Model-Based Derivative-Free Optimization with Unrelaxable Constraints

Trever Hallock

September 13, 2021

Introduction

- We develop an algorithm to find local optima of constrained problems
- Derivatives not available, only function values
- This algorithm is designed for problems with unrelaxable constraints

Unrelaxable Constraints

Function values are unavailable for infeasible points

Derivative Free Problem Formulation

$$\min_{x} \quad f(x)$$
$$c_i(x) \leq 0 \quad \forall\, 1 \leq i \leq m$$

- All functions are black-box functions, meaning that we have no information about their derivatives
- For example, optimization problems where the objective or some of the constraints depend on an expensive simulation
- Function values may not be available outside of the feasible region, we call these unrelaxable constraints

Applications

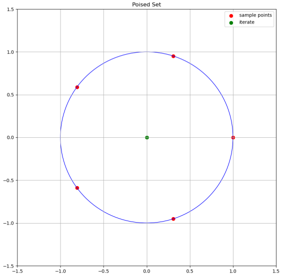
Problems can arise several different ways [Digabel and Wild, 2015]:

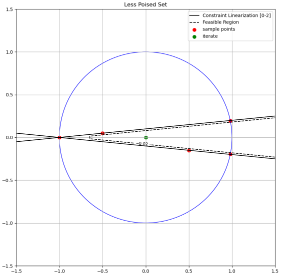
- Physical models are not meaningful for certain inputs
  - A simulation quantity represents a concentration level
  - Well rates in groundwater problems
  - When decision variables must be ordered
- Simulations may not converge
  - Inverse Transport Problem [Armstrong and Favorite, 2016]

Why are Unrelaxable Constraints Hard?

Building accurate models is harder

- Models require feasible sample points:





Why are Unrelaxable Constraints Hard?

Constraint boundaries are uncertain

- Infeasible function calls are wasteful... How do we avoid them?

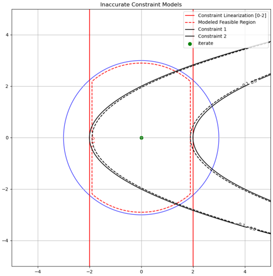


Table of Contents

1 Introduction

2 Derivative Free Background

3 Always Feasible Algorithm

4 Conclusion

Model-Based Trust Region Framework

Step 1 **Construct a model for the current point**
Choose sample points to construct a model

Step 2 **Check optimality.**
Compute $\chi\left(x^{(k)}\right)$ and compare to a threshold

Step 3 **Compute the trial step, and evaluate**
Solve the trust region subproblem using the model functions

Step 4 **Check reduction and update radius**
If little progress was made, decrease the trust region radius.
Otherwise, set the next iterate to the trial point

Model-Based Trust Region Methods

- Approximate f and c using model functions interpolated over a sample set
- We approximate f using a second order model:
  $m_f(x) =$
  $f_k + \left(x - x^{(k)}\right)^T g_f^{(k)} + \frac{1}{2}\left(x - x^{(k)}\right)^T H_f^{(k)}\left(x - x^{(k)}\right) \approx f(x)$
  - $g_f^{(k)} \approx \nabla f(x^{(k)})$
  - $H_f^{(k)} \approx \nabla^2 f(x^{(k)})$

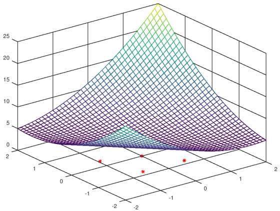
  $m_{c_i}(x) = c_i^{(k)}(x^{(k)}) + \left(x - x^{(k)}\right)^T g_{c_i}^{(k)} \approx c_i(x)$
  - $g_{c_i}^{(k)} \approx \nabla c_i(x^{(k)})$
- The model's accuracy depends on the sample point choices

