

Electronic Companion

EC.1. Proof of Theorem 1

Based on Proposition 6, Lemma 3, and a well-known equivalence between a convex program and a VIP (Kinderlehrer and Stampacchia 2000), we first establish the following lemma.

LEMMA EC.1. For any $\mathbf{p}^t \in \mathcal{P}$, $\mathbf{p}^{t+1} = h(\mathbf{p}^t; \mathbf{z})$ if and only if for all $w \in \mathcal{W}$,

$$\langle r \cdot c_w(\mathbf{p}^t; \mathbf{z}) + \nabla \phi_w(\mathbf{p}_w^{t+1}) - \nabla \phi_w(\mathbf{p}_w^t), \mathbf{p}_w - \mathbf{p}_w^{t+1} \rangle \geq 0, \quad \forall \mathbf{p}_w \in Q_w(\mathbf{p}_w^t). \quad (\text{EC.1})$$

Proof. We refer the readers to Appendix EC.1.1 for the proof. ■

Based on Lemma EC.1, we then derive the following lemma, which provides conditions that guarantee a fixed point of ILD to be a WE. Plainly, it indicates that under the assumption that $c(\cdot; \mathbf{z})$ is c_z -cocoercive, a fixed point $\hat{\mathbf{p}}$ is a WE as long as one can find a WE $\mathbf{p}^*$ whose support is contained in the support of $\hat{\mathbf{p}}$.

LEMMA EC.2. Suppose that $c(\cdot; \mathbf{z})$ is c_z -cocoercive on $\mathcal{P}$. Given any $\hat{\mathbf{p}} \in \mathcal{P}$ such that $\hat{\mathbf{p}} = h(\hat{\mathbf{p}}; \mathbf{z})$, if there exists $\mathbf{p}^* \in \mathcal{P}^*(\mathbf{z})$ such that $\mathbf{p}^* \in Q(\hat{\mathbf{p}})$, then $\hat{\mathbf{p}} \in \mathcal{P}^*(\mathbf{z})$.

Proof. We refer the readers to Appendix EC.1.2 for the proof. ■

Then, to analyze the convergence of $\mathbf{p}^t$, for any $\mathbf{p}, \mathbf{p}' \in \mathcal{P}$, we define $\tilde{D}_\phi(\mathbf{p}, \mathbf{p}') = \sum_{w \in \mathcal{W}} D_{\phi_w}(\mathbf{p}_w, \mathbf{p}'_w)$. Based on Lemma EC.1 and the c_z -cocoercivity of $c(\cdot; \mathbf{z})$, we then prove the following lemma.

LEMMA EC.3. Suppose that the assumptions in Theorem 1 hold. For all $\mathbf{p}^* \in \mathcal{P}^*(\mathbf{z})$ satisfying $\mathbf{p}^* \in Q(\mathbf{p}^0)$ and all $t \geq 0$, we have

$$\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t) - \tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^{t+1}) \geq \frac{2c_z - r}{4c_z} \cdot \|\mathbf{p}^t - \mathbf{p}^{t+1}\|_2^2. \quad (\text{EC.2})$$

Proof. See Appendix EC.1.3 for the proof. ■

Per assumption, there exists WE $\mathbf{p}^* \in \mathcal{P}^*(\mathbf{z}) \cap Q(\mathbf{p}^0)$. For such $\mathbf{p}^*$, Lemma 3 implies $\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^0) < \infty$, and Lemma EC.3 implies the divergence $\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t)$ is monotonically decreasing as long as $r < 2c_z$. Thus, the limit of $\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t)$ exists according to the monotone convergence theorem. Hence, by letting $t \rightarrow \infty$ on both sides of Equation (EC.16), the squeeze theorem ensures $\|\mathbf{p}^t - \mathbf{p}^{t+1}\|_2 \rightarrow 0$.

We should note the existence of $\lim_{t \rightarrow \infty} \tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t)$ for some $\mathbf{p}^*$ does not imply the limit must be zero. Nor does it ensure the convergence of $\{\mathbf{p}_t\}$. However, since $\mathcal{P}$ is compact, the Bolzano-Weierstrass theorem guarantees the sequence $\{\mathbf{p}^t\}$ must have a convergent subsequence $\{\mathbf{p}^{t_j}\}$. Denote $\hat{\mathbf{p}}$ as the limit of $\mathbf{p}^{t_j}$ when $j \rightarrow \infty$. Assume $\|h(\hat{\mathbf{p}}; \mathbf{z}) - \hat{\mathbf{p}}\| = \delta$ for some $\delta > 0$. That is, the limit of the subsequence $\hat{\mathbf{p}}$ is not a fixed point of the ILD. We proceed to establish a contradiction. Note that

$$\begin{aligned} \|\mathbf{p}^{t_j+1} - \mathbf{p}^{t_j}\|_2 &= \|h(\mathbf{p}^{t_j}; \mathbf{z}) - \mathbf{p}^{t_j}\|_2 = \|h(\mathbf{p}^{t_j}; \mathbf{z}) - h(\hat{\mathbf{p}}; \mathbf{z}) + h(\hat{\mathbf{p}}; \mathbf{z}) - \hat{\mathbf{p}} + \hat{\mathbf{p}} - \mathbf{p}^{t_j}\|_2 \\ &\geq \|h(\hat{\mathbf{p}}; \mathbf{z}) - \hat{\mathbf{p}}\|_2 - \|h(\mathbf{p}^{t_j}; \mathbf{z}) - h(\hat{\mathbf{p}}; \mathbf{z}) + \hat{\mathbf{p}} - \mathbf{p}^{t_j}\|_2 \geq \delta - \|h(\mathbf{p}^{t_j}; \mathbf{z}) - h(\hat{\mathbf{p}}; \mathbf{z})\|_2 - \|\hat{\mathbf{p}} - \mathbf{p}^{t_j}\|_2. \end{aligned} \quad (\text{EC.3})$$

