COBYQA

A derivative-free trust-region SQP method for nonlinearly constrained optimization

Tom M. Ragonneau Zaikun Zhang

ICIAM 2023, Tokyo, Japan, 2023

Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong, China

This work was partially supported by the Hong Kong PhD Fellowship Scheme.

General context

We design a method named COBYQA for solving

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & g(x) \le 0, \; h(x) = 0, \\ & l \le x \le u, \end{aligned}$$

when derivatives of f, g, and h are unavailable.

General context

We design a method named COBYQA for solving

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(x) \\
\text{s.t.} \quad & g(x) \le 0, \ \cancel{h(x) = 0}, \\
& l \le x \le u,
\end{aligned}
$$

when derivatives of f, g, and h are unavailable.

- We omit the equality constraints for simplicity.
- COBYQA aims at being a successor to COBYLA (Powell 1994).
- We implement COBYQA into a Python solver.
- The bound constraints are unrelaxable:
  - They often represent inalienable restrictions.
  - f, g, or h may not be well-defined outside the bounds.

Table of contents

General framework of COBYQA

A derivative-free trust-region SQP method

COBYQA iteratively solves the trust-region SQP subproblem

$$\begin{aligned}
\min_{s \in \mathbb{R}^n} \quad & \nabla f(x_k)^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k) s \\
\text{s.t.} \quad & g(x_k) + \nabla g(x_k) s \le 0, \\
& l \le x_k + s \le u, \\
& \|s\| \le \Delta_k,
\end{aligned}$$

with $\mathcal{L}(x, \lambda) = f(x) + \lambda^\mathsf{T} g(x)$.

A derivative-free trust-region SQP method

COBYQA iteratively solves the trust-region SQP subproblem

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & \nabla \hat{f}_k(x_k)^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2_{xx} \widehat{\mathcal{L}}_k(x_k, \lambda_k) s \\ \text{s.t.} \quad & \hat{g}_k(x_k) + \nabla \hat{g}_k(x_k) s \le 0, \\ & l \le x_k + s \le u, \\ & \|s\| \le \Delta_k, \end{aligned}$$

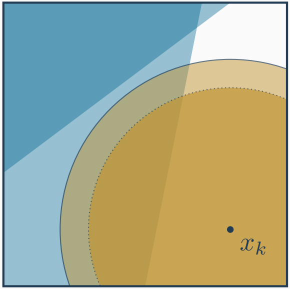
with $\widehat{\mathcal{L}}_k(x, \lambda) = \hat{f}_k(x) + \lambda^\mathsf{T} \hat{g}_k(x)$, given some models $\hat{f}_k$ and $\hat{g}_k$.

- We only require an approximate solution s_k.
- The solution must satisfy $l \le x_k + s_k \le u$.
- The subproblem may be infeasible. What is a solution?

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



Trust region

Reduced trust region

Linear constraints

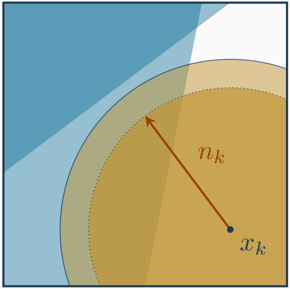
Standard approach[1] vs. new one.

[1] See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



Trust region

Reduced trust region

Linear constraints

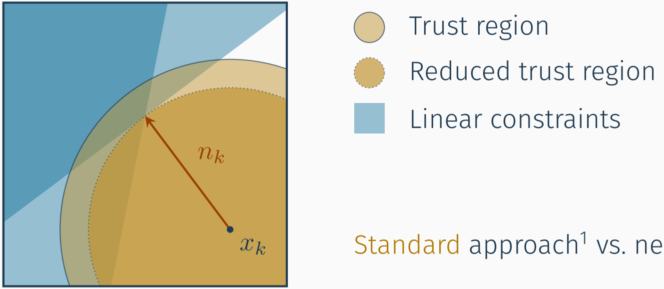
Standard approach[1] vs. new one.

[1]See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



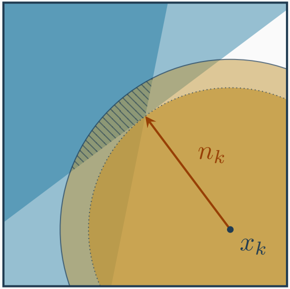
Standard approach[1] vs. new one.

