# Fundamentals of Algorithms for Non-linear Constrained Optimisation

Saurav Samantaray

Department of Mathematics

Indian Institute of Technology Madras

April 22, 2024

# Non-linear Constrained Problem

Consider the general constrained optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} c_i(x) & = 0, \quad i \in \mathscr{E} \\ c_i(x) & \geq 0, \quad i \in \mathscr{I} \end{cases} \tag{1}$$

- $f$ is the objective function.
- $c_i : G \subset \mathbb{R}^n \to \mathbb{R}$ smooth,
- $\mathscr{I}$ and $\mathscr{E}$ are the finite index sets of inequality and equality constraints.
- We focus on fundamental concepts and building blocks that are common to more than one algorithm.

# Special cases (for which specialized algorithms exist):

- *Linear programming (LP):* $f$, all $c_i$ linear; solved by simplex & interior-point methods.
- *Quadratic programming (QP):* $f$ quadratic, all $c_i$ linear.
- *Linearly constrained optimization:* all $c_i$ linear.
- *Bound-constrained optimization:* constraints are only of the form $x_i \geq l_i$ or $x_i \leq u_i$.
- *Convex programming:* $f$ convex, equality $c_i$ linear, inequality $c_i$ concave.

# Categorization of algorithms

**Quadratic programming:**

- for solving quadratic programming problems its particular characteristics can be exploited by efficient algorithms,
- quadratic programming sub-problems need to be solved by sequential quadratic programming methods and certain interior-point methods for non-linear programming.
- Consist of active set, interior-point, and gradient projection methods.

# Categorisation of algorithms

## Penalty and augmented Lagrangian methods

- Combining the objective function and constraints into a penalty function $\phi(x;\mu)$, attack problem (1) by solving a sequence of unconstrained problems.
- $\mu$ is called the penalty parameter $\mu > 0$.
- e.g. if only equality constraints exist:

$$\phi(x;\mu) = f(x) + \frac{\mu}{2}\sum_{i\in\mathscr{E}} c_i(x)^2$$

- Minimise this unconstrained function, for a series of increasing values of $\mu$, until the solution of the constrained optimisation problem is identified to sufficient accuracy.

# Categorisation of algorithms

## Penalty methods

- If we use an exact penalty function, it may be possible to find a local solution of by solving a single unconstrained optimisation problem.
- For the equality-constrained problem, the function defined by

$$\phi(x;\mu) = f(x) + \mu \sum_{i \in \mathcal{E}} |c_i(x)|$$

  is usually an exact penalty function, for a sufficiently large value of $\mu > 0$.
- Although they often are non-differentiable.
- Exact penalty functions can be minimised by solving a sequence of smooth sub-problems.

# Categorisation of algorithms

## Augmented Lagrangian methods:

- Define a function that combines the Lagrangian and a quadratic penalty.
- Example if only equality constraints exist:

$$\mathcal{L}_A(x, \lambda; \mu) = f(x) - \sum_{i \in \mathcal{E}} \lambda_i c_i(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x)$$

- Methods based on this function fix $\lambda$ to some estimate of the optimal Lagrange multiplier vector and fix $\mu$ to some positive value, then find a value of $x$ that approximately minimises $\mathcal{L}_A(., \lambda; \mu)$.
- At this new $x$-iterate, $\lambda$ and $\mu$ may be updated; then the process is repeated.

# Categorisation of algorithms

**sequential quadratic programming (SQP) methods**:

- Model (1) by a quadratic programming subproblem at each iterate and define the search direction to be the solution of this subproblem.
- The basic SQP method, defines the search direction $p_k$ at the iterate $(x_k, \lambda_k)$ to be the solution of

$$\begin{aligned} \min_{p} \ & \frac{1}{2} p^T \nabla^2_{xx} \mathscr{L}(x_k, \lambda_k) p + \nabla f(x_k)^T p \\ \text{subject to } & \nabla c_i(x_k)^T p + c_i(x_k) = 0, \ i \in \mathscr{E}, \\ & \nabla c_i(x_k)^T p + c_i(x_k) \geq 0, \ i \in \mathscr{I}, \end{aligned}$$

- $\mathscr{L}$ is the Lagrangian function.

# Categorisation of algorithms

**sequential quadratic programming (SQP) methods**:

- The objective in this subproblem is an approximation to the change in the Lagrangian function in moving from $x_k$ to $x_k + p$.
- While the constraints are linearisations of the constraints in (1).

# ELIMINATION OF VARIABLES

- Obtain a simpler problem with fewer degrees of freedom, by using the constraints to eliminate some of the variables from the problem.
- Elimination techniques must be used with care, as they may alter the problem or introduce ill conditioning.
- **Example**

$$\min f(x) = f(x_1, x_2, x_3, x_4) \quad \text{subject to} \quad \begin{aligned} x_1 + x_3^2 - x_4 x_3 &= 0, \\ -x_2 + x_4 + x_3^2 &= 0, \end{aligned}$$

there is no risk in setting

$$x_1 = -x_3^2 + x_4 x_3, \text{ and } x_2 = x_4 + x_3^2,$$

to obtain a function of two variables

$$h(x_3, x_4) = f(x_4 x_3 - x_3^2, x_4 + x_3^2, x_3, x_4)$$

which can be minimised using the unconstrained optimisation techniques.

