Andy’s Early Work: 1971 – 1982

ICCOPT Berlin 2019
A Gradient Type Method for Locating Constrained Extrema

Adviser: Tomasz Pietrzykowski

Research Area: Exact Penalization in Nonlinear Programming

The extension of steepest descent to nonsmooth optimization and the origins of vertical and horizontal steps.
Exact Penalization

NLP minimize $f(x)$
subject to $\phi_i(x) \leq 0$  $i = 1, \ldots, k$
$\phi_i(x) = 0$  $i = k + 1, \ldots, \ell$

where $f$ and all $\phi_i$ are continuous mappings from $\mathbb{R}^n$ to $\mathbb{R}$.

Feasible region:

$\mathcal{F} := \left\{ x \mid \begin{array}{c}
\phi_i(x) \leq 0, \ i = 1, \ldots, k, \\
\phi_i(x) = 0, \ i = k + 1, \ldots, \ell
\end{array} \right\}$
Exact Penalization

NLP minimize $f(x)$
subject to $\phi_i(x) \leq 0 \quad i = 1, \ldots, k$
$\phi_i(x) = 0 \quad i = k + 1, \ldots, \ell$

where $f$ and all $\phi_i$ are continuous mappings from $\mathbb{R}^n$ to $\mathbb{R}$.

Feasible region:

$$\mathcal{F} := \left\{ x \left| \begin{array}{l}
\phi_i(x) \leq 0, \ i = 1, \ldots, k,
\phi_i(x) = 0, \ i = k + 1, \ldots, \ell
\end{array} \right. \right\}$$

$l_1$ Exact Penalization:

$$l_1-\text{NLP} \quad \min p_\mu(x) := \mu f(x) + \sum_{i=1}^{k} \max(0, \phi_i(x)) + \sum_{i=k+1}^{\ell} |\phi_i(x)|$$
Thm: (CQ) If $\bar{x}$ solves NLP, then, for all $\mu > 0$ small, $\bar{x}$ solves $\ell_1$-NLP:

$$\ell_1 - \text{NLP} \quad \min_x p_\mu(x) := \mu f(x) + \sum_{i=1}^{k} \max(0, \phi_i(x)) + \sum_{i=k+1}^{\ell} |\phi_i(x)| .$$
Theoretical Foundations

**Thm:** (CQ) If \( \bar{x} \) solves NLP, then, for all \( \mu > 0 \) small, \( \bar{x} \) solves \( \ell_1 \)-NLP:

\[
\ell_1 - \text{NLP} \quad \min_x p_{\mu}(x) := \mu f(x) + \sum_{i=1}^{k} \max(0, \phi_i(x)) + \sum_{i=k+1}^{\ell} |\phi_i(x)|.
\]

**Convex Case** (finite-valued): Eremin (1966), Zangwill (1967)

Slater CQ: \( \phi_i \) are affine for \( i = k + 1, \ldots, \ell \) and

\[
\exists \hat{x} \in \mathcal{F} \text{ such that } \phi_i(\hat{x}) < 0, \ i = 1, \ldots, k.
\]
Theoretical Foundations

**Thm:** (CQ) If $\bar{x}$ solves NLP, then, for all $\mu > 0$ small, $\bar{x}$ solves $\ell_1$-NLP:

$$\ell_1 - \text{NLP} \quad \min_x p_\mu(x) := \mu f(x) + \sum_{i=1}^k \max(0, \phi_i(x)) + \sum_{i=k+1}^\ell |\phi_i(x)|.$$  

**Convex Case** (finite-valued): Eremin (1966), Zangwill (1967)

Slater CQ: $\phi_i$ are affine for $i = k + 1, \ldots, \ell$ and

$$\exists \hat{x} \in \mathcal{F} \text{ such that } \phi_i(\hat{x}) < 0, i = 1, \ldots, k.$$  

**Smooth Case:** Pietrzykowski (1969)

(LICQ): The active constraint gradients,

$$\nabla \phi_i(x) \quad i \in A(x,0), \quad \text{are linearly independent,}$$

where, for $\varepsilon \geq 0$,

$$A(x, \varepsilon) := \{ i \mid |\phi_i(x)| \leq \varepsilon, \ i \in \{1, \ldots, k\} \}$$

are the $\varepsilon$-active constraints.
Vertical and Horizontal Steps

*Constrained Optimization Using a Nondifferentiable Penalty Function,*
SIAM J. Numerical Analysis, 10(1973)760–784.