[1] See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



- Trust region
- Reduced trust region
- Linear constraints
- Feasible region for t_k

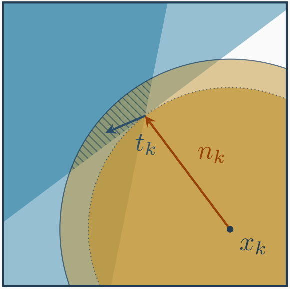
Standard approach[1] vs. new one.

---

[1] See Conn, Gould, and Toint (2000, §15.4.4).

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



Trust region

Reduced trust region

Linear constraints

Feasible region for t_k

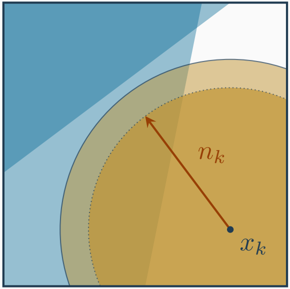
Standard approach[1] vs. new one.

[1]See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



Trust region

Reduced trust region

Linear constraints

Feasible region for t_k

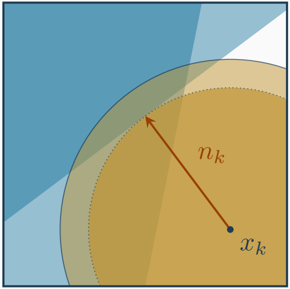
Standard approach[1] vs. new one.

[1] See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



Trust region

Reduced trust region

Linear constraints

Feasible region for t_k

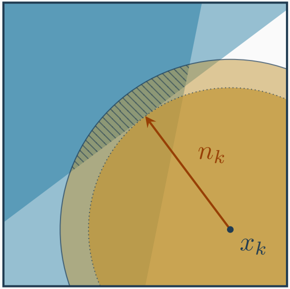
Standard approach[1] vs. new one.

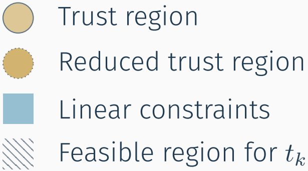
[1] See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.





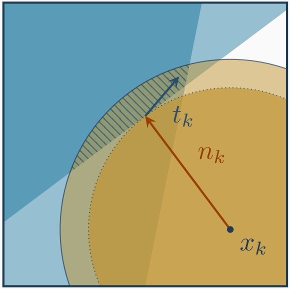
Standard approach[1] vs. new one.

[1]See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach

We compute $s_k = n_k + t_k$, where

- the normal step n_k reduces the (possible) constraint violation, and
- the tangential step t_k reduces the quadratic objective function.



- Trust region
- Reduced trust region
- Linear constraints
- Feasible region for t_k

Standard approach[1] vs. new one.

The feasible region for t_k is wider in the new approach.

[1]See Conn, Gould, and Toint (2000, §15.4.4).

A new Byrd-Omojokun approach (cont'd)

Standard approach:

- The normal step n_k solves approximately (for some $\zeta < 1$)

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & \left\| \left[\hat{g}_k(x_k) + \nabla \hat{g}_k(x_k) s \right]^+ \right\| \\ \text{s.t.} \quad & l \leq x_k + s \leq u, \\ & \|s\| \leq \zeta \Delta_k. \end{aligned}$$

- The tangential step t_k solves approximately

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & \left[\nabla \hat{f}_k(x_k) + \nabla^2_{xx} \widehat{\mathcal{L}}_k(x_k, \lambda_k) n_k \right]^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2_{xx} \widehat{\mathcal{L}}_k(x_k, \lambda_k) s \\ \text{s.t.} \quad & \nabla \hat{g}_k(x_k) s \leq 0, \\ & l \leq x_k + n_k + s \leq u, \\ & \|n_k + s\| \leq \Delta_k. \end{aligned}$$

A new Byrd-Omojokun approach (cont'd)

New approach:

- The normal step n_k solves approximately (for some $\zeta < 1$)

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & \left\| \left[\hat{g}_k(x_k) + \nabla \hat{g}_k(x_k) s \right]^+ \right\| \\ \text{s.t.} \quad & l \leq x_k + s \leq u, \\ & \|s\| \leq \zeta \Delta_k. \end{aligned}$$

- The tangential step t_k solves approximately

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & \left[\nabla \hat{f}_k(x_k) + \nabla^2_{xx} \widehat{\mathcal{L}}_k(x_k, \lambda_k) n_k \right]^\mathsf{T} s + \frac{1}{2} s^\mathsf{T} \nabla^2_{xx} \widehat{\mathcal{L}}_k(x_k, \lambda_k) s \\ \text{s.t.} \quad & \nabla \hat{g}_k(x_k) s \leq \left[\hat{g}_k(x_k) + \nabla \hat{g}_k(x_k) n_k \right]^-, \\ & l \leq x_k + n_k + s \leq u, \\ & \|n_k + s\| \leq \Delta_k. \end{aligned}$$

Interpolation-based models

Interpolation-based quadratic models

COBYQA builds quadratic models of f and g by interpolation.