Letting $j \rightarrow \infty$ on both sides of Equation (EC.3), the left-hand side then converges to 0. Meanwhile, on the right-hand side, we also have $\|\hat{\mathbf{p}} - \mathbf{p}^{t_j}\| \rightarrow 0$ and thus $\|h(\mathbf{p}^{t_j}; \mathbf{z}) - h(\hat{\mathbf{p}}; \mathbf{z})\| \rightarrow 0$ due to the continuity of $h(\mathbf{p}; \mathbf{z})$ with respect to $\mathbf{p}$. We then have $0 = \|h(\mathbf{p}^{t_j}; \mathbf{z}) - \mathbf{p}^{t_j}\|_2 \geq \delta > 0$, a contradiction. Thus, $\hat{\mathbf{p}} = h(\hat{\mathbf{p}}; \mathbf{z})$.

We next claim $\mathbf{p}^* \in Q(\mathbf{p}^0)$ implies $\mathbf{p}^* \in Q(\hat{\mathbf{p}})$. Otherwise, $\tilde{D}_\phi(\mathbf{p}^*, \hat{\mathbf{p}})$ would be unbounded, which is impossible given $\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t)$ is monotonically decreasing. This, along with the fact that $\hat{\mathbf{p}}$ is a fixed point of ILD, allows us to invoke Lemma EC.2 to show the limit $\hat{\mathbf{p}}$ is indeed a WE.

To recapitulate, we have established (i) $\mathbf{p}^{tj} \rightarrow \hat{\mathbf{p}}$ when $j \rightarrow \infty$; (ii) $\hat{\mathbf{p}} \in \mathcal{P}^*(\mathbf{z})$; and (iii) the limit of $\tilde{D}_\phi(\hat{\mathbf{p}}, \mathbf{p}^t)$ exists. Combining all three, we conclude $\tilde{D}_\phi(\hat{\mathbf{p}}, \mathbf{p}^t) \rightarrow 0$ when $t \rightarrow \infty$, which implies $\mathbf{p}^t \rightarrow \hat{\mathbf{p}}$. ■

EC.1.1. Proof of Lemma EC.1

The following lemma (Kinderlehrer and Stampacchia 2000) characterizes the relation between a convex program and a VIP.

LEMMA EC.4. *Given a closed and convex set $\mathcal{Y} \subseteq \mathbb{R}^m$ and a continuously differentiable and convex function $f : \mathcal{Y} \rightarrow \mathbb{R}$, consider the optimization problem $\min_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{y})$. Then $\mathbf{y}^* \in \mathcal{Y}$ is an optimal solution to this convex program if and only if $\langle \nabla f(\mathbf{y}^*), \mathbf{y} - \mathbf{y}^* \rangle \geq 0$ for all $\mathbf{y} \in \mathcal{Y}$.*

Based on Lemma EC.4, we can then prove Lemma EC.1. First, we have $\mathbf{p}^{t+1} = h(\mathbf{p}^t; \mathbf{z})$ if and only if $\mathbf{p}_w^{t+1}$ solves Equation (7) per Proposition 6. As the objective function in Equation (7), denoted as $f_w(\mathbf{p}_w) = r \cdot \langle c_w(\mathbf{p}^t; \mathbf{z}), \mathbf{p}_w \rangle + D\phi_w(\mathbf{p}_w, \mathbf{p}_w^t)$, is finite if and only if $\mathbf{p}_w \in Q_w(\mathbf{p}_w^t)$, we may equivalently minimize $f_w(\mathbf{p}_w)$ over $Q_w(\mathbf{p}_w^t)$, which is a closed set. Meanwhile, we have $\nabla f_w(\mathbf{p}_w) = r \cdot c_w(\mathbf{p}^t; \mathbf{z}) + \nabla \phi_w(\mathbf{p}_w) - \nabla \phi_w(\mathbf{p}_w^t)$ and $\nabla^2 f_w(\mathbf{p}_w) = \nabla^2 \phi_w(\mathbf{p}_w)$, which is positively definite as ϕ_w is strongly convex on $\mathcal{P}_w$. Hence, $f_w(\mathbf{p}_w)$ is also strongly convex on $\mathcal{P}_w$. The proof is then concluded by directly applying Lemma EC.4. ■

EC.1.2. Proof of Lemma EC.2

By replacing both $\mathbf{p}^t$ and $\mathbf{p}^{t+1}$ by $\hat{\mathbf{p}}$ in Equation (EC.1) in Lemma EC.1, summing over $w \in \mathcal{W}$, and dividing both sides by the positive constant r , we arrive at the following: $\hat{\mathbf{p}} \in \mathcal{P}$ satisfies $\hat{\mathbf{p}} = h(\hat{\mathbf{p}}; \mathbf{z})$ if and only if

$$\langle c(\hat{\mathbf{p}}; \mathbf{z}), \mathbf{p} - \hat{\mathbf{p}} \rangle \geq 0, \quad \forall \mathbf{p} \in Q(\hat{\mathbf{p}}). \quad (\text{EC.4})$$

As $\mathbf{p}^* \in Q(\hat{\mathbf{p}})$ by assumption, we then have $\langle c(\hat{\mathbf{p}}; \mathbf{z}), \mathbf{p}^* - \hat{\mathbf{p}} \rangle \geq 0$, which subsequently implies that

$$\langle c(\hat{\mathbf{p}}; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle + \langle c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle = -\langle c(\hat{\mathbf{p}}; \mathbf{z}), \mathbf{p}^* - \hat{\mathbf{p}} \rangle \leq 0. \quad (\text{EC.5})$$