*Linear Programming via a Nondifferentiable Penalty Function*

*A Penalty Function Method Converging Directly to a Constrained Optimum*
with Tomasz Pietrzykowski
Vertical and Horizontal Steps

For simplicity assume $\mathcal{F} := \{x \mid \phi_i(x) \leq 0, i = 1, \ldots, \ell\}$.

$$A(x, \varepsilon) := \{i \mid ||\phi_i(x)|| \leq \varepsilon, \ i \in \{1, \ldots, \ell\}\} \quad \varepsilon\text{-active}$$

$$I(x, \varepsilon) := \{i \mid ||\phi_i(x)|| > \varepsilon, \ i \in \{1, \ldots, \ell\}\} \quad \varepsilon\text{-inactive}$$

$$\hat{I}(x, \varepsilon) := I(x, \varepsilon) \cap \{i \mid \phi_i(x) > 0, \ i = 1, \ldots, \ell\} \quad \text{infeas. \varepsilon\text{-inactive}}$$

Keys: The construction of $P$ and the evaluation of $\sigma, \tau \geq 0$. 
Vertical and Horizontal Steps

For simplicity assume $ \mathcal{F} := \{ x | \phi_i(x) \leq 0, i = 1, \ldots, \ell \}$.

$A(x, \varepsilon) := \{ i | \| \phi_i(x) \| \leq \varepsilon, i \in \{1, \ldots, \ell\} \}$  \hspace{1cm} \varepsilon$-active

$I(x, \varepsilon) := \{ i | \| \phi_i(x) \| > \varepsilon, i \in \{1, \ldots, \ell\} \}$  \hspace{1cm} \varepsilon$-inactive

$\hat{I}(x, \varepsilon) := I(x, \varepsilon) \cap \{ i | \phi_i(x) > 0, i = 1, \ldots, \ell \}$  \hspace{1cm} $\varepsilon$-inactive

“Steepest Descent” for $p_\mu$: $r(x, \varepsilon) := -\mu \nabla f(x) - \sum_{i \in \hat{I}(x, \varepsilon)} \nabla \phi_i(x)$

Keys: The construction of $P$ and the evaluation of $\sigma, \tau \geq 0$. 
Vertical and Horizontal Steps

For simplicity assume $\mathcal{F} := \{x \mid \phi_i(x) \leq 0, \ i = 1, \ldots, \ell \}$.

$$A(x, \varepsilon) := \{i \mid ||\phi_i(x)|| \leq \varepsilon, \ i \in \{1, \ldots, \ell\}\}$$ \hspace{1em} \varepsilon\text{-active}

$$I(x, \varepsilon) := \{i \mid ||\phi_i(x)|| > \varepsilon, \ i \in \{1, \ldots, \ell\}\}$$ \hspace{1em} \varepsilon\text{-inactive}

$$\hat{I}(x, \varepsilon) := I(x, \varepsilon) \cap \{i \mid \phi_i(x) > 0, \ i = 1, \ldots, \ell\}$$ \hspace{1em} infeas. \varepsilon\text{-inactive}

“Steepest Descent” for $p_\mu$: $r(x, \varepsilon) := -\mu \nabla f(x) - \sum_{i \in \hat{I}(x, \varepsilon)} \nabla \phi_i(x)$

Let $P$ be (almost) the projection onto the subspace orthogonal to the \varepsilon-active constraint gradients:

$$\text{Span}[\{\nabla \phi_i(x) \mid i \in A(x, \varepsilon)\}]^\perp.$$

Keys: The construction of $P$ and the evaluation of $\sigma, \tau \geq 0.$
Vertical and Horizontal Steps

For simplicity assume $\mathcal{F} := \{ x \mid \phi_i(x) \leq 0, i = 1, \ldots, \ell \}$.

