A new derivative-free interior point method for constrained black-box optimization

A. Brilli, G. Liuzzi, **S. Lucidi**

17-22 July 2022

Derivative-Free Optimization: Linking Algorithms and Applications
University of British Columbia Okanagan (UBCO), Canada

Outline of the talk

Derivative-free & black-box optimization

We consider

$$\min_{x\in\mathbb{R}^n} \; f(x)$$
$$s.t. \; g_i(x) \geq 0, \quad i \in \mathcal{I} = \{1,\ldots,m\}$$

where evaluations of f and g_i are the results of (possibly) complex computer simulations

$x \in \mathbb{R}^n \longrightarrow$  $\longrightarrow f(x), g(x)$

- calls to the simulator are expensive
- f is not defined when $g_i(x) < 0$
- derivatives are not available (or untrustworthy, or difficult to obtain)

Motivations

1. In many real-world problems objective and constraint function values are the **result of complex computer simulation codes**
2. **First** (and higher order) **derivatives are unavailable** or impractical to obtain or untrustworthy
3. **Objective function** could be **undefined outside the feasible region**

Other approaches

- [Alarie,Audet,Jacquot,Le Digabel, ORL 2022] Hierarchically constrained problems
- [Le Digabel, ACM 2011], [Audet et al., arXiv 2021] NOMAD
  Extreme and progressive barrier, smooth and nonsmooth, hidden constraints
- [Cristofari,Rinaldi, SIOPT 2021] ORD Structured constrained problems, smooth
- [Audet,Tribes, COAP 2018] Mesh-based Nelder-Meade
- [Audet,Conn,Le Digabel,Peyrega, COAP 2018] Progressive barrier trust-region
- [Audet,Hare, 2017] Derivative-Free and Blackbox Optimization
- [Reggis,Wild, OMS 2017] use of RBF in trust-region for constrained probs
- [Diouane,Gratton,Vicente, COAP 2015] Use of barrier functions within evolution strategies
- [Fasano,L.,Lucidi,Rinaldi, SIOPT 2014] DFN Exterior exact penalty, nonsmooth
- [L.,Lucidi,Sciandrone, SIOPT 2010] PENSEQ Exterior sequential penalty, smooth
- [Conn,Scheinberg,Vicente 2009] Introduction to Derivative-Free Optimization
- [Audet,Dennis Jr, SIOPT 2009] Progressive barrier for constrained DF
- [L.,Lucidi, SIOPT 2009] Exterior exact ℓ_∞ penalty, smooth

Introduction

Given problem

$$\begin{aligned} &\min\ f(x) \\ &s.t.\ x \in \mathcal{S} = \{x \in \mathbb{R}^n :\ g_i(x) \geq 0, i \in \mathcal{I}\} \\ &\qquad \mathcal{I} = \{1, 2, \ldots, m\} \end{aligned}$$

assume

- f and $g_i, i \in \mathcal{I}$ black-box type functions
- f and $g_i, i \in \mathcal{I}$ continuously differentiable
- $g_i \geq 0, i \in \mathcal{I}$ not relaxable constraints
- $\mathcal{S}$ is compact
- $\mathring{\mathcal{S}} \neq \emptyset$ and a strictly feasible $x_0 \in \mathring{\mathcal{S}}$ is known

The Lagrangian function and its gradient (w.r.t. x) are

$$L(x, \lambda) = f(x) - \lambda^\top g(x)$$
$$\nabla_x L(x, \lambda) = \nabla f(x) - \nabla g(x)\lambda$$

Definition [KKT point]

$\bar{x} \in \mathcal{S}$ is a KKT point if $\bar{\lambda}_i$, $i \in \mathcal{I}$, exist such that

$$\nabla_x L(\bar{x}, \bar{\lambda}) = 0,$$
$$\bar{\lambda} \geq 0,\ \bar{\lambda} \perp g(\bar{x})$$

Log-barrier penalty function

We introduce

$$P(x;\mu) = f(x) - \mu \sum_{i=\mathcal{I}} \log(g_i(x))$$

$$\nabla P(x;\mu) = \nabla f(x) - \sum_{i\in\mathcal{I}} \frac{\mu}{g_i(x)} \nabla g_i(x)$$

and consider, for $\mu > 0$, the penalized problem

$$\begin{aligned} &\min \;\; P(x;\mu) \\ &s.t. \; x \in \mathring{\mathcal{S}} \end{aligned}$$

