# Algorithms for nonlinear programming problems I

Martin Branda

Charles University in Prague
Faculty of Mathematics and Physics
Department of Probability and Mathematical Statistics

Computational Aspects of Optimization

# Algorithm classification

- **Order of derivatives[1]:** derivative-free, first order (gradient), second-order (Newton)
- **Feasibility of the constructed points**: interior and exterior point methods
- **Deterministic/randomized**
- **Local**/global

[1] If possible, deliver the derivatives.

# Unconstrained problems

Let $f : \mathbb{R}^n \to \mathbb{R}$, $x^0$ be a starting point, $d^k \in \mathbb{R}^n$ be a **descent direction**, and $\lambda \in \mathbb{R}$ be a **step length**.

Find a **descent direction** $d^k$, solve the **line search problem**

$$\lambda^k = \arg \min_{0 \le \lambda \le \lambda_{max}} f(x^k + \lambda d^k)$$

and set

$$x^{k+1} = x^k + \lambda^k d^k.$$

Iterate until a convergence criterion is not satisfied, e.g. $\left\| d^k \right\| < \varepsilon$ or $|f(x^k) - f(x^{k+1})| < \varepsilon$.

# Review of line search methods

Bazaraa et al. (2006):

- Derivative-free: dichotomous search, golden section method, Fibonacci search
- Using derivatives: bisection search, Newton's method

# Descent directions – Steepest descent

A vector $d$ is called a descent direction of a function $f$ at $x$ if there exists a $\delta > 0$ such that

$$f(x + \lambda d) < f(x), \ \lambda \in (0, \delta).$$

**Steepest descent** $d$ with $\|d\| = 1$ minimizes the limit

$$f'(x; d) := \lim_{\lambda \to 0_+} \frac{f(x + \lambda d) - f(x)}{\lambda} < 0.$$

If $f$ is differentiable at $x$ with a nonzero gradient, then

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$$

leading to the gradient (Cauchy) method.

$$f'(x; d) = \nabla f(x)^T d.$$

# Descent directions

If we set

$$h(\lambda) := f(x + \lambda d),$$

then

$$h'(0) = \nabla f(x)^T d.$$

$h$ is decreasing $\Leftrightarrow$ $f$ is decreasing in direction $d$.

# Descent directions

Steepest descent – works well during the early steps, the **zigzagging** phenomenon often appears in later steps, see Bazaraa et al. (2006), Example 8.6.2

# Descent directions – Newton direction

Approximation of $f$ by a limited Taylor expansion around $x^k$

$$g(x) := f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k)$$

Setting $\nabla_x g(x) = 0$, we obtain the **Newton direction**

$$d = -\left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k).$$

If $\nabla^2 f(x^k) > 0$, then $d$ is a descent direction[2].

[2] In general, $d = -A\nabla f(x^k)$ for $A > 0$ is a descent direction $\to$ **Quasi-Newton methods**.

# Descent directions – Newton direction

**Convergence of the algorithm**: Bazaraa et al. (2006), Theorem 8.6.5 ($f \in C^2$, $\nabla f(\overline{x}) = 0$ and $\nabla^2 f(\overline{x}) > 0$ at a local minimum $\overline{x}$, starting point is sufficiently close.)

# Descent directions – Example

$$\min_{x,y} (x-y)^4 + 2x^2 + y^2 - x + 2y$$

Partial derivatives

$$\begin{aligned} \frac{\partial f(x,y)}{\partial x} &= 4(x-y)^3 + 4x - 1 = 0, \\ \frac{\partial f(x,y)}{\partial y} &= -4(x-y)^3 + 2y + 2 = 0. \end{aligned} \tag{1}$$

Second-order partial derivatives

$$\begin{aligned} \frac{\partial^2 f(x,y)}{\partial x^2} &= 12(x-y)^2 + 4, \\ \frac{\partial^2 f(x,y)}{\partial x \partial y} &= -12(x-y)^2, \\ \frac{\partial^2 f(x,y)}{\partial y^2} &= 12(x-y)^2 + 2. \end{aligned} \tag{2}$$

Compare directions $\nabla f(x)$ and $d = -\left(\nabla^2 f(x)\right)^{-1} \nabla f(x)$ ...

# Conjugate gradient method

Nocedal and Wright (2006), Chapter 5: Consider (unconstrained) quadratic programming problem

$$\min \frac{1}{2} x^T A x - b^T x.$$

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. We say that vectors $p^1, \ldots p^n$ are conjugate with respect to $A$ if

$$(p^i)^T A p^j = 0 \text{ for all } i \neq j.$$

If we set $x^{k+1} = x^k + \alpha^k p^k$, where

$$\begin{aligned} r^k &= A x^k - b, \\ \alpha^k &= - \frac{r^{k\,T} p^k}{(p^k)^T A p^k}, \end{aligned} \qquad (3)$$

then $x^{n+1}$ is an optimal solution.