Noting that $\langle c(\hat{\mathbf{p}}; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle \geq c_z \cdot \|c(\hat{\mathbf{p}}; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z})\|_2^2$ by the c_z -cocoercivity of $c(\cdot; \mathbf{z})$, we subsequently obtain

$$c_z \cdot \|c(\hat{\mathbf{p}}; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z})\|_2^2 + \langle c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle \leq 0. \quad (\text{EC.6})$$

As $\langle c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle \geq 0$ by Proposition 1, we must have $c(\hat{\mathbf{p}}; \mathbf{z}) = c(\mathbf{p}^*; \mathbf{z})$ as well as

$$\langle c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle = 0. \quad (\text{EC.7})$$

Denoting $c_k^* = c_k(\hat{\mathbf{p}}; \mathbf{z}) = c_k(\mathbf{p}^*; \mathbf{z})$ and $b_w^* = \min_{k' \in \mathcal{K}_w} \{c_{k'}^*\}$, we then prove $c_k^* > b_w^* \Rightarrow \hat{\mathbf{p}}_k = 0$ for all $w \in \mathcal{W}$ and $k \in \mathcal{K}_w$, which shall indicate $\hat{\mathbf{p}}$ is a WE by definition. To prove this, suppose that there exist $w' \in \mathcal{W}$ and $k' \in \mathcal{K}_{w'}$ such that $c_{k'}^* > b_{w'}^*$ and $\hat{\mathbf{p}}_{k'} > 0$. Denoting $\delta = c_{k'}^* - b_{w'}^*$, we will then have

$$\langle c(\hat{\mathbf{p}}; \mathbf{z}), \hat{\mathbf{p}} \rangle \geq \sum_{w \in \mathcal{W}} b_w^* + \delta \cdot \hat{\mathbf{p}}_{k'}.$$

Noting that $\langle c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^* \rangle = \sum_{w \in \mathcal{W}} b_w^*$ and $c(\mathbf{p}^*; \mathbf{z}) = c(\hat{\mathbf{p}}; \mathbf{z})$, we then derive that $\langle c(\mathbf{p}^*; \mathbf{z}), \hat{\mathbf{p}} - \mathbf{p}^* \rangle > 0$, which contradicts Equation (EC.7). Hence, we must have $\hat{\mathbf{p}} \in \mathcal{P}^*(\mathbf{z})$, which concludes the proof. ■

EC.1.3. Proof of Lemma EC.3

We first provide several lemmas that will be used in the proof. In all of these lemmas, we assume that $\mathcal{Y}$ is a closed convex set.

LEMMA EC.5 (Chen and Teboulle (1993)). *For any Bregman divergence $D_\phi : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R} \cup \{\infty\}$ and any $\mathbf{y}, \mathbf{y}', \mathbf{y}'' \in \mathcal{Y}$, we have*

$$D_\phi(\mathbf{y}, \mathbf{y}') + D_\phi(\mathbf{y}', \mathbf{y}'') - D_\phi(\mathbf{y}, \mathbf{y}'') = \langle \mathbf{y} - \mathbf{y}', \nabla \phi(\mathbf{y}'') - \nabla \phi(\mathbf{y}') \rangle. \quad (\text{EC.8})$$

LEMMA EC.6. *If ϕ is σ -strongly convex with respect to $\|\cdot\|$ on $\mathcal{Y}$, then for all $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, the induced Bregman divergence $D_\phi(\mathbf{y}, \mathbf{y}')$ satisfies*

$$D_\phi(\mathbf{y}, \mathbf{y}') \geq \frac{\sigma}{2} \cdot \|\mathbf{y} - \mathbf{y}'\|^2. \quad (\text{EC.9})$$

LEMMA EC.7 (Beck and Teboulle (2003)). *The negative entropy function $\phi(\mathbf{y}) = \langle \mathbf{y}, \log(\mathbf{y}) \rangle$, which induces the KL divergence, is 1-strongly convex with respect to ℓ_1 norm $\|\cdot\|_1$.*

LEMMA EC.8. *For all $\mathbf{y} \in \mathcal{Y}$, we have $\|\mathbf{y}\|_2 \leq \|\mathbf{y}\|_1$.*

LEMMA EC.9 (Marcotte and Wu (1995)). *For any two vectors $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, we have*

$$\langle \mathbf{y}, \mathbf{y}' \rangle - \|\mathbf{y}'\|_2^2 \leq \frac{1}{4} \cdot \|\mathbf{y}\|_2^2. \quad (\text{EC.10})$$

Now we are ready to prove Theorem 1. For all $\mathbf{p}^* \in \mathcal{P}^*(\mathbf{z})$, Lemma EC.5 first implies that

$$\begin{aligned} D_{\phi_w}(\mathbf{p}_w^*, \mathbf{p}_w^{t+1}) &= D_{\phi_w}(\mathbf{p}_w^*, \mathbf{p}_w^t) - D_{\phi_w}(\mathbf{p}_w^{t+1}, \mathbf{p}_w^t) \\ &\quad + \langle \nabla \phi_w(\mathbf{p}_w^t) - \nabla \phi_w(\mathbf{p}_w^{t+1}), \mathbf{p}_w^* - \mathbf{p}_w^{t+1} \rangle. \end{aligned} \quad (\text{EC.11})$$