- When μ is fixed:
  - problem is "essentially" unconstrained
  - it can be solved by easily adapting a LS derivative-free method
- μ must be driven to zero to solve the original problem

The algorithm when μ is fixed

Algorithm 1: DF Linesearch `DFL`

Data: $x_0 \in \mathring{\mathcal{S}}$, $\mu > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \ldots, n$

for $k = 0, 1, 2, \ldots$ **do**

  Set $y_k^1 = x_k$

  for $i = 1, 2, \ldots, n$ **do**

    if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu)$ *can be suff. reduced along* d_k^i **then**

      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*

The algorithm when μ is fixed

Algorithm 2: DF Linesearch DFL

Data: $x_0 \in \overset{\circ}{\mathcal{S}}$, $\mu > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \dots, n$

for $k = 0, 1, 2, \dots$ **do**

Set $y_k^1 = x_k$

for $i = 1, 2, \dots, n$ **do**

if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and* $P(y_k^i;\mu)$ *can be suff. reduced along* d_k^i **then**

compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*

else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and* $P(y_k^i;\mu)$ *can be suff. reduced along* $-d_k^i$ **then**

compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*

The algorithm when μ is fixed

Algorithm 3: DF Linesearch `DFL`

Data: $x_0 \in \mathring{\mathcal{S}}$, $\mu > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \dots, n$

for $k = 0, 1, 2, \dots$ **do**

  Set $y_k^1 = x_k$

  for $i = 1, 2, \dots, n$ **do**

    if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu)$ *can be suff. reduced along* d_k^i **then**

      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*

    else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu)$ *can be suff. reduced along* $-d_k^i$ **then**

      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*

    else

      Set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$ *(failure step)*

    end

The algorithm when μ is fixed

Algorithm 4: DF Linesearch `DFL`

Data: $x_0 \in \overset{\circ}{\mathcal{S}}$, $\mu > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \dots, n$
for $k = 0, 1, 2, \dots$ **do**
    Set $y_k^1 = x_k$
    for $i = 1, 2, \dots, n$ **do**
        if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and $P(y_k^i; \mu)$ can be suff. reduced along d_k^i* **then**
            compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*
        else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and $P(y_k^i; \mu)$ can be suff. reduced along $-d_k^i$* **then**
            compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*
        else
            Set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$ *(failure step)*
        end
        Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$
    end

The algorithm when μ is fixed

Algorithm 5: DF Linesearch `DFL`

Data: $x_0 \in \mathring{\mathcal{S}}$, $\mu > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \dots, n$
for $k = 0, 1, 2, \dots$ **do**
  Set $y_k^1 = x_k$
  for $i = 1, 2, \dots, n$ **do**
    if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu)$ *can be suff. reduced along* d_k^i **then**
      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*
    else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu)$ *can be suff. reduced along* $-d_k^i$ **then**
      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*
    else
      Set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$ *(failure step)*
    end
    Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$
  end
  Find $x_{k+1} \in \mathring{\mathcal{S}}$ s.t. $P(x_{k+1}; \mu) \le P(y_k^{n+1}; \mu)$ *(search step)*
end

The expansion step

Given $\delta \in (0,1)$, point y_k^i, a tentative step $\tilde\alpha_k^i$ and a direction p_k^i $(\pm e^i)$ such that

$$y_k^i \in \mathring{\mathcal{S}}, \quad y_k^i + \tilde\alpha_k^i p_k^i \in \mathring{\mathcal{S}}, \quad P(y_k^i + \tilde\alpha_k^i p_k^i; \mu) \le P(y_k^i;\mu) - \gamma(\tilde\alpha_k^i)^2$$

produce $\alpha_k^i = \tilde\alpha_k^i / \delta^h$ with h smallest integer in $\{0,1,\dots\}$ s.t.

$$y_k^i + \alpha_k^i p_k^i \in \mathring{\mathcal{S}}, \quad \text{and} \quad P(y_k^i + \alpha_k^i p_k^i; \mu) \le P(y_k^i;\mu) - \gamma(\alpha_k^i)^2$$

$$\left\langle \begin{array}{ll} \text{either } y_k^i + \frac{\alpha_k^i}{\delta} p_k^i \in \mathring{\mathcal{S}}, \quad \text{and} & P\left(y_k^i + \frac{\alpha_k^i}{\delta} p_k^i; \mu\right) > P(y_k^i;\mu) - \gamma\left(\frac{\alpha_k^i}{\delta}\right)^2 \\ \text{or} \quad y_k^i + \frac{\alpha_k^i}{\delta} p_k^i \notin \mathring{\mathcal{S}} & \end{array}\right.$$