# Dangers of Nonlinear Elimination

- **Example** Consider the problem

$$\min \ x^2 + y^2 \qquad \text{subject to} \ \ (x-1)^3 = y^2.$$

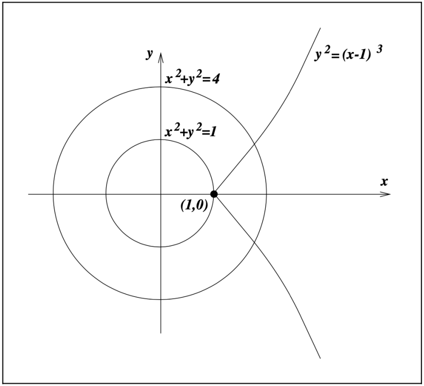


Figure: The danger of nonlinear elimination.

# Dangers of Nonlinear Elimination

- Attempt to solve this problem by eliminating $y$

$$h(x) = x^2 + (x-1)^3$$

- $h(x) \to -\infty$ as $x \to -\infty$
- By blindly applying this transformation we may conclude that the problem is unbounded.
- Ignores the fact that the constraint $(x-1)^3 = y^2$ implicitly imposes the bound $x \geq 1$.
- If we wish to eliminate $y$, we should explicitly introduce the bound $x \geq 1$ into the problem.

# SIMPLE ELIMINATION USING LINEAR CONSTRAINTS

- Non-linear function subject to a set of linear equality constraints

$$\min \; f(x) \quad \text{subject to} \quad Ax = b, \tag{2}$$

  $A$ is $m \times n$ matrix $m \leq n$.
- Let $A$ has full row rank.
- If such is not the case, either the problem is inconsistent or some of the constraints are redundant and can be deleted without affecting the solution of the problem.
- A subset of $m$ columns of $A$ can be found that is linearly independent.
- Gather these columns into an $m \times m$ matrix $B$ and define $n \times n$ permutation matrix $P$ that swaps these columns to the first $m$ column positions in $A$,

$$AP = [B|N]$$

# SIMPLE ELIMINATION USING LINEAR CONSTRAINTS

- $N$ denotes the $n - m$ remaining columns of $A$.
- Define the sub-vectors $x_B \in \mathbb{R}^m$ and $x_N \in \mathbb{R}^{N-m}$ as:

$$\begin{bmatrix} x_B \\ x_N \end{bmatrix} = P^T x,$$

- call $x_B$ the basic variables and $B$ the basis matrix.
- $PP^T = I \implies$ the constraint $Ax = b$ as

$$b = Ax = AP(P^T x) = Bx_B + Nx_N.$$

- By rearranging this formula, the basic variables can be expressed as

$$x_B = B^{-1}b - B^{-1}Nx_N$$

# SIMPLE ELIMINATION USING LINEAR CONSTRAINTS

- Compute a feasible point for the constraints $Ax = b$ by choosing any value of $x_N$ and then setting $x_B$ according to the previous formula.
- The problem is therefore equivalent to the unconstrained problem

$$\min_{x_N} h(x_N) \stackrel{\text{def}}{=} f\left(P\begin{bmatrix} B^{-1}b - B^{-1}Nx_N \\ x_N \end{bmatrix}\right) \qquad (3)$$

- The expression for $x_B$ is referred to as simple elimination of variables.
- This discussion shows that a non-linear optimisation problem with linear equality constraints is, the same as an unconstrained problem.

# EXAMPLE

- Consider the problem

$$\begin{aligned} \min\ & \sin(x_1 + x_2) + x_3^2 + \frac{1}{3}(x_4 + x_5^4 + x_6/2) \\ \text{subject to } & 8x_1 - 6x_2 + x_3 + 9x_4 + 4x_5 = 6 \\ & 3x_1 + 2x_2 - x_4 + 6x_5 + 4x_6 = -4. \end{aligned} \tag{4}$$

- Define the permutation matrix $P$ to reorder the components of $x$ as $x^T = (x_3, x_6, x_1, x_2, x_4, x_5)^T$.
- The coefficient matrix $AP$ is

$$AP = \left[\begin{array}{cc|cccc} 1 & 0 & 8 & -6 & 9 & 4 \\ 0 & 4 & 3 & 2 & -1 & 6 \end{array}\right]$$

- Basis matrix B is diagonal.

