Algorithms for constrained problems

- So far, we have discussed algorithms for only unconstrained problems in $\mathbb{R}^n$ (possible exception: random search).
- Usual form of algorithm:
$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$$
- What if we have constraints?
- The iterates $\boldsymbol{x}^{(k)}$ may not satisfy the constraints.
- Need to modify the algorithms to take into account constraints.

Box constraints

- Consider the problem
$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \Omega \end{array}$$
where the constraint set $\Omega \subset \mathbb{R}^n$ is given by
$$\Omega = \{\boldsymbol{x} : l_i \leq x_i \leq u_i,\ i = 1, \ldots, n\}.$$
- Ω is a "box" in $\mathbb{R}^n$.
- How do we apply our previous optimization algorithms to the above problem?
- One way: use "projection".

Projections

- Consider the box constraint set Ω.
- Given a point $\boldsymbol{x} \in \mathbb{R}^n$, define $\boldsymbol{y} = \boldsymbol{\Pi}[\boldsymbol{x}] \in \mathbb{R}^n$ by
$$y_i = \begin{cases} u_i & \text{if } x_i > u_i \\ x_i & \text{if } l_i \leq x_i \leq u_i \\ l_i & \text{if } x_i < l_i \end{cases}$$
- $\boldsymbol{\Pi}[\boldsymbol{x}]$ is the "projection" of $\boldsymbol{x}$ onto Ω.
- Note that $\boldsymbol{\Pi}[\boldsymbol{x}]$ is actually the closest point in Ω to $\boldsymbol{x}$.

Projected algorithm

- We can modify the usual algorithm

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$$

  by using projections:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{\Pi}[\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}].$$

- Note that the iterates $\boldsymbol{x}^{(k)}$ all lie inside Ω.
- Under appropriate conditions, can prove global convergence.

More general constraints

- Consider the general constrained problem

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{x} \in \Omega. \end{aligned}$$

- If Ω is sufficiently "nice", we can define the projection onto Ω:

$$\boldsymbol{\Pi}[\boldsymbol{x}] = \arg\min_{\boldsymbol{z} \in \Omega} \|\boldsymbol{z} - \boldsymbol{x}\|.$$

- We can similarly apply the projected algorithm

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{\Pi}[\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}].$$

- In general, given $\boldsymbol{x}$, the computation of $\boldsymbol{\Pi}[\boldsymbol{x}]$ may not be easy.
- In some cases, there is a formula for computing $\boldsymbol{\Pi}[\boldsymbol{x}]$.

  For example:

  - Ω = Box constraint set
  - Ω = Linear variety (plane) [see §4.2]

- In general, the projection $\boldsymbol{\Pi}[\boldsymbol{x}]$ may have to be computed numerically.
- Note that the numerical computation of $\boldsymbol{\Pi}[\boldsymbol{x}]$ itself involves an optimization algorithm.
- The computation of $\boldsymbol{\Pi}[\boldsymbol{x}]$ may be as difficult as the original optimization problem!
- Example:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{x}\|^2 \\ \text{subject to} \quad & \boldsymbol{x} \in \Omega. \end{aligned}$$

Penalty method

- Another method of dealing with constraints is to approximate the constrained problem with an unconstrained problem.
- Consider the constrained problem

$$\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x} \in \Omega. \end{array}$$

- Construct an associated unconstrained problem

$$\text{minimize } f(\boldsymbol{x}) + \gamma P(\boldsymbol{x}),$$

  where $\gamma > 0$ and P is a continuous function with the property that

$$P(\boldsymbol{x}) \begin{cases} = 0 & \text{if } \boldsymbol{x} \in \Omega \\ > 0 & \text{if } \boldsymbol{x} \notin \Omega. \end{cases}$$

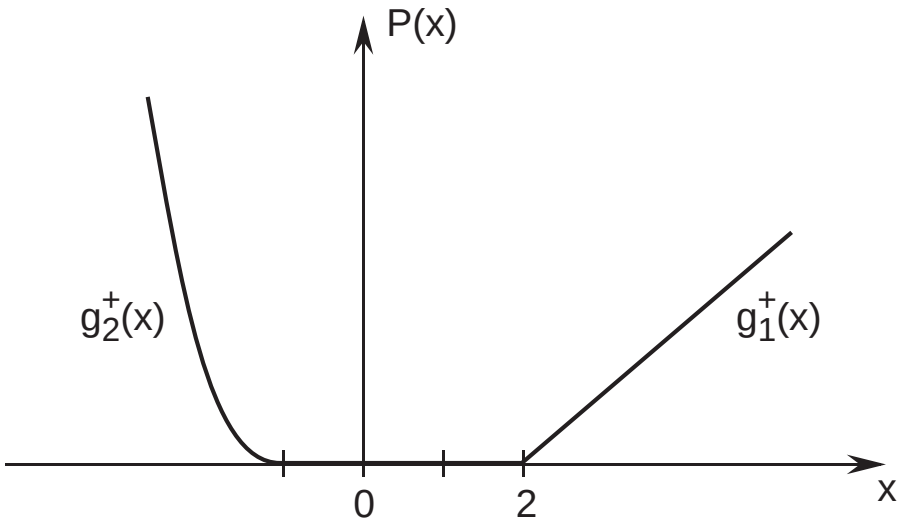
- γ: *penalty parameter*
- P: *penalty function*

- We use the solution to the unconstrained problem as an approximation to the solution of the constrained problem.
- Intuition: points $\boldsymbol{x}$ outside Ω will unlikely be minimizers because they are "penalized" by the term $\gamma P(\boldsymbol{x})$.
- We expect that the larger the value of γ, the better the approximation.

Example

- Constraint set $\Omega = [-1, 2]$.
- Penalty function:

$$P(x) = \begin{cases} x - 2 & \text{if } x > 2 \\ 0 & \text{if } -1 \leq x \leq 2 \\ -(x+1)^3 & \text{if } x < -1 \end{cases} .$$



- Note that to apply our previous optimization algorithms, we need P to be differentiable.
- Is the solution to the unconstrained problem a good approximation to the true solution?
- Depends on the penalty parameter γ and the penalty function P.
- Under certain conditions, can show that the approximation becomes exact as $\gamma \to \infty$ (§22.2).
- May have to perform a sequence of minimization runs with increasing γ values.

Exact penalty functions

- Is it possible to find a *finite* γ such that the solution to the unconstrained problem is exactly the solution to the original (constrained) problem?
- If yes, then we only need to do a single minimization run.
- In such a case, we say that the penalty function is *exact*.
- Bad news: it is necessary for an exact penalty function be nondifferentiable (Bertsekas 75).