- α_k^i gives suff. reduction
- $\frac{\alpha_k^i}{\delta}$ gives a “failure”

Convergence result for DFL

It is customary to prove the following

Lemma [Expansion is well-defined]

The expansion step is well defined, i.e. for all i and k it always produces a step size α_k^i

Convergence result for DFL

It is customary to prove the following

Lemma [Expansion is well-defined]

The expansion step is well defined, i.e. for all i and k it always produces a step size α_k^i

Proposition [Stepsizes go to zero]

$$\lim_{k\to\infty} \max_{i=1,\dots,n} \{\alpha_k^i, \tilde{\alpha}_k^i\} = 0$$

Convergence result for DFL

It is customary to prove the following

Lemma [Expansion is well-defined]

The expansion step is well defined, i.e. for all i and k it always produces a step size α_k^i

Proposition [Stepsizes go to zero]

$$\lim_{k\to\infty} \max_{i=1,\dots,n} \{\alpha_k^i, \tilde{\alpha}_k^i\} = 0$$

Theorem [Convergence to stationary points]

Every limit point $\bar{x}$ of $\{x_k\}$ is s.t. $\bar{x} \in \mathring{\mathcal{S}}$ and it is stationary for the log-barrier function, i.e.

$$\nabla_x P(\bar{x};\mu) = 0$$

Main ingredients

To define an IP algorithm converging to KKT points we would need

1. Algorithm DFL, i.e. DF minimization of a smooth function onto an open set $\mathring{\mathcal{S}}$
2. Barrier parameter μ cannot stay fixed to prove convergence
3. Define a rule to produce $\{\mu_k\} \searrow 0$

Basic ideas

The following quantities are at hand

1. $\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\}$ is a rough measure of stationarity for

$$\min_{x \in \overset{\circ}{\mathcal{S}}} P(x; \mu_k)$$

2. μ_k roughly measures the quality of the approximation performed by $P(x; \mu_k)$
3. a rough measure of proximity to the boundary of $\mathcal{S}$ of iterates

$$\min_{j \in \mathcal{I}, i=1,\dots,n+1} \{g_j(y_k^i)\} = (g_{\min})_k$$

Barrier parameter update rule

We need that the measure of stationarity $\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\}$ goes to zero faster than

- μ_k and
- $(g_{\min})_k$

i.e. first order information must be recovered faster than

1. how rapidly precision of the approximation performed by $P(x;\mu_k)$ gets better
2. how rapidly sampled points approach the boundary of $\mathcal{S}$

We propose the following

Barrier parameter update rule

We need that the measure of stationarity $\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\}$ goes to zero faster than

- μ_k and
- $(g_{\min})_k$

i.e. first order information must be recovered faster than

1. how rapidly precision of the approximation performed by $P(x;\mu_k)$ gets better
2. how rapidly sampled points approach the boundary of $\mathcal{S}$

We propose the following

Updating rule

$\mu_{k+1} = \theta\mu_k$, $\theta \in (0,1)$ when

$$\max\{\alpha_k^i, \tilde{\alpha}_k^i\} \le \min\{\mu_k^2, (g_{min})_k^2\}$$

The algorithm

Algorithm 6: `LOG-DFL`

Data: $x_0 \in \overset{\circ}{\mathcal{S}}$, $\mu_0 > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \ldots, n$

for $k = 0, 1, 2, \ldots$ **do**

  Set $y_k^1 = x_k$

  for $i = 1, 2, \ldots, n$ **do**

    if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and* $P(y_k^i; \mu_k)$ *can be suff. reduced along* d_k^i **then**

      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*

    else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \overset{\circ}{\mathcal{S}}$ *and* $P(y_k^i; \mu_k)$ *can be suff. reduced along* $-d_k^i$ **then**