$$A(x, \varepsilon) := \{ i \mid |\phi_i(x)| \leq \varepsilon, \ i \in \{1, \ldots, \ell\} \} \quad \varepsilon\text{-active}$$

$$I(x, \varepsilon) := \{ i \mid |\phi_i(x)| > \varepsilon, \ i \in \{1, \ldots, \ell\} \} \quad \varepsilon\text{-inactive}$$

$$\hat{I}(x, \varepsilon) := I(x, \varepsilon) \cap \{ i \mid \phi_i(x) > 0, \ i = 1, \ldots, \ell \} \quad \text{infeas. } \varepsilon\text{-inactive}$$

“Steepest Descent” for $p_\mu$: $r(x, \varepsilon) := -\mu \nabla f(x) - \sum_{i \in \hat{I}(x, \varepsilon)} \nabla \phi_i(x)$

Let $P$ be (almost) the projection onto the subspace orthogonal to the $\varepsilon$-active constraint gradients:

$$\text{Span}[\{\nabla \phi_i(x) \mid i \in A(x, \varepsilon)\}]^\perp.$$

$$h(x, \varepsilon) := P r(x, \varepsilon) \quad \text{the horizontal step}$$

$$v(x, \varepsilon) := (I - P)r(x, \varepsilon) \quad \text{the vertical step}$$

$$w(x, \varepsilon) := \sigma v(x, \varepsilon) + \tau v(x, \varepsilon) \quad \text{the step}$$

Keys: The construction of $P$ and the evaluation of $\sigma, \tau \geq 0$. 
Extensions

UV-decompositions are an example of recent ideas in this direction, where the horizontal step is in the U direction and the vertical step is in the V direction.


Second-Order Theory and Algorithms

*Second-Order Conditions for and Exact Penalty Function*  
with Tom Coleman  

*Nonlinear Programming via and Exact Penalty Function:  
Asymptotic Analysis*  
with Tom Coleman  

*Nonlinear Programming via and Exact Penalty Function:  
Global Analysis*  
with Tom Coleman  
Theory:
Andy and Tom established second-order necessary and sufficient conditions for the $\ell_1$ exact penalty function using techniques from NLP under LICQ.

- The theory applies at both feasible and infeasible points.
- When feasible, they show equivalence with the NLP strong second-order theory.
Second-Order Theory and Algorithms

Algorithms:
Again, the basic idea rests on the notion of vertical and horizontal steps.

But now the horizontal step $h^k$ is based on a second-order approximation to the Lagrangian over the subspace perpendicular to the active constraint gradients.

Multiplier estimates are given by a least-squares solution to the first-order optimality conditions.
Second-Order Theory and Algorithms

**Algorithms:**
Again, the basic idea rests on the notion of vertical and horizontal steps.

But now the horizontal step $h^k$ is based on a second-order approximation to the Lagrangian over the subspace perpendicular to the active constraint gradients.

Multiplier estimates are given by a least-squares solution to the first-order optimality conditions.

Once the second-order step is chosen, a vertical step $v^k$ is chosen at the point $x^k + h^k$ using the data at $x^k$ to give the final step $x^k + h^k + v^k$. 

This work is one of the initial contributions toward second-order correction steps (Fletcher) to overcome the Marotos effect.
Second-Order Theory and Algorithms

Algorithms:
Again, the basic idea rests on the notion of vertical and horizontal steps.

But now the horizontal step $h^k$ is based on a second-order approximation to the Lagrangian over the subspace perpendicular to the active constraint gradients.

Multiplier estimates are given by a least-squares solution to the first-order optimality conditions.

Once the second-order step is chosen, a vertical step $v^k$ is chosen at the point $x^k + h^k$ using the data at $x^k$ to give the final step $x^k + h^k + v^k$.

This work is one of the initial contributions toward second-order correction steps (Fletcher) to overcome the Marotos effect.
Second-Order Theory and Algorithms

Convergence Theory:

Local: Andy and Tom establish the two step local super-linear convergence of their method under a strong second-order sufficiency.

Global:
• A break-point line-search procedure is introduced to ensure global convergence.
• Under a strong second-order sufficiency condition, the Newton step is accepted and two step super-linear convergence is achieved.
Thank You Andy!!

An inspiring leader, mentor, community builder, and researcher.