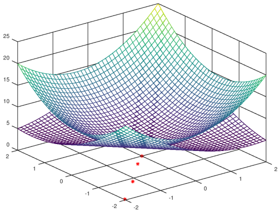
Geometry of the Sample Set

- Geometry refers to the relative positions of the sample set of sample points
- When the points are not “well poised”, the constructed model can be inaccurate

Geometry of the Sample Set

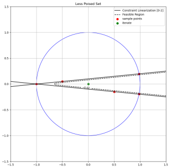
- Geometry refers to the relative positions of the sample set of sample points
- When the points are not "well poised", the constructed model can be inaccurate





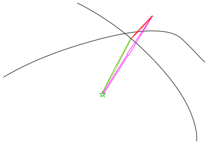
Geometry of the Sample Set

- Geometry refers to the relative positions of the sample set of sample points
- When the points are not "well poised", the constructed model can be inaccurate
- Constructing poised sets over ellipses is well known [Conn et al.,]
- Constraints limit what points are available for the sample set
- With narrow constraints, well poised sets may not exist



Stopping Conditions

- The criticality measure $\chi(x)$ measures how close to optimality a point x may be
- For example, with convex constraints, we can use
  $\chi(x) = \|P_F(x - \nabla f(x)) - x\|$,
  $\chi_m^{(k)}(x) = \left\|P_{F_m^{(k)}}\left(x - \nabla m_f^{(k)}(x)\right) - x\right\|$
- If x satisfies the first order necessary conditions for optimality, then $\chi(x) = 0$
- To show a DFO algorithm converges, we show $\chi\left(x^{(k)}\right) \to 0$. This can done by showing $\Delta_k \to 0$ and $\chi_m^{(k)}\left(x^{(k)}\right) \to 0$



Trust Region Subproblem

- In each iteration, we attempt to solve the trust region subproblem to compute a step direction s

$$\begin{array}{llcl} min_s & m_f(x^{(k)}+s) & & \\ s.t. & m_{c_i}(x^{(k)}+s) & \leq & 0 \quad \forall\, 1 \leq i \leq m \\ & s & \in & B_\infty(0,\Delta_k). \end{array}$$

- Replaced true functions with model functions
- Added trust region constraint
- The solution is then used as a trial point for the next iterate

Evaluating the Trial Point

- After evaluating, we check if the trial point produced enough reduction
- One quantity to measure the model's accuracy and the trial points reduction is

$$\rho_k = \frac{f\left(x^{(k)}\right) - f\left(x^{(k)} + s^{(k)}\right)}{m_k\left(x^{(k)}\right) - m_k\left(x^{(k)} + s^{(k)}\right)}$$

  - If $\rho_k < \gamma_{\mathrm{min}}$, $x^{(k+1)} = x^{(k)}$ (reject) and decrease radius
  - If $\gamma_{\mathrm{min}} \le \rho_k < \gamma_{\mathrm{suff}}$, $x^{(k+1)} = x^{(k)} + s^{(k)}$ (accept) and decrease radius
  - If $\gamma_{\mathrm{suff}} \le \rho_k$, $x^{(k+1)} = x^{(k)} + s^{(k)}$ (accept) and increase radius

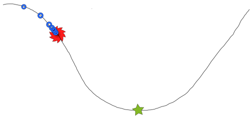
Feasible Derivative Free Algorithm

- Our algorithm is based on a general algorithmic framework proposed by [Conejo et al., 2013]
- This paper provides convergence analysis, without depending on implementation details
- This framework assumes the following properties:
  - Quadratic or linear model functions
  - Ability to satisfy an efficiency condition
  - Ability to satisfy an accuracy condition
  - A method for projecting points onto the feasible set

Algorithm Assumptions

- The algorithm can satisfy the efficiency condition if it produces trial points that reduce the objective's model:

$$m_f^{(k)}\left(x^{(k)}\right) - m_f^{(k)}\left(x^{(k+1)}\right)$$
$$\geq c_1 \chi^{(k)} \min\left\{\frac{\chi^{(k)}}{1 + \left\|\nabla^2 m_f^{(k)}\left(x^{(k)}\right)\right\|}, \Delta_k, 1\right\}$$