Combining Equation (EC.1) in Lemma EC.1 and Equation (EC.11), we arrive at

$$D_{\phi_w}(\mathbf{p}_w^*, \mathbf{p}_w^t) - D_{\phi_w}(\mathbf{p}_w^*, \mathbf{p}_w^{t+1}) \geq D_{\phi_w}(\mathbf{p}_w^{t+1}, \mathbf{p}_w^t) + r \cdot \langle c_w(\mathbf{p}^t; \mathbf{z}), \mathbf{p}_w^{t+1} - \mathbf{p}_w^* \rangle. \quad (\text{EC.12})$$

Summarizing Equation (EC.12) for all $w \in \mathcal{W}$ leads to

$$\begin{aligned} \tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t) - \tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^{t+1}) &\geq \tilde{D}_\phi(\mathbf{p}^{t+1}, \mathbf{p}^t) + r \cdot \langle c(\mathbf{p}^t; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^* \rangle \\ &\geq \tilde{D}_\phi(\mathbf{p}^{t+1}, \mathbf{p}^t) + r \cdot \langle c(\mathbf{p}^t; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^* \rangle, \end{aligned} \quad (\text{EC.13})$$

where the second inequality holds because $\langle c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^* \rangle \geq 0$. Below, we further bound the two terms on the right-hand side of Equation (EC.13). First,

$$\tilde{D}_\phi(\mathbf{p}^{t+1}, \mathbf{p}^t) \geq \frac{1}{2} \cdot \sum_{w \in \mathcal{W}} \|\mathbf{p}_w^t - \mathbf{p}_w^{t+1}\|_1^2 \geq \frac{1}{2} \cdot \sum_{w \in \mathcal{W}} \|\mathbf{p}_w^t - \mathbf{p}_w^{t+1}\|_2^2 = \frac{1}{2} \cdot \|\mathbf{p}^t - \mathbf{p}^{t+1}\|_2^2, \quad (\text{EC.14})$$

where the first inequality follows from Lemmas EC.6 and EC.7, and the second follows from Lemma EC.8. Second,

$$\begin{aligned} \langle c(\mathbf{p}^t; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^* \rangle &= \langle c(\mathbf{p}^t; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^t + \mathbf{p}^t - \mathbf{p}^* \rangle \\ &\geq \langle c(\mathbf{p}^t; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z}), \mathbf{p}^{t+1} - \mathbf{p}^t \rangle + c_z \cdot \|c(\mathbf{p}^t; \mathbf{z}) - c(\mathbf{p}^*; \mathbf{z})\|_2^2 \\ &\geq -\frac{1}{4c_z} \cdot \|\mathbf{p}^t - \mathbf{p}^{t+1}\|_2^2, \end{aligned} \quad (\text{EC.15})$$

where the first inequality is guaranteed by the cocoercivity of $c(\mathbf{p}; \mathbf{z})$ and the second follows from Lemma EC.9. Plugging Equations (EC.14) and (EC.15) into Equation (EC.13) results in

$$\tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^t) - \tilde{D}_\phi(\mathbf{p}^*, \mathbf{p}^{t+1}) \geq \frac{2c_z - r}{4c_z} \cdot \|\mathbf{p}^t - \mathbf{p}^{t+1}\|_2^2, \quad (\text{EC.16})$$

which concludes the proof. ■

EC.2. Algorithms Included in the Numerical Experiments

We first provide additional implementation details of the proposed algorithms (Appendix EC.2.1) before proceeding to introduce other general SCG algorithms included for comparison (Appendix EC.2.2). Application-specific heuristics are presented in Appendix EC.2.3.

EC.2.1. DoIMD and SiIMD with route generation

Algorithm EC.1 details a version of DoIMD in which routes are not enumerated beforehand but generated iteratively on the fly. The initial $\mathbf{p}^0$ is set according to the initial $\mathcal{K}^+$: travelers are split equally to all routes currently present between each OD pair. At the end of each outer iteration, the shortest route between each OD pair is identified and included in $\mathcal{K}^+$ if it is new. Whenever $\mathcal{K}^+$ is updated, $\mathbf{p}^0$ is reset accordingly.

Algorithm EC.1 DoIMD implemented with dynamic route generation.

- 1: **Input:** $\mathcal{K}^+ \subseteq \mathcal{K}$, $\mathbf{z}^0 \in \mathcal{Z}$, step sizes $\rho > 0$ and $r > 0$, threshold values $\varepsilon > 0$ and $\xi > 0$.
 - 2: Set $\mathbf{p}^0 = (p_k^0)_{k \in \mathcal{K}}$, where $p_k^0 = 1/|\mathcal{K}_w^+|$ if $k \in \mathcal{K}_w^+$ and $p_k^0 = 0$ if $k \in \mathcal{K}_w \setminus \mathcal{K}_w^+$ for all $w \in \mathcal{W}$.
 - 3: **for** $i = 0, 1, \dots$ **do**
 - 4: **FP** (running Algorithm 1 until convergence):
 - 5: **for** $t = 0, 1, \dots$ **do**
 - 6: Run $\mathbf{p}^{t+1} = h(\mathbf{p}^t; \mathbf{z}^i)$.
 - 7: If $\delta(\mathbf{p}^{t+1}; \mathbf{z}^i) \leq \varepsilon$, break and set $T = t + 1$ and $\mathbf{l}^T = \mathbf{l}(\bar{\mathbf{A}}\mathbf{p}^T; \mathbf{z}^i)$.
 - 8: **end for**
 - 9: **BP:** Calculate $\mathbf{l}_z = \partial \mathbf{l}^T / \partial \mathbf{z}^i$ by unrolling FP via AD (AD tools automatically programmed and executed Algorithm 2).
 - 10: Set $\mathbf{z}^{i+1} = \arg \min_{\mathbf{z} \in \mathcal{Z}} \rho \cdot \langle \mathbf{l}_z, \mathbf{z} - \mathbf{z}^i \rangle + D_\psi(\mathbf{z}, \mathbf{z}^i)$.
 - 11: Set newroute = false.
 - 12: **for** $w \in \mathcal{W}$ **do**
 - 13: Find $k^* = \arg \min_{k \in \mathcal{K}_w} c_k(\mathbf{p}^T; \mathbf{z}^{i+1})$ through a shortest route algorithm.
 - 14: If $k^* \notin \mathcal{K}_w^+$, set $\mathcal{K}^+ = \mathcal{K}^+ \cup \{k^*\}$ and newroute = true.
 - 15: **end for**
 - 16: If newroute = true, reset $\mathbf{p}^0 = (p_k^0)_{k \in \mathcal{K}}$, where $p_k^0 = 1/|\mathcal{K}_w^+|$ if $k \in \mathcal{K}_w^+$ and $p_k^0 = 0$ if $k \in \mathcal{K}_w \setminus \mathcal{K}_w^+$ for all $w \in \mathcal{W}$.
 - 17: If $\|\mathbf{z}^{i+1} - \mathbf{z}^i\|_\infty < \xi$ and newroute = false, break and set $\mathbf{z}^* = \mathbf{z}^{i+1}$.
 - 18: **end for**
-