# Method of Zoutendijk

Bazaraa et al. (2006), Section 10.1: $f : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^n \to \mathbb{R}$ **differentiable**

$$\min_x f(x) \text{ s.t. } g_j(x) \le 0,\ j = 1, \dots, m.$$

(Extension including equality constraints is possible.)

Method based on **improving feasible directions** (remember the "directional" optimality conditions).

# Method of Zoutendijk

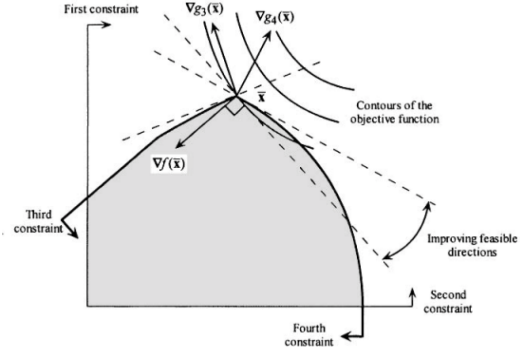
0. Start with a **feasible** $x^1$. For $k = 1, \dots (, K_{max})$ do
1. Set $J(x^k) = \{j : g_j(x^k) = 0\}$ and solve **linear programming problem for finding a direction**:

$$\begin{aligned}
\min_{z,d} \ & z \\
\text{s.t. } & \nabla f(x^k)^T d \le z, \\
& \nabla g_j(x^k)^T d \le z, \ j \in J(x^k), \\
& -1 \le d_i \le 1, \ i = 1, \dots, n.
\end{aligned}$$

Denote by $(z^k, d^k) \in \mathbb{R}^{1+n}$ the optimal solution.

- If $z^k = 0$ then STOP (We have found a Fritz-John point).
- Else if $z^k < 0$ then continue with STEP 2.

# Method of Zoutendijk




Bazaraa et al. (2006)

# Method of Zoutendijk

2. Find **maximal possible step**

$$\lambda_{max} := \sup\{\lambda : \ g_j(x^k + \lambda d^k) \leq 0, j = 1, \dots, m\},$$

solve the **line search problem**

$$\lambda^k = \arg \min_{0 \leq \lambda \leq \lambda_{max}} f(x^k + \lambda d^k)$$

and set

$$x^{k+1} = x^k + \lambda^k d^k.$$

Continue with STEP 1.

# Method of Zoutendijk

*Where could be a problem?* Direction as well as line search need not to be closed...

**Convergence:** Bazaraa et al. (2006), part 10.2.

# Method of Zoutendijk – example

Bazaraa et al. (2006), Example 10.1.8

$$\begin{aligned} \min\ & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\ \text{s.t. } & x_1 + x_2 \leq 5, \\ & 2x_1^2 - x_2 \leq 0, \\ & -x_1 \leq 0, \\ & -x_2 \leq 0. \end{aligned} \tag{4}$$

$$\nabla f(x) = (4x_1 - 2x_2 - 4,\ 4x_2 - 2x_1 - 6)^T \tag{5}$$

# Method of Zoutendijk – Example

Starting point $x^0 = (0, 0.75)^T$, $\nabla f(x^0) = (-5.5, -3)^T$, $J(x^0) = \{3\}$. The direction finding problem is then

$$\begin{aligned} &\min\ z \\ &\text{s.t. } -5.5d_1 - 3d_2 \le z, \\ &\qquad -d_1 \le z, \\ &\qquad -1 \le d_1, d_2 \le 1. \end{aligned} \tag{6}$$

with optimal solution $d^1 = (1, -1)$, $z^1 = -1$.

# Method of Zoutendijk – Example

Then

$$x^0 + \lambda d^1 = (\lambda,\ 0.75 - \lambda)$$

and

$$f(x^0 + \lambda d^1) = 6\lambda^2 - 2.5\lambda - 3.375.$$

Maximize it over the set of feasible solutions $M$ to obtain $\lambda_{max} = 0.4114$. Finally

$$\begin{aligned} &\min\ 6\lambda^2 - 2.5\lambda - 3.375 \\ &\text{s.t. } 0 \le \lambda \le \lambda_{max}. \end{aligned} \tag{7}$$

$\lambda^1 = 0.2083.$

# Cutting plane method

$f : \mathbb{R}^n \to \mathbb{R}$, $g_j : \mathbb{R}^n \to \mathbb{R}$

$$\min_x f(x) \text{ s.t. } g_j(x) \leq 0,\ j = 1, \dots, m.$$

Denote $M = \{x \in \mathbb{R} :\ g_j(x) \leq 0,\ j = 1, \dots, m\}$.