Algorithm Assumptions

- The algorithm can satisfy the efficiency condition if it produces trial points that reduce the objective's model:

$$m_f^{(k)}\left(x^{(k)}\right) - m_f^{(k)}\left(x^{(k+1)}\right)$$
$$\geq c_1 \chi_m^{(k)} \min\left\{\frac{\chi_m^{(k)}}{1+\left\|\nabla^2 m_f^{(k)}\left(x^{(k)}\right)\right\|}, \Delta_k, 1\right\}$$

- The accuracy condition ensures:

$$\left\|\nabla m_f^{(k)}\left(x^{(k)}\right) - \nabla f\left(x^{(k)}\right)\right\| \leq c_2 \Delta_k$$

The Algorithm for Linear Constraints

- The linear algorithm is an implementation of the algorithm within [Conejo et al., 2013]
- We satisfy the hypothesis presented in this article
- We need sample points to ensure the model satisfies the accuracy condition

$$\left\| \nabla m_f^{(k)} \left(x^{(k)} \right) - \nabla f \left(x^{(k)} \right) \right\| \leq c_2 \Delta_k$$

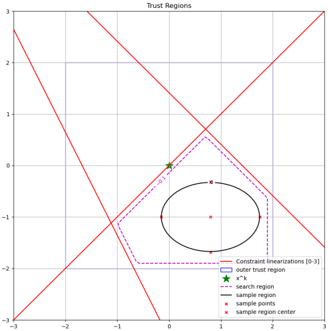
- We were able to rely on classic model improving algorithms by choosing an ellipsoidal sample region

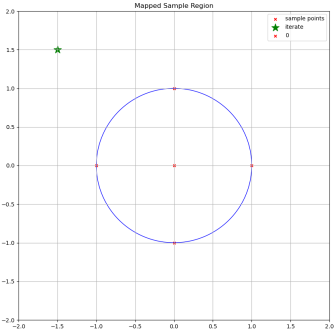
Feasible Derivative Free Trust Regions

- The outer trust region:
  - Is an L_∞ ball
  - Can contain infeasible points
  - Contains both the other trust regions.
- The sample region:
  - Used to construct sample points
  - Must become feasible
  - Must be constructed from the previous models
- The search region:
  - Used to construct trial points
  - Must become feasible
  - Must allow for sufficient reduction

Feasible Derivative Free Trust Regions

We map the sample region to a sphere while constructing the sample points





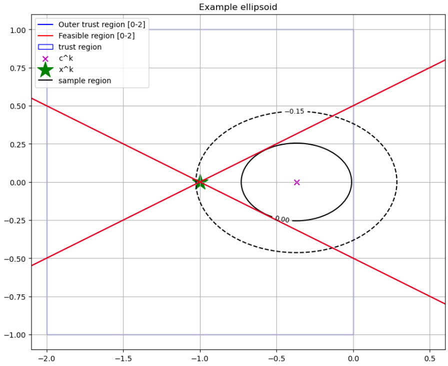
Sample Region Requirements

- The sample region is given by the ellipsoid

$$\left\{ x \in \mathbb{R}^n \middle| \left(x - c^{(k)}\right)^T Q^{(k)} \left(x - c^{(k)}\right) \leq \frac{1}{2}\delta_k^2 \right\}$$

- The condition number of $Q^{(k)}$ must be bounded
- The ellipsoid must be near the current iterate
- The ellipsoid must be feasible