In Algorithm EC.1, route generation is performed only between outer iterations. This design is adopted for several reasons. First, adding routes inside the inner ILD loop would require repeated shortest-route searches and updates of the route-link and route-OD incidence data structure, which could significantly increase the running time. Second, it would introduce a discrete nonsmooth operation into the trajectory. Under the current design, during each FP/BP calculation, the generated route set $\mathcal{K}^+$ is fixed, and AD is applied only to the smooth finite ILD trajectory defined on this fixed route set. Since the shortest-route routine used to expand $\mathcal{K}^+$ is not part of the computational graph, the unrolled computation is smooth.

For very large instances, such as Chicago Sketch, the memory needed to store the full FP graph may still be substantial. In such cases, DoIMD can be implemented with truncated AD (Shaban et al. 2019). Specifically, the FP

iterations can first be run without recording the computational graph until the equilibrium gap falls below a loose threshold $\varepsilon_{\text{tr}} > \varepsilon$. The graph is then recorded only for the remaining iterations needed to reach the final threshold ε . This implementation reduces memory usage, at the cost of treating the earlier FP iterations as detached and hence introducing an additional truncation approximation in the computed gradient.

Algorithm EC.2 describes how SilMD is implemented with route generation. The main difference concerns how the route choice strategy is initialized when a new route is found. In SilMD, if a new route k^* is added to the route set, its choice probability should be manually set. Otherwise, the flow on that route will always remain zero since, unlike DolMD, route choice strategies are not reset every time a route expansion occurs (see Algorithm EC.1).

Algorithm EC.2 SilMD implemented with dynamic route generation

- 1: **Input:** $\mathcal{K}^+ \subseteq \mathcal{K}$, $\mathbf{z}^0 \in \mathcal{Z}$, step sizes $\rho > 0$ and $r > 0$, threshold values $\varepsilon > 0$ and $\xi > 0$.
 - 2: Set $\mathbf{p}^0 = (p_k^0)_{k \in \mathcal{K}}$, where $p_k^0 = 1/|\mathcal{K}_w^+|$ if $k \in \mathcal{K}_w^+$ and $p_k^0 = 0$ if $k \in \mathcal{K}_w \setminus \mathcal{K}_w^+$ for all $w \in \mathcal{W}$.
 - 3: **for** $i = 0, 1, \dots$ **do**
 - 4: **FP:** Calculate $l^T = g^{(T)}(\mathbf{p}^i; \mathbf{z}^i)$.
 - 5: **BP:** Calculate $l_z = \partial l^T / \partial \mathbf{z}^i$ by unrolling FP via AD.
 - 6: Set $\mathbf{z}^{i+1} = \arg \min_{\mathbf{z} \in \mathcal{Z}} \rho \cdot \langle l_z, \mathbf{z} - \mathbf{z}^i \rangle + D_\psi(\mathbf{z}, \mathbf{z}^i)$ and $\mathbf{p}^{i+1} = h^{(T)}(\mathbf{p}^i; \mathbf{z}^i)$.
 - 7: Set newroute = false.
 - 8: **for** $w \in \mathcal{W}$ **do**
 - 9: Find $k^* = \arg \min_{k \in \mathcal{K}_w} c_k(\mathbf{p}^{i+1}; \mathbf{z}^{i+1})$ through a shortest route algorithm.
 - 10: If $k^* \notin \mathcal{K}_w^+$, set $\mathcal{K}^+ = \mathcal{K}^+ \cup \{k^*\}$, initialize $p_{k^*}^{i+1}$ with a small positive value, and set newroute = true.
 - 11: **end for**
 - 12: Set $\mathbf{p}^{i+1} = \mathbf{p}^{i+1} / \Sigma^\top \Sigma \mathbf{p}^{i+1}$ (scaling $\mathbf{p}^{i+1}$ so that it satisfies the constraint $\Sigma \mathbf{p}^{i+1} = \mathbf{1}$).
 - 13: If $\delta(\mathbf{p}^{i+1}; \mathbf{z}^{i+1}) \leq \varepsilon$, $\|\mathbf{z}^{i+1} - \mathbf{z}^i\|_\infty < \xi$, and newroute = false, break and set $\bar{\mathbf{z}} = \mathbf{z}^{i+1}$.
 - 14: **end for**
-

For both DolMD and SilMD, in our implementation, the parameter r is fixed as a small constant, while ρ is decreased at a harmonic rate. Meanwhile, noting that $\mathcal{Z}$ is a box constraint in both CNDPs and SCTPs, we choose D_ψ as the squared Euclidean distance when implementing both algorithms.