Derivative-free symmetric Broyden update (Powell 2004)

The kth model $\hat{f}_k$ of f solves

$$\begin{aligned} \min_{Q \in \mathcal{Q}_n} \quad & \left\| \nabla^2 \hat{f}_{k-1} - \nabla^2 Q \right\|_{\mathsf{F}} \\ \text{s.t.} \quad & Q(y) = f(y), \ y \in \mathcal{Y}_k, \end{aligned}$$

for some interpolation set $\mathcal{Y}_k \subseteq \mathbb{R}^n$ (similar for $\hat{g}_k$).

- We recycle $\mathcal{Y}_{k+1} = (\mathcal{Y}_k \cup \{x_k + s_k\}) \setminus \{\bar{y}\}$ for some bad point $\bar{y} \in \mathcal{Y}_k$.
- To compute $\hat{f}_k$, we only need to solve a linear system.

Some alternatives: Conn, Scheinberg, and Toint (1997a,b, 1998), Wild (2008), Custódio, Rocha, and Vicente (2010), Bandeira, Scheinberg, and Vicente (2012), Zhang (2014), and Xie and Yuan (2023).

Many difficulties arise

A lot of questions must be addressed

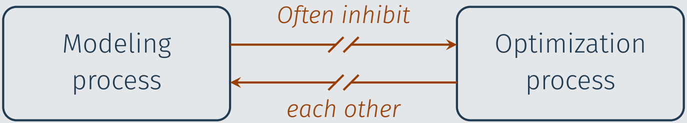
- How to calculate the steps n_k and t_k numerically?
  COBYQA adapts the truncated conjugate gradient method.
- What is the approximate Lagrange multiplier λ_k?
  We choose the least-squares Lagrange multiplier.
- Which merit function should we use?
  COBYQA uses the ℓ_2-merit function.
- How to update the penalty parameter?
  The update incorporates
    - a theoretical value for the exactness of the merit function, and
    - a strategy used by Powell in COBYLA.

These questions (and many more) are addressed in Ragonneau (2022).

A crucial difficulty in the implementation

- What if the interpolation set $\mathcal{Y}_k$ is almost nonpoised?
  A well-known approach: a geometry-improving mechanism.[2]

This is a central difficulty in the implementation of DFO methods



- The iterates $\{x^k\}$ likely lie on a particular path.
- The modeling process does not ponder the optimization problem.

[2] See Conn, Scheinberg, and Vicente (2008a,b) and Fasano, Morales, and Nocedal (2009).

Management of the trust-region radius

We maintain Δ_k and a lower bound $\delta_k \le \Delta_k$

- The lower bound δ_k is never increased.
- We update Δ_k in the usual way, but we always have $\Delta_k \ge \delta_k$.
- This strategy is adapted from Powell (2006, 2009) and LINCOA.

The value of δ_k is an indicator of the current resolution.

- Without $\Delta_k \ge \delta_k$, the value of Δ_k may become too small.
- It prevents the interpolation points from concentrating too much.
- The value of δ_k is only decreased when necessary.
- Hence, stopping when $\delta_k \le \delta_{\text{end}}$ is reasonable ($\delta_{\text{end}} > 0$).

Implementation and experiments

The Python implementation of COBYQA

From Powell (2006)

> "The development of NEWUOA has taken nearly three years. The work was very frustrating [...]"

The development of COBYQA was not easier.