Example Trust Region

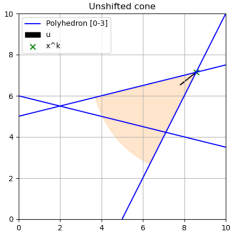


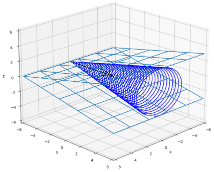
Ellipsoid Construction

- Construct direction feasible with respect to the active constraints

$$\hat{u}^{(k)} = \arg\max_{\|u\|=1} \min_{i \in \mathcal{A}_k} u^T \frac{-\nabla m_{c_i}\left(x^{(k)}\right)}{\left\|\nabla m_{c_i}\left(x^{(k)}\right)\right\|}$$

- Build an ellipsoid within the widest second order cone



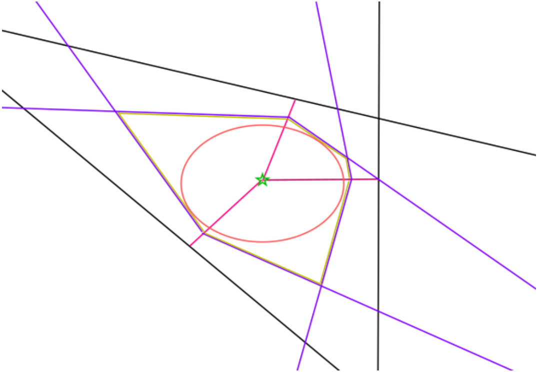


Linear Algorithm Summary

- We implemented an instance of Conejo et al's algorithm using only feasible sample point
- We provided a set of criteria for sample regions that provide accurate model functions
- We provided one construction satisfying this criteria
- This implementation satisfies the assumptions used in Conejo's proof
- We also explored other implementations, such as a polyhedral sample region

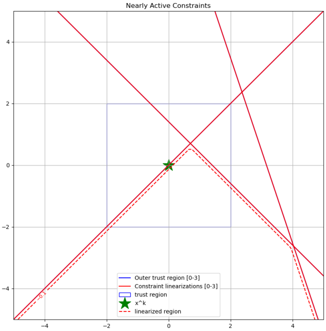
Nonlinear Algorithm

To handle inaccurate constraint models, we chose to buffer the sample and search regions from the constraint boundaries



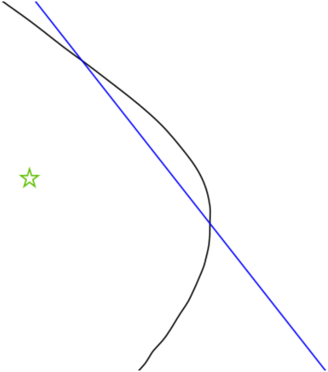
Nearly-Active Constraints

- To construct the buffered region, we first identify nearly active constraints



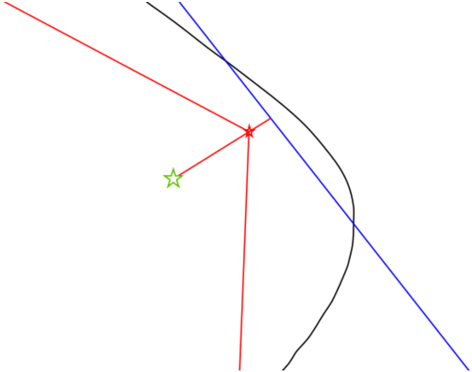
Buffering Cones

We construct the buffering cones as follows:



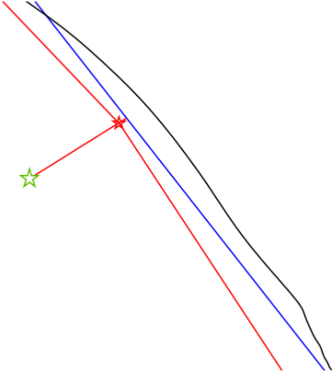
Buffering Cones

The cone's vertex is the linearization's zero, scaled towards the current iterate



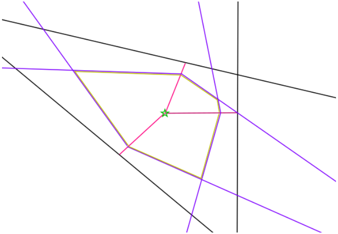
Buffering Cones

As the $\Delta_k \to 0$, the buffered region approaches the linearization.