EC.2.2. Other general SCG algorithms

EC.2.2.1. SAB algorithm Algorithm EC.3 presents an SAB algorithm for solving SCGs, which shares a similar overall structure with DolMD. At each iteration, both algorithms also perform the following three tasks: (i) find the lower-level WE; (ii) evaluate the gradient of the leader's cost at WE; and (iii) improve the leader's decision based on the gradient. For Task (i), Algorithm EC.3 enjoys the benefit of a faster WE solver, namely iGP, rather than sticking to ILD as in DolMD. For Task 2, it relies on Yang and Bell (2007)'s sensitivity analysis approach given in Proposition 5. As for Task (iii), Algorithm EC.3 also leverages the MD method, identical to DolMD.

As highlighted in Proposition 5, the algorithm requires the selected WE strategy $\bar{\mathbf{p}}^*$ satisfy the strict complementary condition, which enforces $\text{supp}(\bar{\mathbf{p}}^*) = \cup_{\mathbf{p}^* \in \mathcal{P}^*} \text{supp}(\mathbf{p}^*)$ (“no-route-left-behind”). Our experience suggests that the algorithm tends to work fine when we simply select $\bar{\mathbf{p}}^*$ as a non-degenerate extreme point of $\mathcal{X}^*$, as suggested in Tobin and Friesz (1988). Note that this approach would avoid the cost of finding the MSIC for $[\mathbf{A}_+^\top, \Sigma_+^\top]^\top$, though it may become infeasible when all extreme points of $\mathcal{X}^*$ are degenerate (Josefsson and Patriksson 2003).

Algorithm EC.3 SAB algorithm for solving Problem (2).

- 1: **Input:** $\mathbf{z}^0 \in \mathcal{Z}$, step size $\rho > 0$, threshold values $\varepsilon > 0$ and $\xi > 0$.
 - 2: **for** $i = 0, 1, \dots$ **do**
 - 3: Solve a $\mathbf{p}^i \in \mathcal{P}^*(\mathbf{z}^i)$ via iGP, terminating which when $\delta(\mathbf{p}^i; \mathbf{z}^i) < \varepsilon$. Set $\mathbf{x}^i = \bar{\mathbf{L}}\mathbf{p}^i$.
 - 4: Calculate $\nabla x^*(\mathbf{z}^i)$ according to the sensitivity analysis method detailed in Proposition 5.
 - 5: Set $\mathbf{z}^{i+1} = \arg \min_{\mathbf{z} \in \mathcal{Z}} \rho \cdot \langle l_{\mathbf{z}}, \mathbf{z} - \mathbf{z}^i \rangle + D_{\psi}(\mathbf{z}, \mathbf{z}^i)$, where $l_{\mathbf{z}} = \nabla_{\mathbf{z}} l(\mathbf{x}^i; \mathbf{z}^i) + \nabla x^*(\mathbf{z}^i) \cdot \nabla_{\mathbf{x}} l(\mathbf{x}^i; \mathbf{z}^i)$.
 - 6: If $\|\mathbf{z}^{i+1} - \mathbf{z}^i\|_{\infty} < \xi$, break and set $\mathbf{z}^* = \mathbf{z}^{i+1}$.
 - 7: **end for**
-

EC.2.2.2. Cutting plane algorithm Underlying the cutting plane (CP) algorithm is the following reformulation of the original problem (2),

$$\begin{aligned} \min_{\mathbf{z} \in \mathcal{Z}, \mathbf{p}^* \in \mathcal{P}} \quad & l(\mathbf{x}^*; \mathbf{z}), \\ \text{s.t.} \quad & \mathbf{x}^* = \bar{\mathbf{L}}\mathbf{p}^*, \quad \langle u(\mathbf{x}^*; \mathbf{z}), \mathbf{y} - \mathbf{x}^* \rangle \geq 0, \quad \forall \mathbf{y} \in \mathcal{X}, \end{aligned} \tag{EC.17}$$

where the lower-level WE is replaced by its link-based VIP formulation. Note that Problem (EC.17) has an infinite number of constraints. Lawphongpanich and Hearn (2004) noted that the VIP in Problem (EC.17) holds as long as

$$\langle u(\mathbf{x}^*; \mathbf{z}), \mathbf{y}^i - \mathbf{x}^* \rangle \geq 0, \quad \forall \text{ vertex } \mathbf{y}^i \text{ of } \mathcal{X}. \tag{EC.18}$$

This equivalence allows Problem (EC.17) to be further transformed into a standard nonlinear program with a finite number of nonlinear constraints. To avoid enumerating all vertices of $\mathcal{X}$, they further devised a cutting plane scheme that iteratively generates cuts (vertex $\mathbf{y}^i$ of $\mathcal{X}$). As described in Algorithm EC.4, each sub-problem is equivalent to a shortest route problem: its solution $\mathbf{y}^{i+1}$ corresponds to the all-or-nothing assignment where link costs equal $u(\mathbf{x}^i; \mathbf{z}^i)$.

Algorithm EC.4 Cutting plane algorithm for solving Problem (2).