We implemented COBYQA in Python and made it publicly available.

www.cobyqa.com

```
> pip install cobyqa
```

Comparing COBYQA with existing DFO solvers

- We assess the quality of points based on the merit function

$$\varphi(x) = \begin{cases} f(x) & \text{if } v_\infty(x) \le 10^{-10}, \\ \infty & \text{if } v_\infty(x) \ge 10^{-5}, \\ f(x) + 10^5 v_\infty(x) & \text{otherwise,} \end{cases}$$

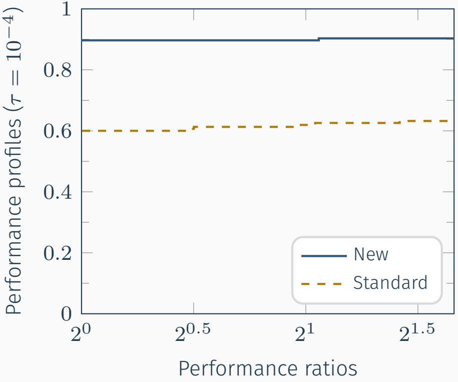
  where v_∞ denotes the ℓ_∞-constraint violation.
- The problems are from the CUTEst set.
- The problems are of dimension at most 50 (this is not small).
- Problems with unrelaxable bounds replace f with

$$\tilde{f}(x) = \begin{cases} f(x) & \text{if } l \le x \le u, \\ \infty & \text{otherwise.} \end{cases}$$

Performance of the new Byrd-Omojokun approach

We compare the new and the standard Byrd-Omojokun approaches

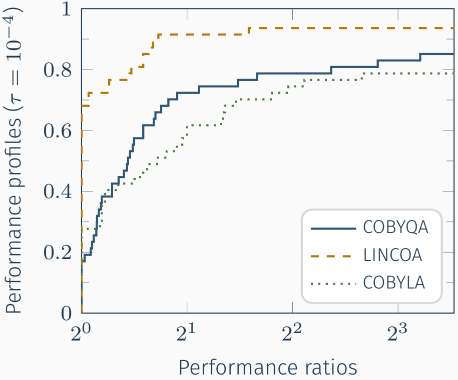
- on linearly and nonlinearly constrained problems,
- in the implementation of COBYQA.



Performance on linearly constrained problems

We compare COBYQA, LINCOA, and COBYLA

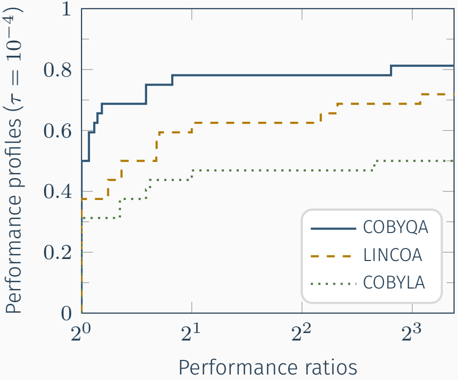
- on linearly constrained problems,



Performance on linearly constrained problems

We compare COBYQA, LINCOA, and COBYLA

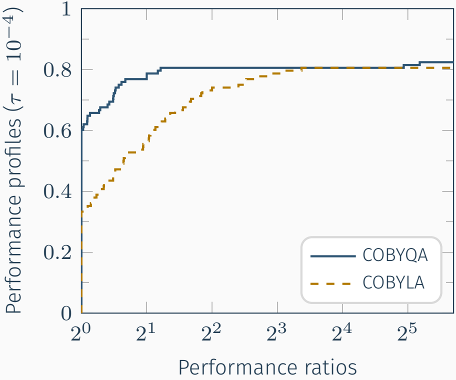
- on linearly constrained problems,
- with unrelaxable bounds.



Performance on nonlinearly constrained problems

We compare COBYQA and COBYLA

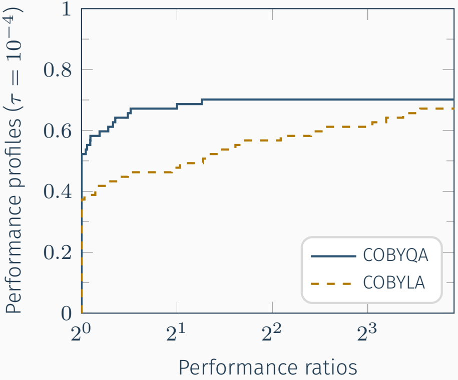
- on nonlinearly constrained problems,



Performance on nonlinearly constrained problems

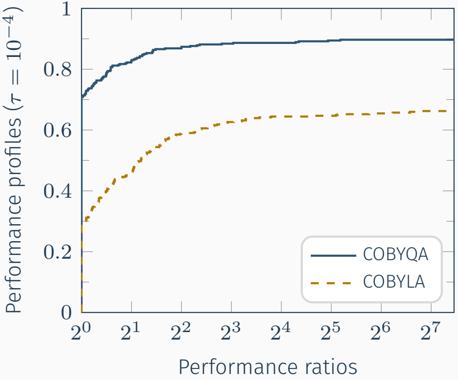
We compare COBYQA and COBYLA

- on nonlinearly constrained problems,
- with unrelaxable bounds.