Buffering Cones

We show the buffered region–the intersection of these cones–is feasible for small Δ_k



Buffering Cones

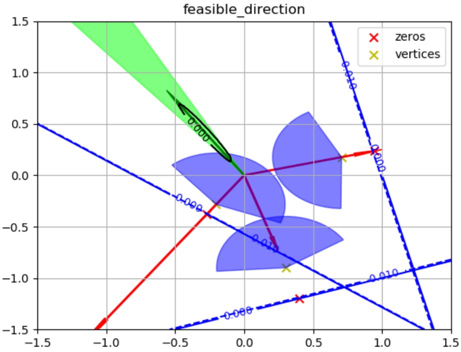
Theorem 4.24

Suppose that assumptions 4.4-4.9 hold. Suppose that $C^{(i,k)}$ is the buffering cone for the i-th constraint during iteration k, that F is the true feasible region, and that $\mathcal{A}^{(k)}$ is the set of active constraints at iteration k.

There exists a $\Delta_{\text{feasible}} > 0$ such that if $\Delta_k \leq \Delta_{\text{feasible}}$, then $\left[\bigcap_{i\in\mathcal{A}^{(k)}} C^{(i,k)}\right] \cap \left[B_\infty\left(x^{(k)}, \Delta_k\right) \cup B_\infty\left(x^{(k+1)}, \Delta_{k+1}\right)\right] \subseteq F$.

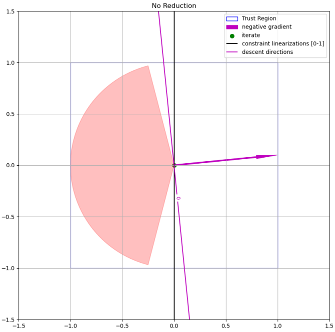
Conservative Construction

We construct an ellipsoid within the recession cone of the buffered region



No Buffered Reduction Possible

While Δ_k is large, the buffered region may not provide reduction



Sufficient Reduction

- We use the the buffered region as the search region, which limits trial points
- We can no longer use well-known algorithms for computing an efficient trial point

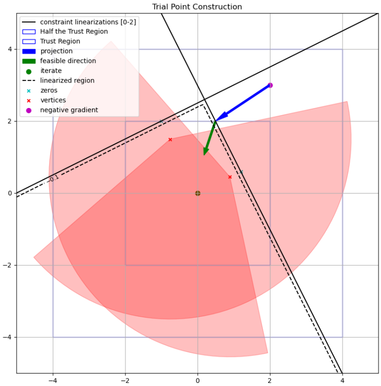
Theorem 4.27

Suppose that assumptions 4.4-4.7 and 4.9 hold. If $\chi^{(k)} \geq \kappa_\chi \Delta_k^{p_\Delta}$ and $\Delta_k \leq \Delta_{\mathrm{sf}}$, then there is a v in the buffered region that satisfies the efficiency condition.

- Requires small Δ_k, so we must explicitly check for reduction.

Feasible Trial Point

Moving a solution to the buffered region



Criticality Measure

- The classic criticality for convex constraints is

$$\chi(x) = \left\| x^{(k)} - \mathbf{P}_{\mathcal{F}}\left(x - \nabla f(x)\right) \right\|$$

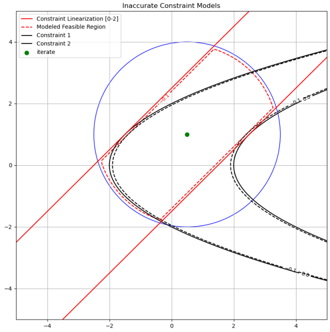
- We only have access to models:

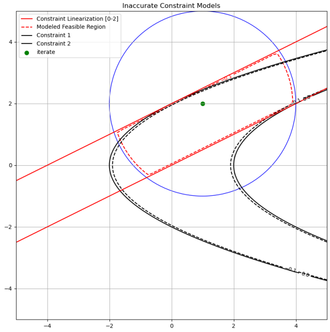
$$\chi_m^{(k)}(x) = \left\| x^{(k)} - \mathbf{P}_{\mathcal{F}^{(k)}}\left(x - \nabla m_f^{(k)}(x)\right) \right\|$$

- We showed that $\left| \chi_m^{(k)}\left(x^{(k)}\right) - \chi\left(x^{(k)}\right) \right| \to 0$

Convergence of Criticality Measure

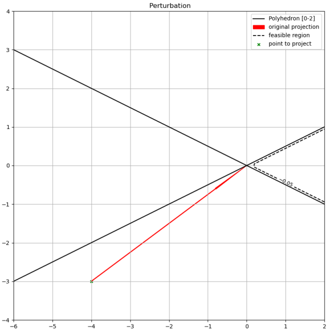
The criticality measure changes with constraint model changes

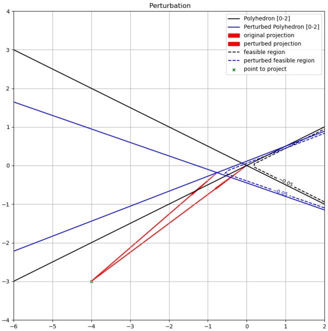




Bounded Projection

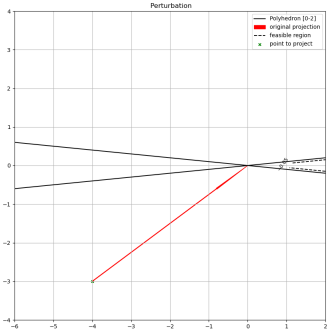
Projections don't move far when some constraints are perturbed...

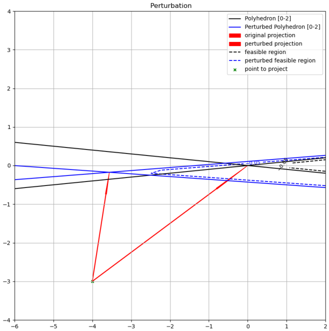




Bounded Projection

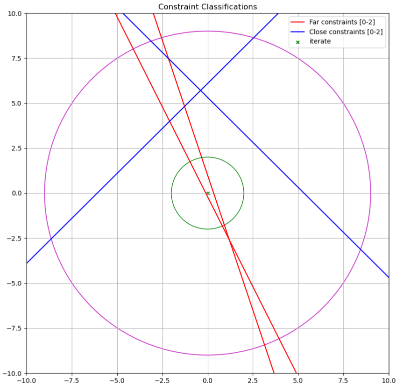
...but projections can for narrow constraints





Bounded Projection

Only constraints near the current iterate can make a small angle



Bounded Projection

- How far a projection onto the linearized feasible region moves depends on $\min_{\|u\|=1, u \geq 0} \left\| \sum_i u_i \nabla \hat{c}_i \left(x^{(k)} \right) \right\|$
- This quantity is bounded by a regularity assumption

Theorem 4.41

Suppose that assumptions 4.3-4.9 hold. Let $F_m^{(k)}$ and $F_c^{(k)}$ be the model's and constraints linearized feasible region for iteration k respectively, and $d^{(k)} = x^{(k)} - \nabla m_f \left(x^{(k)} \right)$. Then,

$$\begin{aligned} \left\| P_{F_m^{(k)}} \left(d^{(k)} \right) - P_{F_c^{(k)}} \left(d^{(k)} \right) \right\| &\rightarrow 0 \qquad \text{and} \\ \left\| P_{F_m^{(k)}} \left(d^{(k)} \right) - P_{F_m^{(l)}} \left(d^{(k)} \right) \right\| &\rightarrow 0. \end{aligned}$$