**ASS.** $f$ is affine, $g$ are convex and differentiable, $M$ is compact.

# Cutting plane method

0. Start with a polyhedral set $M^0$ such that $M \subset M^0$, e.g. a box $M^0 = [lb_1, ub_1] \times \cdots \times [lb_m, ub_m]$. For $k = 0, \dots (, K_{max})$ do
1. Solve the **linear programming problem**

$$\min_x f(x) \text{ s.t. } x \in M^k,$$

and obtain $x^k \in M^k$. If $x^k \in M$, then STOP, we have found an optimal solution. Otherwise continue with STEP 2.

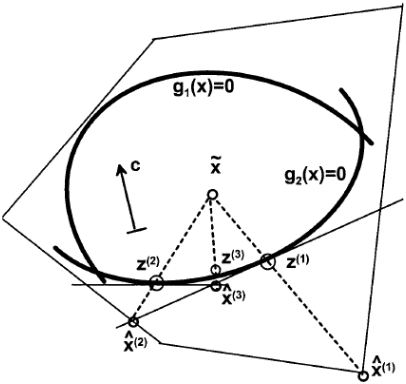
2. If $x^k \notin M$, then find $j^k = \arg\max_j g_j(x^k)$, construct a **cutting plane** and set

$$M^{k+1} = M^k \cap \left\{ x \in \mathbb{R} : \; g_{j^k}(x^k) + \nabla g_{j^k}(x^k)^T (x - x^k) \leq 0 \right\}.$$

Note that $x^k$ violates the cut, and no $x \in M$ is cut off[3] (compare with the integer programming cuts). Return to STEP 1.

[3] From convexity $g_{j^k}(x^k) + \nabla g_{j^k}(x^k)^T (x - x^k) \leq g_{j^k}(x) \leq 0$.

# Cutting plane method




Kall and Mayer (2005).

# Cutting plane method – Example

$$\begin{aligned}\min_x \ & -x_1 - x_2 \\ \text{s.t. } & x_1^2 + x_2^2 - 1 \leq 0, \\ & x_1, x_2 \geq 0.\end{aligned}$$

Set $M = \{(x_1, x_2) : \ x_1^2 + x_2^2 - 1 \leq 0, x_1, x_2 \geq 0\}$, $\nabla g(x)^T = (2x_1, 2x_2)$.

# Cutting plane method – Example

0. Set $M^0 = [0,1]^2$.
1. Solve $\min_x -x_1 - x_2$ s.t. $x \in M^0$ with optimal solution $x^0 = (1,1)^T$.
2. Since $x^0 \notin M$, construct the cut

$$g(x^0) + \nabla g(x^0)^T (x - x^0) \leq 0,$$

and set

$$M^1 = M^0 \cap \{(x_1, x_2) : \; x_1 + x_2 \leq 3/2\}.$$

Continue with STEP 1.

# Penalty method

**Perfect penalty** (rather theoretical)

$$PP(x) = \begin{cases} 0 & \text{if } g(x) \le 0, h(x) = 0, \\ \infty & \text{otherwise.} \end{cases}$$

Compare with Lagrangian duality (sup over multipliers).

The following problem is equivalent to the original constrained one.

$$\min_x f(x) + PP(x).$$

# Penalty method

$L_{p,q}-$ **penalty function**:

$$PF_N(x) = N \cdot \left( \sum_{j=1}^{m} [g_j(x)]_+^p + \sum_{i=1}^{l} |h_i(x)|^q \right),$$

where $N$ is the penalty parameter, $[\cdot]_+ = \max\{\cdot, 0\}$.

More general penalty using $\Phi(y) = 0$ for $y \leq 0$ and $\Phi(y) > 0$ for $y > 0$ and $\Psi(y) = 0$ for $y = 0$ and $\Psi(y) > 0$ for $y \neq 0$.

# Penalty method

**Algorithm:**

0. Set $\varepsilon > 0$, $N^1 > 0$, $\beta > 1$. For $k = 1, \dots (, K_{max})$ do:
1. Solve
$$\min_x f(x) + PF_{N^k}(x).$$
and obtain $x^k$
2. IF $PF_{N^k}(x^k) < \varepsilon$, then STOP. ELSE set $N^{k+1} = N^k \cdot \beta$ and continue with STEP 1.

Exterior point method!

# Penalty method

Convergence of the method: Bazaraa et al. (2006), Theorem 9.2.2 (continuous $f, g_j, h_i$, $x_k \in X$ compact).