Comparison with COBYLA

We compare COBYQA and COBYLA on all 388 problems.



Conclusion

Conclusion

- COBYQA already received positive feedback from practitioners.
- It will soon be included in
  - PDFO as a successor for COBYLA, and
  - GEMSEO, an industrial software package for MDO.
- We will soon investigate the convergence properties of COBYQA.

For more information, visit:

COBYQA

My website

My thesis

References i

- Bandeira, A. S., Scheinberg, K., and Vicente, L. N. (2012). **"Computation of sparse low degree interpolating polynomials and their application to derivative-free optimization."** *Math. Program.* 134.1, pp. 223–257.
- Conn, A. R., Gould, N. I. M., and Toint, Ph. L. (2000). ***Trust-Region Methods.*** MOS-SIAM Series on Optimization. Philadelphia, PA, USA: SIAM.
- Conn, A. R., Scheinberg, K., and Toint, Ph. L. (1997a). **"On the convergence of derivative-free methods for unconstrained optimization."** In: *Approximation Theory and Optimization: Tributes to M. J. D. Powell.* Ed. by M. D. Buhmann and A. Iserles. Cambridge, UK: Cambridge University Press, pp. 83–108.
- — (1997b). **"Recent progress in unconstrained nonlinear optimization without derivatives."** *Math. Program.* 79.1–3, pp. 397–414.

- Conn, A. R., Scheinberg, K., and Toint, Ph. L. (1998). **"A derivative free optimization algorithm in practice."** In: *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. St. Louis, MO, USA: AIAA, pp. 129–139.
- Conn, A. R., Scheinberg, K., and Vicente, L. N. (2008a). **"Geometry of interpolation sets in derivative free optimization."** *Math. Program.* 111.1–2, pp. 141–172.
- — (2008b). **"Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation."** *IMA J. Numer. Anal.* 28.4, pp. 721–748.
- Custódio, A. L., Rocha, H., and Vicente, L. N. (2010). **"Incorporating minimum frobenius norm models in direct search."** *Comput. Optim. Appl.* 46.2, pp. 265–278.

References iii

- Fasano, G., Morales, J. L., and Nocedal, J. (2009). **"On the geometry phase in model-based algorithms for derivative-free optimization."** *Optim. Methods Softw.* 24.1, pp. 145–154.
- Powell, M. J. D. (1994). **"A direct search optimization method that models the objective and constraint functions by linear interpolation."** In: *Advances in Optimization and Numerical Analysis*. Ed. by S. Gomez and J. P. Hennart. Vol. 275. Mathematics and Its Applications. Dordrecht, The Netherlands: Springer, pp. 51–67.
- — (2004). **"Least Frobenius norm updating of quadratic models that satisfy interpolation conditions."** *Math. Program.* 100.1, pp. 183–215.
- — (2006). **"The NEWUOA software for unconstrained optimization without derivatives."** In: *Large-Scale Nonlinear Optimization*. Ed. by G. Di Pillo and M. Roma. Vol. 83. Nonconvex Optimization and Its Applications. Boston, MA, USA: Springer, pp. 255–297.

References iv

- Powell, M. J. D. (2009). ***The BOBYQA algorithm for bound constrained optimization without derivatives.*** Tech. rep. DAMTP 2009/NA06. Cambridge, UK: Department of Applied Mathematics and Theoretical Physics, University of Cambridge.
- Ragonneau, T. M. (2022). **"Model-Based Derivative-Free Optimization Methods and Software."** PhD thesis. Hong Kong, China: The Hong Kong Polytechnic University.
- Wild, S. M. (2008). **"MNH: a derivative-free optimization algorithm using minimal norm Hessians."** In: *Tenth Copper Mountain Conference on Iterative Methods.*
- Xie, P. and Yuan, Y. (2023). ***Least H^2 norm updating quadratic interpolation model function for derivative-free trust-region algorithms.*** arXiv:2302.12017.

References v

- Zhang, Z. (2014). "Sobolev seminorm of quadratic functions with applications to derivative-free optimization." *Math. Program.* 146.1–2, pp. 77–96.