      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*

    else

      Set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$

    end

    Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$ *(failure step)*

  end

The algorithm

Algorithm 7: `LOG-DFL`

Data: $x_0 \in \mathring{\mathcal{S}}$, $\mu_0 > 0$, $d_0^i = e^i$, $\tilde{\alpha}_0^i > 0$, $i = 1, \dots, n$
for $k = 0, 1, 2, \dots$ **do**
  Set $y_k^1 = x_k$
  for $i = 1, 2, \dots, n$ **do**
    if $y_k^i + \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu_k)$ *can be suff. reduced along* d_k^i **then**
      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = d_k^i$ *(LS along d_k^i)*
    else if $y_k^i - \tilde{\alpha}_k^i d_k^i \in \mathring{\mathcal{S}}$ *and* $P(y_k^i; \mu_k)$ *can be suff. reduced along* $-d_k^i$ **then**
      compute α_k^i and set $\tilde{\alpha}_{k+1}^i = \alpha_k^i$, $d_{k+1}^i = -d_k^i$ *(LS along $-d_k^i$)*
    else
      Set $\alpha_k^i = 0$ and $\tilde{\alpha}_{k+1}^i = \theta \tilde{\alpha}_k^i$
    end
    Set $y_k^{i+1} = y_k^i + \alpha_k^i d_k^i$ *(failure step)*
  end
  if $\max_{i=1,\dots,n}\{\alpha_k^i, \tilde{\alpha}_k^i\} \leq \min\{\mu_k^2, (g_{\min})_k^2\}$ **then**
    Set $\mu_{k+1} = \theta \mu_k$, $\theta \in (0, 1)$ *(barrier parameter update)*
  else
    Set $\mu_{k+1} = \mu_k$
  Find x_{k+1} s.t. $P(x_{k+1}; \mu_k) \leq P(y_k^{n+1}; \mu_k)$ *(search step)*
end

Preliminaries

Proposition [Covergence to zero]

Let $\{\mu_k\}$, $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$ be sequences produced by `LOG-DFL` with the updating rule. Then

$$\lim_{k\to\infty} \mu_k = 0 \quad (1)$$

$$\lim_{k\to\infty} \max\{\alpha_k^i, \tilde{\alpha}_k^i\} = 0 \quad (2)$$

Sketch of Proof. First prove (1).

Preliminaries

Proposition [Covergence to zero]

Let $\{\mu_k\}$, $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$ be sequences produced by `LOG-DFL` with the updating rule. Then

$$\lim_{k\to\infty} \mu_k = 0 \tag{1}$$

$$\lim_{k\to\infty} \max\{\alpha_k^i, \tilde{\alpha}_k^i\} = 0 \tag{2}$$

Sketch of Proof. First prove (1). Assume $\mu_k = \bar{\mu}$ for k suff. large

Preliminaries

Proposition [Covergence to zero]

Let $\{\mu_k\}$, $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$ be sequences produced by `LOG-DFL` with the updating rule. Then

$$\lim_{k\to\infty} \mu_k = 0 \tag{1}$$

$$\lim_{k\to\infty} \max\{\alpha_k^i, \tilde{\alpha}_k^i\} = 0 \tag{2}$$

Sketch of Proof. First prove (1). Assume $\mu_k = \bar{\mu}$ for k suff. large

- for k suff. large

$$P(x_{k+1}; \bar{\mu}) \leq P(x_k; \bar{\mu}) \quad \Rightarrow \quad \lim_{k\to\infty} P(x_k; \bar{\mu}) = \bar{P} < +\infty$$

Preliminaries

Proposition [Covergence to zero]

Let $\{\mu_k\}$, $\{\tilde{\alpha}_k^i\}$, $\{\alpha_k^i\}$ be sequences produced by `LOG-DFL` with the updating rule. Then

$$\lim_{k\to\infty} \mu_k = 0 \quad (1)$$

$$\lim_{k\to\infty} \max\{\alpha_k^i, \tilde{\alpha}_k^i\} = 0 \quad (2)$$

Sketch of Proof. First prove (1). Assume $\mu_k = \bar{\mu}$ for k suff. large

- for k suff. large

$$P(x_{k+1}; \bar{\mu}) \leq P(x_k; \bar{\mu}) \quad \Rightarrow \quad \lim_{k\to\infty} P(x_k; \bar{\mu}) = \bar{P} < +\infty$$

- by the updating rule, $(g_{\min})_k \to 0$, hence $x_k \to \bar{x} \in \partial\mathcal{S}$. Hence $P(\bar{x}; \bar{\mu}) = +\infty$

Now, proving (2)