- 1: **Input:** a vertex $\mathbf{y}^0$ of $\mathcal{X}$.
 - 2: **for** $i = 0, 1, \dots$ **do**
 - 3: **The master problem:** let $(\mathbf{z}^i, \mathbf{p}^i)$ be the solution to

$$\begin{aligned} \min_{\mathbf{z} \in \mathcal{Z}, \mathbf{p}^* \in \mathcal{P}} \quad & l(\mathbf{x}^*; \mathbf{z}), \\ \text{s.t.} \quad & \mathbf{x}^* = \bar{\mathbf{L}}\mathbf{p}^*, \quad \langle u(\mathbf{x}^*; \mathbf{z}), \mathbf{y}^j - \mathbf{x}^* \rangle \geq 0, \quad j = 0, \dots, i, \end{aligned}$$
 - 4: **The sub-problem:** solve $\mathbf{y}^{i+1} = \arg \min_{\mathbf{y} \in \mathcal{X}} \langle u(\mathbf{x}^i; \mathbf{z}^i), \mathbf{y} \rangle$, where $\mathbf{x}^i = \bar{\mathbf{L}}\mathbf{p}^i$.
 - 5: If $l^*(\mathbf{z}^i) - l(\mathbf{x}^i; \mathbf{z}^i)$ is sufficiently small, break and set $\mathbf{z}^* = \mathbf{z}^i$.
 - 6: **end for**
-

When dealing with a toy problem like Braess, the CP algorithm can be implemented without the need to dynamically generate cuts since the vertices of $\mathcal{X}$ can be easily enumerated. However, applying Algorithm EC.4 directly to larger networks presents significant challenges for two main reasons. First, the master problem is a highly nonconvex and nonlinear program, making it difficult to solve globally and exactly. Second, the constraint $\mathbf{p}^* \in \mathcal{P} = \{\mathbf{p} \in \mathbb{R}_+^{|\mathcal{K}|} : \mathbf{\Sigma}\mathbf{p} = \mathbf{1}\}$ in the master problem requires route enumeration to create the OD-route incidence matrix $\mathbf{\Sigma}$.

EC.2.3. Application-specific heuristics

EC.2.3.1. Heuristics for CNDPs

IOA. As described in Algorithm EC.5, our implementation solves the corresponding WE via the iGP algorithm. The optimization step minimizes $l(\mathbf{x}^i; \mathbf{z})$ over $\mathbf{z} \in \mathcal{Z}$, a nonlinear program with simple box constraints that can be handled by standard nonlinear program solvers.

Algorithm EC.5 IOA algorithm for solving CNDPs.

- 1: **Input:** $\mathbf{z}^0 \in \mathcal{Z}$, threshold values $\varepsilon > 0$ and $\xi > 0$.
 - 2: **for** $i = 1, 2, \dots$ **do**
 - 3: **The assignment step:** solve a $\mathbf{p}^i \in \mathcal{P}^*(\mathbf{z}^{i-1})$ via iGP, terminating which when $\delta(\mathbf{p}^i; \mathbf{z}^{i-1}) < \varepsilon$; set $\mathbf{x}^i = \bar{\Lambda} \mathbf{p}^i$.
 - 4: **The optimization step:** solve $\mathbf{z}^i \in \arg \min_{\mathbf{z} \in \mathcal{Z}} l(\mathbf{x}^i; \mathbf{z})$.
 - 5: If $\|\mathbf{z}^i - \mathbf{z}^{i-1}\|_\infty < \xi$, break and set $\tilde{\mathbf{z}} = \mathbf{z}^i$.
 - 6: **end for**
-

SO. The SO algorithm accepts $\tilde{\mathbf{z}}$ from the following optimization problem as an approximate solution

$$(\hat{\mathbf{z}}, \hat{\mathbf{p}}) \in \arg \min_{\mathbf{z} \in \mathcal{Z}, \mathbf{p} \in \mathcal{P}} l(\mathbf{x}; \mathbf{z}), \quad \text{s.t. } \mathbf{x} = \bar{\Lambda} \mathbf{p}. \quad (\text{EC.19})$$

On large networks, we find solving (EC.19) with commercial convex program solvers is not efficient. This is because specifying the constraint $\mathbf{p}^* \in \mathcal{P} = \{\mathbf{p} \in \mathbb{R}_+^{|\mathcal{K}|} : \Sigma \mathbf{p} = \mathbf{1}\}$ requires route enumeration. To improve efficiency, we employed a coordinate descent algorithm (see Algorithm EC.6), which iterates between two steps.

- **The assignment step:** given $\mathbf{z}^{i-1} \in \mathcal{Z}$, find $\mathbf{p}^i \in \mathcal{P}$ that minimizes $l(\mathbf{x}; \mathbf{z}^{i-1})$, where $\mathbf{x} = \bar{\Lambda} \mathbf{p}^i$. To see how this sub-problem can be transformed into a standard traffic assignment problem, we note that $l(\mathbf{x}; \mathbf{z}^{i-1}) = \langle \mathbf{x}, u(\mathbf{x}; \mathbf{z}^{i-1}) \rangle + \beta \cdot m(\mathbf{z}^{i-1})$, which indicates that the sub-problem is equivalent to an SO traffic assignment problem (SO-TAP) that aims to find $\mathbf{p} \in \mathcal{P}$ to minimize the total travel time $\langle \mathbf{x}, u(\mathbf{x}; \mathbf{z}^{i-1}) \rangle$. The resulting SO-TAP may be further transformed into an equivalent WE traffic assignment problem by adding marginal costs to the link cost functions (Sheffi 1985), which can be readily and efficiently solved via the iGP algorithm. Let us denote the solution set to this WE-TAP as $\mathcal{P}_{\text{so}}^*(\mathbf{z}^{i-1})$.

- **The optimization step:** given $\mathbf{x}^i = \bar{\Lambda} \mathbf{p}^i$, find $\mathbf{z}^i \in \mathcal{Z}$ that minimizes $l(\mathbf{x}^i; \mathbf{z})$, identical to the optimization step in the IOA algorithm, which can be solved via off-the-shelf solvers.

Algorithm EC.6 SO algorithm for solving CNDPs.

