CIVIL-477: Notes on Gradient Projection Algorithm

March 27, 2025

1 Pseudo code of GP algorithm

Algorithm 1 Gradient projection algorithm

Inputs: Network $G(N, A)$; demand vector $\mathbf{q}$; max iteration N; gap threshold of main loop ε.
Outputs: Equilibrium path flow $\mathbf{f}^*$
Initialization:
- Set iteration $n = 0$, gap $g = \infty$
- Initialize path flow $f_k = q_w/|P_w|$, $\forall k \in P_w$, and non-base path set $P_{NB,w} = \emptyset$ for each OD pair $w \in W$

while $n \leq N$ and $g > \varepsilon$ **do**
    Step 0: Update path cost and link cost derivative
    - compute path cost $\mathbf{c}^n$ using path flow $\mathbf{f}^n$
    - compute link cost derivative $\partial \mathbf{t}^n = \mathbf{t}'(\Delta \mathbf{f}^n)$
    for Each OD pair $w \in W$ **do**
        Step 1: Construct decent direction
        - Find the shortest/base path k^* and update non-base path set $P_{NB,w} = P_w \setminus \{k^*\}$
        - Compute direction $\mathbf{d}_w^n = c_{k^*}\mathbf{1} - \mathbf{c}_w^n$
        Step 2: Compute step size
        - Compute $\alpha_w^n = \left[\text{diag}\Big((\Delta_w - \Delta_{k^*})^T \partial \mathbf{t}^n (\Delta_w - \Delta_{k^*})\Big)\right]^{-1}$
        Step 3: Update path flow
        - Compute candidate flow $\mathbf{y}_w^n = \mathbf{f}_w^n + \alpha_w^n \mathbf{d}_w^n$
        - Update non-base path flows by projection $f_{w,k}^{n+1} = \max\{0, y_{w,k}^n\}$, $\forall k \in P_{NB,w}$
        - Update flow on base path $f_{k^*}^{n+1} = q_w - \sum_{k \in P_{NB,w}} f_{w,k}^{n+1}$
    end for
    Step 4: Convergence check
    - Compute gap $g = ||\mathbf{f}^{n+1} - \mathbf{f}^n||_2$
    Set $n = n + 1$
end while
Set $\mathbf{f}^* = \mathbf{f}^{n+1}$

Notes:

- Δ is link-path incidence matrix, Δ_w is the submatrix of Δ that corresponds to path set P_w, and Δ_{k^*} is a column in Δ that corresponds to base path k^*
- $\text{diag}(A)$ denotes the diagonal elements in matrix A

2 Quasi-Newton method

In the GP algorithm, we use a quasi-Newton method to compute the step size n, which helps improve the convergence rate, i.e., reaching the optimal solution much faster. The term "quasi-Newtwon" comes from the Newton method (`https://en.wikipedia.org/wiki/Newton%27s_method`).

In brief, the Newtown method is rooted in the second-order approximation of a function

$$g(\mathbf{y}) = f(\mathbf{x}) + \nabla f^T(\mathbf{x})(\mathbf{y}-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathbf{x})^T H_{\mathbf{x}}(\mathbf{y}-\mathbf{x}), \tag{1}$$

where $H_x = \nabla^2 f(\mathbf{x})$ is the Hessian matrix evaluated at x.

The optimal solution to the minimization problem over $g(\mathbf{y})$ is thus

$$\nabla g(\mathbf{y}) = \nabla f(\mathbf{x}) + H_{\mathbf{x}}(\mathbf{y}-\mathbf{x}) = \mathbf{0} \quad \Rightarrow \quad \mathbf{y} = \mathbf{x} - H_{\mathbf{x}}^{-1}\nabla f(\mathbf{x}) \tag{2}$$

Recall our general update rule

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \alpha^n \mathbf{d}^n. \tag{3}$$

Then we have update direction $\mathbf{d}^n = -\nabla f(\mathbf{x}^n)$ and step size $\alpha^n = H_{\mathbf{x}^n}^{-1}$.

Our method is "quasi" instead of "exact" because we do not use the full Hessian matrix but only its diagonal elements $\partial^2 f(\mathbf{x})/\partial x_i^2$, while the off-diagonal elements $\partial^2 f(\mathbf{x})/\partial x_i \partial x_j$ are assumed to be zero.