Preliminaries

Proposition [Covergence to zero]

Let $\{\mu_k\}$, $\{\tilde{\alpha}^i_k\}$, $\{\alpha^i_k\}$ be sequences produced by LOG-DFL with the updating rule. Then

$$\lim_{k\to\infty} \mu_k = 0 \tag{1}$$

$$\lim_{k\to\infty} \max\{\alpha^i_k, \tilde{\alpha}^i_k\} = 0 \tag{2}$$

Sketch of Proof. First prove (1). Assume $\mu_k = \bar{\mu}$ for k suff. large

- for k suff. large

$$P(x_{k+1};\bar{\mu}) \leq P(x_k;\bar{\mu}) \quad \Rightarrow \quad \lim_{k\to\infty} P(x_k;\bar{\mu}) = \bar{P} < +\infty$$

- by the updating rule, $(g_{\min})_k \to 0$, hence $x_k \to \bar{x} \in \partial\mathcal{S}$. Hence $P(\bar{x};\bar{\mu}) = +\infty$

Now, proving (2) is straightforward considering again the updating rule

Main theorem

Definition (Mangasarian-Fromowitz C.Q.)

$x \in \mathbb{R}^n$ satisfies the MFCQ if $d \in \mathbb{R}^n$ exists such that

$$\nabla g_i(x)^\top d < 0 \quad \text{for all } i : g_i(x) \leq 0$$

Main theorem

Definition (Mangasarian-Fromowitz C.Q.)

$x \in \mathbb{R}^n$ satisfies the MFCQ if $d \in \mathbb{R}^n$ exists such that

$$\nabla g_i(x)^\top d < 0 \quad \text{for all } i : g_i(x) \leq 0$$

Proposition

Let $\{x_k\}$ be the sequence produced by `LOG-DFL` and assume that every limit point satisfies the MFCQ. Then,

(i) $\left\{ \lambda_i(x_k; \mu_k) = \dfrac{\mu_k}{g_i(x_k)}, \right\}$ for all $i \in \mathcal{I}$ are bounded

(ii) every limit point $\bar{x}$ of $\{x_k\}_K$ $(K = \{k : \ \mu_{k+1} < \mu_k\})$ is a KKT point.

Main theorem

Definition (Mangasarian-Fromowitz C.Q.)

$x \in \mathbb{R}^n$ satisfies the MFCQ if $d \in \mathbb{R}^n$ exists such that

$$\nabla g_i(x)^\top d < 0 \quad \text{for all } i : g_i(x) \leq 0$$

Proposition

Let $\{x_k\}$ be the sequence produced by `LOG-DFL` and assume that every limit point satisfies the MFCQ. Then,

(i) $\left\{ \lambda_i(x_k; \mu_k) = \dfrac{\mu_k}{g_i(x_k)}, \right\}$ for all $i \in \mathcal{I}$ are bounded

(ii) every limit point $\bar{x}$ of $\{x_k\}_K$ $(K = \{k : \ \mu_{k+1} < \mu_k\})$ is a KKT point.

Sketch of Proof. $\{x_k\}_K$ has limit points. We consider one of them $\bar{x}$

$$\lim_{k \to \infty, k \in K'} x_k = \bar{x}$$

Main theorem

Definition (Mangasarian-Fromowitz C.Q.)

$x \in \mathbb{R}^n$ satisfies the MFCQ if $d \in \mathbb{R}^n$ exists such that

$$\nabla g_i(x)^\top d < 0 \quad \text{for all } i : g_i(x) \leq 0$$

Proposition

Let $\{x_k\}$ be the sequence produced by LOG-DFL and assume that every limit point satisfies the MFCQ. Then,

(i) $\left\{ \lambda_i(x_k; \mu_k) = \dfrac{\mu_k}{g_i(x_k)}, \right\}$ for all $i \in \mathcal{I}$ are bounded

(ii) every limit point $\bar{x}$ of $\{x_k\}_K$ $(K = \{k : \ \mu_{k+1} < \mu_k\})$ is a KKT point.

Sketch of Proof. $\{x_k\}_K$ has limit points. We consider one of them $\bar{x}$

$$\lim_{k \to \infty, k \in K'} x_k = \bar{x}$$

To prove (ii), from suff. decrease and M.V. theorem we get

$$\nabla P(u_k^i; \mu_k)^\top d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i}$$

Main theorem

Definition (Mangasarian-Fromowitz C.Q.)

