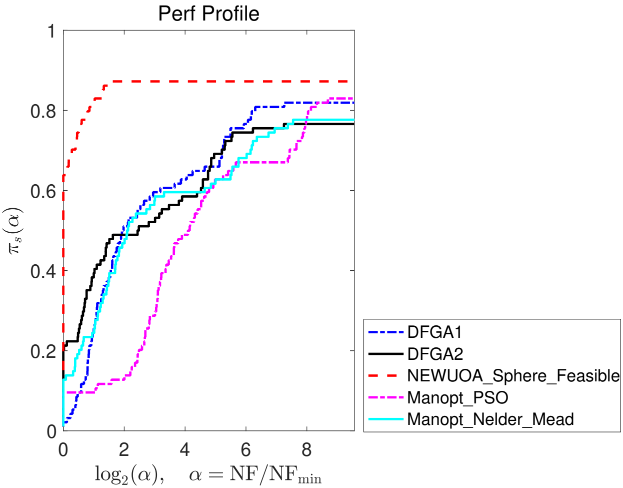


Figure 4: Comparison of the algorithms: Performance profile

Profile

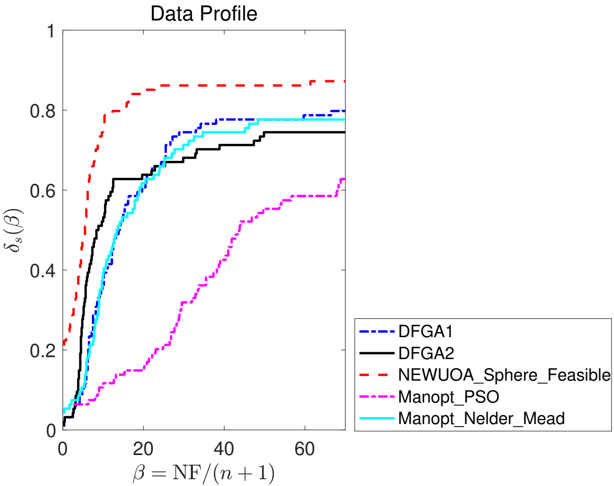


Figure 5: Comparison of the algorithms: Data profile

Profile

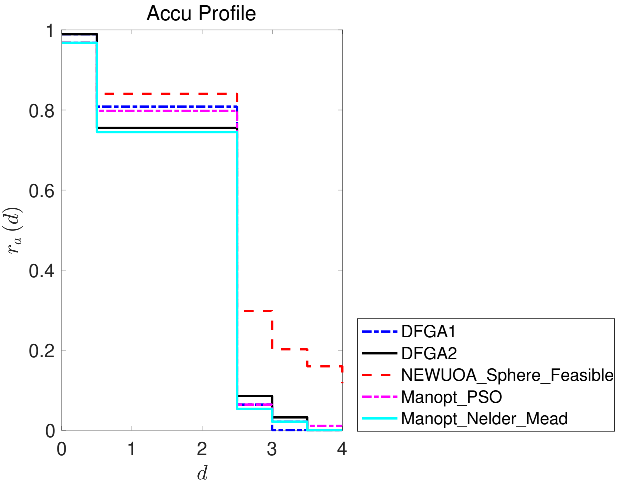


Figure 6: Comparison of the algorithms: Accuracy profile

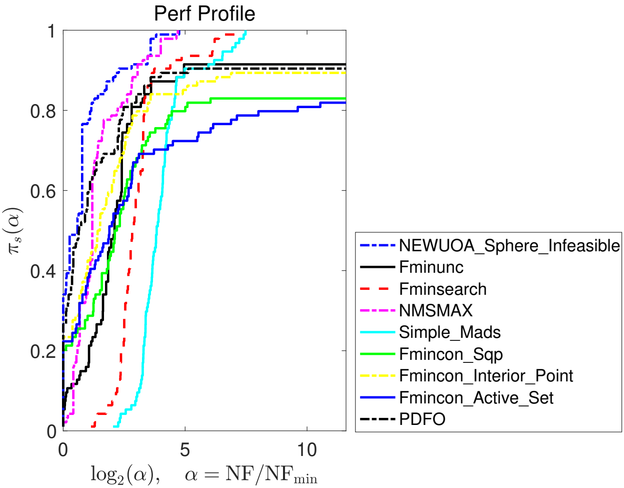


Figure 7: Comparison of the algorithms: Performance profile

Profile

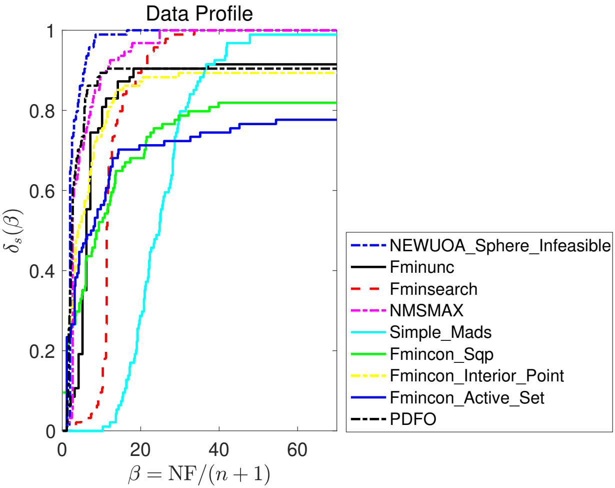


Figure 8: Comparison of the algorithms: Data profile

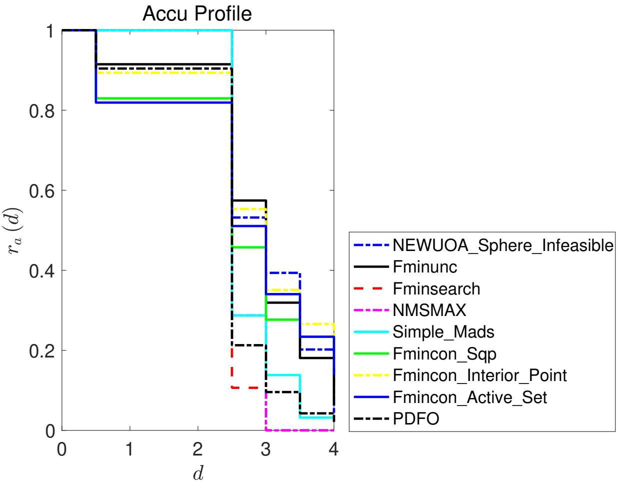


Figure 9: Comparison of the algorithms: Accuracy profile

Conclusion and future work

Conclusion

We have make such improvements in NEWUOA algorithm

- Penalty function
- Projection contraction technique

For grey-box optimization problems with special forms, we give a new model update method.

This model update method well preserves the number of interpolation points, (iteration points) which is still of order $O(n)$.

Future work

In the future, we would pay attention to

- The combination of model-based derivative-free methods and optimization with inexact gradient and function.
- Apply the projection method more deeply in NEWUOA.
- Apply the GBO model to solve more problems such as constrained optimization using penalty functions with variable penalty factors.
- Make more full use of the information we get in GBO.

And we will move on searching for better derivative-free optimization algorithms.

Future work

In the future, we would pay attention to

- The combination of model-based derivative-free methods and optimization with inexact gradient and function.
- Apply the projection method more deeply in NEWUOA.
- Apply the GBO model to solve more problems such as constrained optimization using penalty functions with variable penalty factors.
- Make more full use of the information we get in GBO.

And we will move on searching for better derivative-free optimization algorithms.

Thank you!