Regularity Assumption

- The Mangasarian-Fromovitz constraint qualification at a critical point $x^\star$ requires

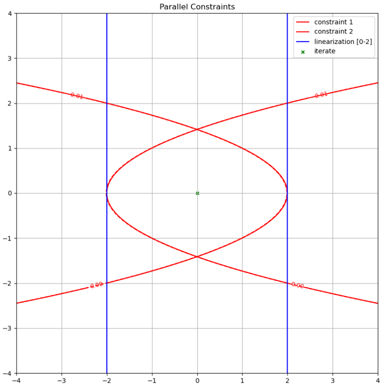
$$\exists d \in \mathbb{R}^n, \forall i, c_i\left(x^\star\right) = 0 \Longrightarrow \nabla c_i\left(x^\star\right)^T d < 0$$

- We strengthened this qualification in two ways:

$$\forall x \exists d \in \mathbb{R}^n \; \forall i, \nabla c_i\left(x\right)^T d < 0$$

Regularity Assumption

Only assume regularity for nearly active constraints



Regularity Assumption

- The Mangasarian-Fromovitz constraint qualification at a critical point $x^\star$ requires

$$\exists d \in \mathbb{R}^n, \forall i, c_i\left(x^\star\right) = 0 \Longrightarrow \nabla c_i\left(x^\star\right)^T d < 0$$

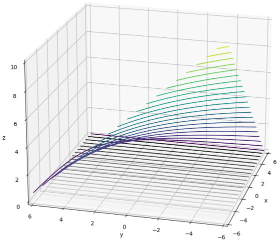
- We strengthened this qualification in two ways:

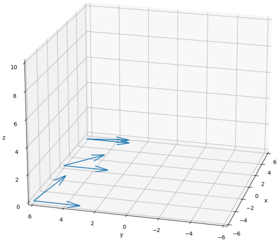
$$\cancel{\forall x \exists d \in \mathbb{R}^n \, \forall i, \nabla c_i\left(x\right)^T d < 0}$$

$$\forall x \exists d \in \mathbb{R}^n \, \forall i, \nabla c_i\left(x\right) \approx 0 \Longrightarrow \nabla c_i\left(x\right)^T d < 0$$

Regularity Assumption

We ensure a uniform bound on the “width” of the feasible set’s tangent cone





Regularity Assumption

- The Mangasarian-Fromovitz constraint qualification at a critical point $x^\star$ requires

$$\exists d \in \mathbb{R}^n, \forall i, c_i\left(x^\star\right) = 0 \Longrightarrow \nabla c_i\left(x^\star\right)^T d < 0$$

- We strengthened this qualification in two ways:

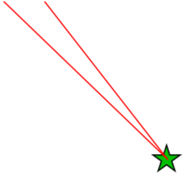
$$\cancel{\forall x \exists d \in \mathbb{R}^n \forall i, \nabla c_i\left(x\right)^T d < 0}$$

$$\cancel{\forall x \exists d \in \mathbb{R}^n \forall i, \nabla c_i\left(x\right) \approx 0 \Longrightarrow \nabla c_i\left(x\right)^T d < 0}$$

$$\exists \epsilon > 0 \forall x \exists d \in \mathbb{R}^n \ \forall i \ c_i\left(x\right) \approx 0 \Longrightarrow \nabla c_i\left(x\right)^T d < -\epsilon$$

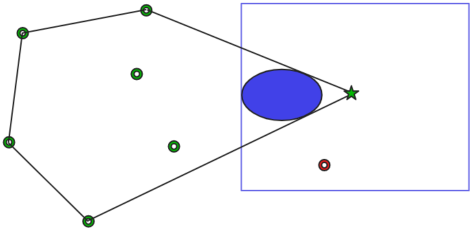
Ellipsoid Recovery

- Given a single feasible point, in general, it can be difficult to find even a second feasible point
- This motivates a feasible starting set
- For general constraints, we assume a recovery subroutine
- For convex constraints, we provide such an algorithm