$x \in \mathbb{R}^n$ satisfies the MFCQ if $d \in \mathbb{R}^n$ exists such that

$$\nabla g_i(x)^\top d < 0 \quad \text{for all } i : g_i(x) \leq 0$$

Proposition

Let $\{x_k\}$ be the sequence produced by `LOG-DFL` and assume that every limit point satisfies the MFCQ. Then,

(i) $\left\{\lambda_i(x_k;\mu_k) = \dfrac{\mu_k}{g_i(x_k)},\right\}$ for all $i \in \mathcal{I}$ are bounded

(ii) every limit point $\bar{x}$ of $\{x_k\}_K$ $(K = \{k : \ \mu_{k+1} < \mu_k\})$ is a KKT point.

Sketch of Proof. $\{x_k\}_K$ has limit points. We consider one of them $\bar{x}$

$$\lim_{k\to\infty, k\in K'} x_k = \bar{x}$$

To prove (ii), from suff. decrease and M.V. theorem we get

$$\nabla P(u_k^i;\mu_k)^\top d^i \geq -\frac{o(\xi_k^i)}{\xi_k^i}$$

$u_k^i \to \bar{x}$. We obtain stationarity by using (i)

Extensions

We considered problem

$$\min_{x\in\mathbb{R}^n} \ f(x)$$
$$s.t.\ g_i(x) \geq 0, \quad i \in \mathcal{I}$$

Extensions

We considered problem

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & f(x) \\ s.t. \ & g_i(x) \geq 0, \quad i \in \mathcal{I} \end{aligned}$$

but the more general problem can be considered

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & f(x) \\ s.t. \ & g_i(x) \geq 0, \quad i \in \mathcal{I} \\ & g_i(x) = 0, \quad i \in \mathcal{E} \\ & \ell \leq x \leq u \end{aligned}$$

using a mixed log-barrier sequential penalty approach preserving the convergence results and explicitly handling the box constraints

Extensions

We considered problem

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \ & f(x)\\ s.t. \ & g_i(x) \geq 0, \quad i \in \mathcal{I}\end{aligned}$$

but the more general problem can be considered

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \ & f(x)\\ s.t. \ & g_i(x) \geq 0, \quad i \in \mathcal{I}\\ & g_i(x) = 0, \quad i \in \mathcal{E}\\ & \ell \leq x \leq u\end{aligned}$$

using a mixed log-barrier sequential penalty approach preserving the convergence results and explicitly handling the box constraints

Note that, g_i, $i \in \mathcal{I}$ s.t. $g_i(x_0) \leq 0$ can be considered

Problems selection

Criteria for problems selection

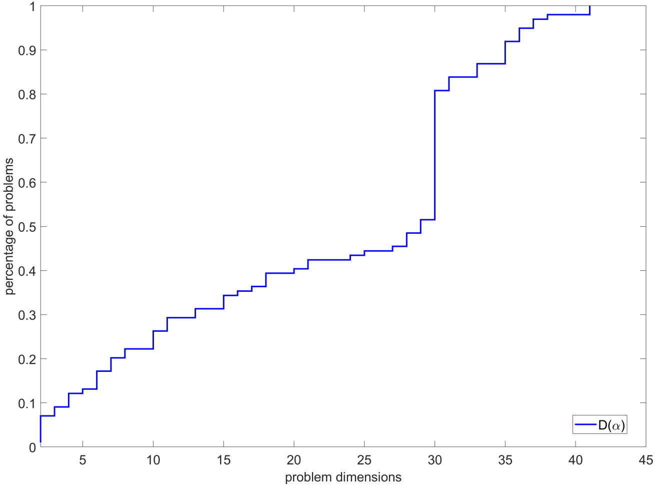
- problems from the CUTEst collection
- with both inequalities and equalities (see previous slide)
- x_0 such that $g_i(x_0) > 0$ for (at least one) $i \in \mathcal{I}$