# EXAMPLE

- $$\begin{bmatrix} x_3 \\ x_6 \end{bmatrix} = -\begin{bmatrix} 8 & -6 & 9 & 4 \\ \frac{3}{4} & \frac{1}{2} & -\frac{1}{4} & \frac{3}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} 6 \\ -1 \end{bmatrix}$$

- By substituting for $x_3$ and $x_6$, the problem becomes

$$\min_{x_1,x_2,x_4,x_5} \sin(x_1 + x_2) + (8x_1 - 6x_2 + 9x_4 + 4x_5 - 6)^2 + \frac{1}{3}(x_4 + x_5^4 - [(1/2) + (3/8)x_1 + (1/4)x_2 - (1/8)x_4 + (3/4)x_5]). \tag{5}$$

# SIMPLE ELIMINATION USING LINEAR CONSTRAINTS

- Assume that the coefficient matrix is already given so that the basic columns appear in the first $m$ positions, that is, $P = I$.
- Any feasible point $x$ for the linear constraints in $Ax = b$ can be written as:

$$\begin{bmatrix} x_B \\ x_N \end{bmatrix} = x = Yb + Zx_N$$

- $$Y = \begin{bmatrix} B^{-1} \\ 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix}$$

- $Z$ has $n - m$ linearly independent columns (because of the presence of the identity matrix in the lower block) and it satisfies $AZ = 0$.
- Therefore, $Z$ is a basis for the null space of $A$.

# SIMPLE ELIMINATION USING LINEAR CONSTRAINTS

- The columns of $Y$ and the columns of $Z$ form a linearly independent set.
- $Yb$ is a particular solution of the linear constraints $Ax = b$.
- In other words, the simple elimination technique expresses feasible points as the sum of a particular solution of $Ax = b$ plus a displacement along the null space of the constraints.
- More general elimination strategies do also exist catering to different pathologies.

# EFFECT OF INEQUALITY CONSTRAINTS

- Elimination of variables is not always beneficial if inequality constraints are present alongside the equalities.
- If problem (4) had the additional constraint $x \geq 0$, then after eliminating the variables $x_3$ and $x_6$, would lead to a problem of minimising the function in subject to the constraints.

$$\begin{aligned} (x_1, x_2, x_4, x_5) &\geq 0, \\ 8x_1 - 6x_2 + 9x_4 + 4x_5 &\leq 6, \\ (3/4)x_1 + (1/2)x_2 - (1/4)x_4 + (3/2)x_5 &\leq -1. \end{aligned}$$

- Hence, the cost of eliminating the equality constraints is to make the inequalities more complicated than the simple bounds $x \geq 0$.

# Measuring Progress

- An algorithm for solving the non-linear programming problem generates a step that reduces the objective function but increases the violation of the constraints.
- Should we accept this step ?
- Question is not easy to answer.
- Look for a way to balance the twin (often competing) goals of reducing the objective function and satisfying the constraints.
- Merit functions and filters are two approaches for achieving this balance.
- A step $p$ will be accepted only if it leads to a sufficient reduction in the merit function $\phi$ or if it is acceptable to the filter.

# MERIT FUNCTIONS

- Unconstrained optimisation:the objective function $f$ is the natural choice for the merit function.
- All the unconstrained optimisation methods described require that $f$ be decreased (non-increasing) at each step.
- Methods for constrained optimisation in which the starting point and all subsequent iterates satisfy all the constraints in the problem, the objective function is still an appropriate merit function.
- But, algorithms that allow iterates to violate the constraints require some means to assess the quality of the steps and iterates.
- The merit function in this case combines the objective with measures of constraint violation.

# MERIT FUNCTIONS

## Exact Merit Function

A merit function $\phi(x;\mu)$ is exact if there is a positive scalar $\mu^*$ such that for any $\mu > \mu^*$, any local solution of the non-linear programming problem is a local minimiser of $\phi(x;\mu)$.

A popular choice of merit function for the non-linear programming problem is the $l_1$ penalty function.

## $l_1$ penalty function

$$\phi_1(x;\mu) = f(x) + \mu \sum_{i\in\mathcal{E}} |c_i(x)| + \mu \sum_{i\in\mathcal{I}} [c_i(x)]^-, \qquad (6)$$

where $[z]^- = \max\{0, -z\}$. The positive scalar $\mu$ is the penalty parameter, which determines the weight that we assign to constraint satisfaction relative to minimisation of the objective.

# MERIT FUNCTIONS

- The $l_1$ merit function $\phi_1$ is not differentiable because of the presence of the absolute value and $[.]^-$ functions, but it is exact.
- It is exact for $\mu^* =$ largest Lagrange multiplier (in absolute value) associated with an optimal solution.
- Many algorithms using this function adjust $\mu$ heuristically to ensure $\mu > \mu^*$ (but not too large).
- It is inexpensive to evaluate but it may reject steps that make good progress toward the solution (Maratos effect)

# MERIT FUNCTIONS

## Fletcher's augmented Lagrangian

when only equality constraints $c(x) = 0$ exist:

$$\phi_F(x;\mu) = f(x) - \lambda(x)^T c(x) + \frac{\mu}{2}||c(x)||_2^2 \qquad (7)$$

where $A(x)$ is the Jacobian of $c(x)$ and $\lambda(x) = (A(x)A(x)^T)^{-1}A(x)\nabla f(x)$ are the least squares multipliers' estimates.

- It is differentiable and exact and does not suffer from the Maratos effect.
- But, since it requires the solution of a linear system to obtain $\lambda(x)$, it is expensive to evaluate; and may be ill-conditioned or not defined.