Ellipsoid Recovery for Convex Constraints

We can construct a sample region within the convex hull of previously evaluated feasible points



Convergence Results: The General Case

Theorem 4.43

Suppose that assumptions 4.3-4.9 hold. Suppose that $F_c^{(k)}$ is the true feasible region's linearization at $x^{(k)}$. If $\gamma_{\min} = 0$,

$$\liminf_{k\to\infty} \chi\left(x^{(k)}\right) = \liminf_{k\to\infty} \left\| P_{F_c^{(k)}}\left(x^{(k)} - \nabla f\left(x^{(k)}\right)\right) - x^{(k)} \right\| = 0.$$

If $\gamma_{\min} > 0$,

$$\lim_{k\to\infty} \chi\left(x^{(k)}\right) = \lim_{k\to\infty} \left\| P_{F_c^{(k)}}\left(x^{(k)} - \nabla f\left(x^{(k)}\right)\right) - x^{(k)} \right\| = 0.$$

Convergence Results: The Convex Case

Corollary 4.44

Suppose that assumptions 4.3-4.8 hold, and that the feasible region F is convex.
If $\gamma_{\min} = 0$,

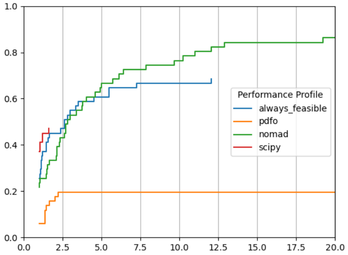
$$\liminf_{k\to\infty} \chi\left(x^{(k)}\right) = \liminf_{k\to\infty} \left\| P_F\left(x^{(k)} - \nabla f\left(x^{(k)}\right)\right) - x^{(k)} \right\| = 0.$$

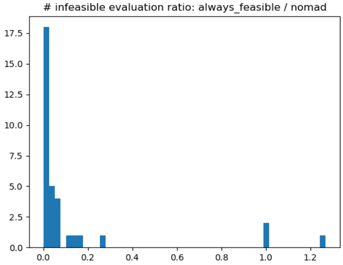
If $\gamma_{\min} > 0$,

$$\lim_{k\to\infty} \chi\left(x^{(k)}\right) = \lim_{k\to\infty} \left\| P_F\left(x^{(k)} - \nabla f\left(x^{(k)}\right)\right) - x^{(k)} \right\| = 0.$$

Numerical Results

We compared our algorithm to PDFO, NOMAD, and SCIPY.optimize





Contributions

- First model-based DFO algorithm for unrelaxable constraints
- Constructed feasible ellipsoids
- Showed sufficient reduction within a buffered region
- Convergence in criticality measure
- A modified regularity assumption
- Numerical results with few infeasible evaluation attempts

Future Work

- Show error bounds for polyhedral trust regions
- Use fewer sample points on narrow constraints
- Make assumptions only reference the true constraints

Questions?

References I

Armstrong, J. C. and Favorite, J. A. (2016).
Using a derivative-free optimization method for multiple solutions of inverse transport problems.
Optimization and Engineering, 17(1).

Conejo, P. D., Karas, E. W., Pedroso, L. G., Ribeiro, A. A., and Sachine, M. (2013).
Global convergence of trust-region algorithms for convex constrained minimization without derivatives.
220:324–330.

Conn, A. R., Scheinberg, K., and Vicente, L. N.
Introduction to Derivative-Free Optimization.
Society for Industrial and Applied Mathematics.

References II

Digabel, S. L. and Wild, S. M. (2015).
A taxonomy of constraints in simulation-based optimization.