This gives us $N = 99$ problems with

- $n \in [2, 41]$ variables
- $m \in [1, 144]$ constraints

Problems selection

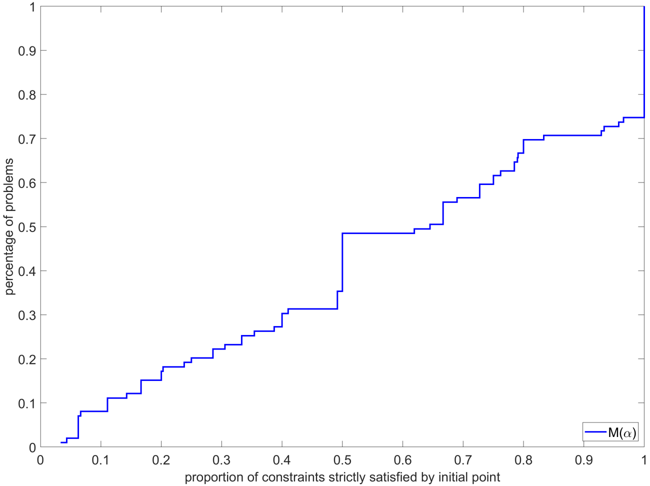
Cumulative distribution of the number of variables $D(\alpha) = \frac{1}{N} \left|\{p : n_p \leq \alpha\}\right|$



Problems selection

Cumulative distribution of the ratio of constraints strictly satisfied at the initial point

$$M(\alpha) = \frac{1}{N}\left|\{p : \frac{\bar{m}_p}{m_p} \leq \alpha\}\right|$$



Comparison with NOMAD

We run NOMAD (3.9.1)[1] [1] using default settings except for `constraint type`

- EB for g_i such that $g_i(x_0) > 0$,
- PEB otherwise

We use performance and data profiles ([Wild, Moré, SIOPT'09]). Stopping criterion:

$$f_k \leq f_L + \tau(\hat{f}(x_0) - f_L),$$

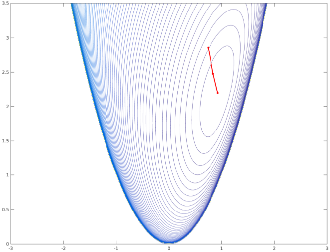
- τ is a given tolerance
- f_L smallest f.value computed by all the solvers with 20000 fun.evals
- $\hat{f}(x_0)$ obj. value of the worst feasible point found by any solver

[1] S. Le Digabel. Algorithm 909: NOMAD: Nonlinear Optimization with the MADS algorithm. ACM Transactions on Mathematical Software, 37(4):44:1–44:15, 2011.

[1] We are aware of the new NOMAD (4.1.0) and we plan to use it

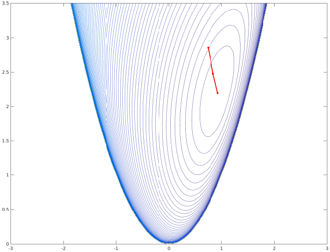
Heuristics within LOG-DFL

1. Use a further direction d_μ defined using two consecutive points where μ updated
   - it should be a good descent direction
   - it points toward the "central path"



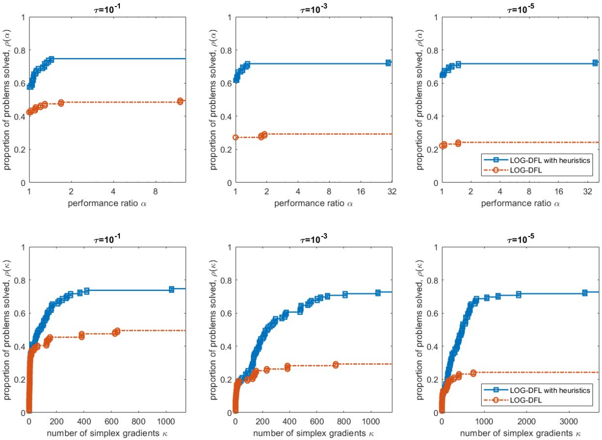
Heuristics within LOG-DFL

1. Use a further direction d_μ defined using two consecutive points where μ updated
   - it should be a good descent direction
   - it points toward the "central path"



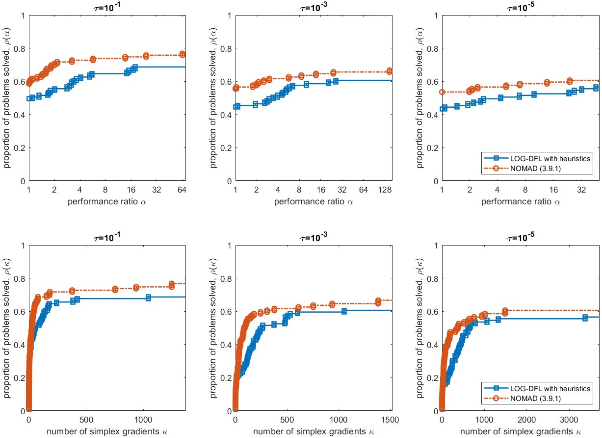
2. Mimic the behavior of the PEB constraint type in NOMAD
   - initially violated constraints are handled by a sequential exterior approach
   - when one of them becomes feasible, we switch to interior penalization