# Penalty functions – Example

Consider

$$\begin{aligned} \min\ & x_1^2 + x_2^2 \\ \text{s.t. } & x_1 + x_2 = 2. \end{aligned}$$

with optimal solution $\hat{x}_1 = \hat{x}_2 = 1$. Penalty function problem

$$\min x_1^2 + x_2^2 + N(x_1 + x_2 - 2)^2.$$

Using optimality conditions

$$\hat{x}_1^N = \hat{x}_2^N = \frac{2N}{2N+1}.$$

# Penalty method

Remarks

- **Sequential Unconstrained Minimization** (SUMT): optimal solution $x^k$ is used as a starting point in the next iteration[4] to solve the penalty problem with $N_{k+1}$.
- **Exact penalty**: Instead of $N \to \infty$ it is sufficient to converge $N \to \overline{N} < \infty$ (numerically more stable).

[4] "warm starting"

# Augmented Lagrangian Method

Nocedal and Wright (2006), Section 17.3: $f : \mathbb{R}^n \to \mathbb{R}$, $h_i : \mathbb{R}^n \to \mathbb{R}$ **differentiable**

$$\min_x \ f(x)$$
$$\text{s.t. } h_i(x) = 0, \ i = 1, \dots, l.$$

(Extension including inequality constraints is possible.)

$$L(x, v) = f(x) - \sum_{i=1}^{l} v_i h_i(x).$$

# Augmented Lagrangian Method

Augmented Lagrangian function – combination of the **Lagrangian function** with the **quadratic penalty term**

$$L_A(x,\lambda,\mu) = f(x) - \sum_{i=1}^{l} \lambda_i h_i(x) + \frac{\mu}{2} \sum_{i=1}^{l} (h_i(x))^2 .$$

$$\begin{aligned} \nabla_x L_A(x,\lambda,\mu) &= \nabla_x f(x) - \sum_{i=1}^{l} \lambda_i \nabla_x h_i(x) + \mu \sum_{i=1}^{l} h_i(x) \nabla_x h_i(x) \\ &= \nabla_x f(x) - \sum_{i=1}^{l} \left(\lambda_i - \mu h_i(x)\right) \nabla_x h_i(x). \end{aligned}$$

We have that $v_i \approx \lambda_i - \mu h_i(x)$.

# Augmented Lagrangian Method

0. Set initial $\mu^1 > 0$, $\beta > 1$ and $\lambda^1$. Select a tolerance $\varepsilon > 0$. For $k = 1, \ldots (, K_{max})$ do:
1. Solve unconstrained problem

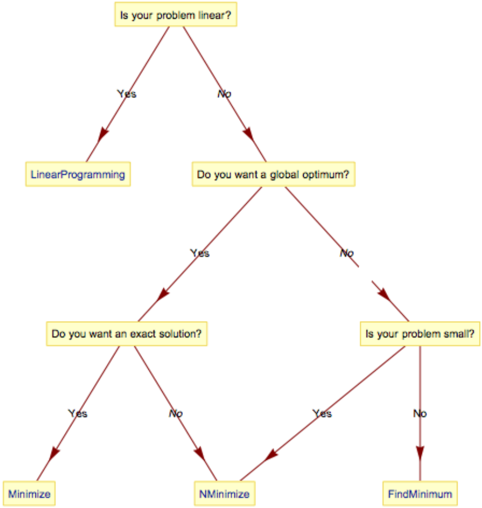
$$\min_x L_A(x, \lambda^k, \mu^k)$$

   and obtain $x^k$. If $\left\| L_A(x^k, \lambda^k, \mu^k) \right\| \le \varepsilon$, STOP. Otherwise continue with STEP 2.
2. Update the Lagrange multipliers $\lambda_i^{k+1} = \lambda_i^k - \mu^k h_i(x^k)$ and the penalty parameter $\mu^{k+1} = \beta \mu^k$. Go to STEP 1.

# Augmented Lagrangian Method

Convergence of the algorithm: Nocedal and Wright (2006), Theorem 17.5 (LICQ, SOSC).

# Mathematica – Solver Decision Tree



# Literature


- Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (2006). **Nonlinear programming: theory and algorithms**, Wiley, Singapore, 3rd edition.
- Boyd, S., Vandenberghe, L. (2004). **Convex Optimization**. Cambridge University Press, Cambridge.
- P. Kall, J. Mayer: **Stochastic Linear Programming: Models, Theory, and Computation**. Springer, 2005.
- Nocedal, J., Wright, J.S. (2006). **Numerical optimization**. Springer, New York, 2nd edition.