Heuristics within LOG-DFL



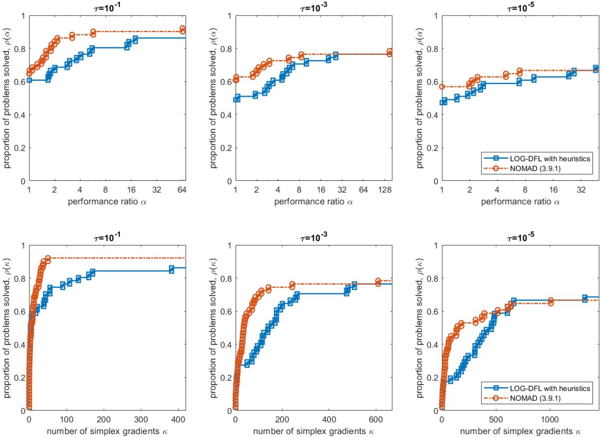
Comparison with NOMAD

Results on the entire test set of problems



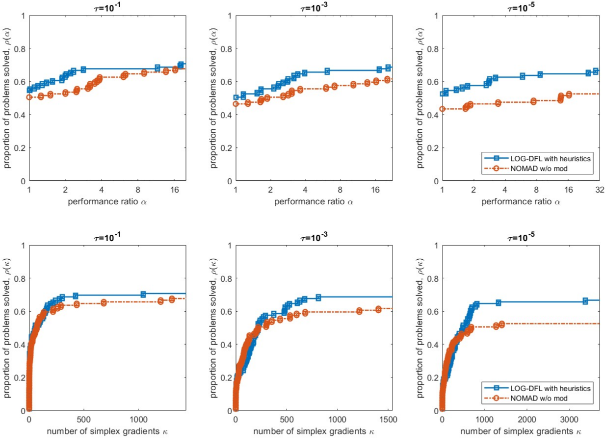
Comparison with NOMAD

Results on problems where both methods find a feasible solution



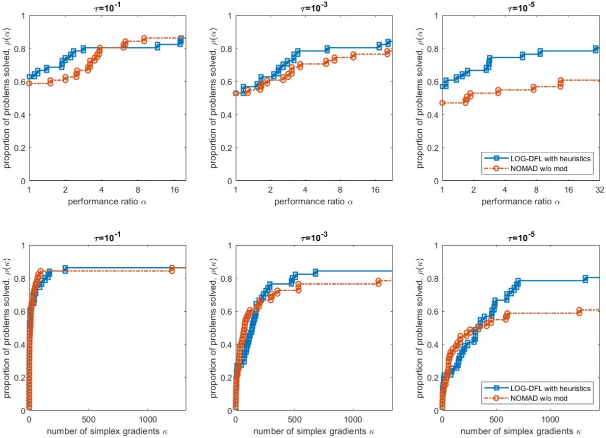
Comparison with `NOMAD` w/o models

Results on the entire test set of problems



Comparison with NOMAD w/o models

Results on problems where both methods find a feasible solution



Conclusions

We presented `LOG-DFL`

- a DF method based on a log-barrier penalty function
- convergence to stationary points w/o using dense sets of directions
- good preliminary numerical results and comparison
- `LOG-DFL` has been coded in Python and is available for free on the Derivative-Free Library (DFL)
  `http://www.iasi.cnr.it/~liuzzi/DFL/`

Future work

- extend the approach to nonsmooth problems

Conclusions

We presented LOG-DFL

- a DF method based on a log-barrier penalty function
- convergence to stationary points w/o using dense sets of directions
- good preliminary numerical results and comparison
- LOG-DFL has been coded in Python and is available for free on the Derivative-Free Library (DFL)
  http://www.iasi.cnr.it/∼liuzzi/DFL/

Future work

- extend the approach to nonsmooth problems

Thank you for your attention!