- 1: **Input:** $\mathbf{z}^0 \in \mathcal{Z}$, threshold values $\varepsilon > 0$ and $\xi > 0$.
 - 2: **for** $i = 1, 2, \dots$ **do**
 - 3: **The assignment step:** solve a $\mathbf{p}^i \in \mathcal{P}_{\text{so}}^*(\mathbf{z}^{i-1})$ via iGP, terminating which when $\delta(\mathbf{p}^i; \mathbf{z}^{i-1}) < \varepsilon$; set $\mathbf{x}^i = \bar{\Lambda} \mathbf{p}^i$.
 - 4: **The optimization step:** solve $\mathbf{z}^i \in \arg \min_{\mathbf{z} \in \mathcal{Z}} l(\mathbf{x}^i; \mathbf{z})$.
 - 5: If $\|\mathbf{z}^i - \mathbf{z}^{i-1}\|_\infty < \xi$, break and set $\tilde{\mathbf{z}} = \mathbf{z}^i$.
 - 6: **end for**
-

EC.2.3.2. Heuristics for SCTPs For SCTPs, we focus on three heuristics proposed by Harks et al. (2015), i.e., MCT, EMCDT, and CT. As mentioned in Section 2.2, all three algorithms require first obtaining the system-optimal (SO) link flow pattern $\mathbf{x}_{\text{so}} = \arg \min_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \mathbf{u}_{\text{time}}(\mathbf{x}) \rangle$.

rMCT. Algorithm EC.7 presents a modified version of MCT, referred to as rMCT in the paper. The only difference between MCT and rMCT lies in Line 4: in MCT, if the condition $x_a^i \leq x_{\text{so},a}$ is met, it sets $z_a^{i+1} = \max\{0, z_a^i - |z_a^i - z_a^{i-1}|\}$, i.e., z_a^i is decreased by $|z_a^i - z_a^{i-1}|$, the absolute difference between the objective values in consecutive iterations. The problem is that in the first iteration ($i = 1$), this value does not exist since there is no iteration 0. In our implementation, instead of reducing z_a^i by the difference, we simply reduce it by a constant that diminishes gradually. We found that rMCT converges smoothly and delivers better solutions than the original version.

Algorithm EC.7 The revised MCT (rMCT) algorithm for solving SCTPs.

- 1: **Input:** $\delta > 0$, threshold values $\varepsilon > 0$ and $\xi > 0$, $\mathbf{z}^1 = (z_a^1)_{a \in \mathcal{A}}$ such that $z_a^1 = u'_{\text{time},a}(x_{\text{so},a}) \cdot x_{\text{so},a}$ for all $a \in \tilde{\mathcal{A}}$ and $z_a^1 = 0$ for all $a \in \mathcal{A} \setminus \tilde{\mathcal{A}}$.
- 2: **for** $i = 1, 2, \dots$ **do**
- 3: Solve a $\mathbf{p}^i \in \mathcal{P}^*(\mathbf{z}^i)$ via iGP, terminating which when $\delta(\mathbf{p}^i; \mathbf{z}^i) < \varepsilon$. Set $\mathbf{x}^i = \bar{\Lambda} \mathbf{p}^i$.
- 4: For all $a \in \tilde{\mathcal{A}}$, set

$$z_a^{i+1} = \begin{cases} z_a^i + 0.9^{i-1} \cdot u'_{\text{time},a}(x_a^i) \cdot x_a^i, & \text{if } x_a^i > x_{\text{so},a} \\ \max\{0, z_a^i - 0.9^{i-1} \cdot \delta\}, & \text{if } x_a^i \leq x_{\text{so},a}. \end{cases}$$

- 5: If $\|\mathbf{z}^{i+1} - \mathbf{z}^i\|_\infty < \xi$, break and set $\tilde{\mathbf{z}} = \mathbf{z}^{i+1}$.
 - 6: **end for**
-

EMCDT and CT. The implementation of EMCDT and CT is given in Algorithms EC.8 and EC.9, respectively. The reader is referred to Harks et al. (2015) for the rationale behind these heuristics.

Algorithm EC.8 The EMCDT algorithm for solving SCTPs.

- 1: **Input:** $\delta > 0$, threshold values $\varepsilon > 0$ and $\xi > 0$, $\mathbf{z}^1 = (z_a^1)_{a \in \mathcal{A}}$ such that $z_a^1 = \max\{\delta, u'_{\text{time},a}(x_{\text{so},a}) \cdot x_{\text{so},a}\}$ for all $a \in \tilde{\mathcal{A}}$ and $z_a^1 = 0$ for all $a \in \mathcal{A} \setminus \tilde{\mathcal{A}}$.
 - 2: **for** $i = 1, 2, \dots$ **do**
 - 3: Solve a $\mathbf{p}^i \in \mathcal{P}^*(\mathbf{z}^i)$ via iGP, terminating which when $\delta(\mathbf{p}^i; \mathbf{z}^i) < \varepsilon$. Set $\mathbf{x}^i = \bar{\Lambda} \mathbf{p}^i$.
 - 4: Set $\alpha = \max_{a \in \tilde{\mathcal{A}}} u'_{\text{time},a}(x_a^i) \cdot x_a^i$.
 - 5: For all $a \in \tilde{\mathcal{A}}$, set $z_a^{i+1} = z_a^i \cdot \exp\left(\frac{0.9^{i-1}}{\max\{1, \alpha\}} \cdot (u'_{\text{time},a}(x_a^i) \cdot x_a^i - u'_{\text{time},a}(x_{\text{so},a}) \cdot x_{\text{so},a})\right)$.
 - 6: If $\|\mathbf{z}^{i+1} - \mathbf{z}^i\|_\infty < \xi$, break and set $\tilde{\mathbf{z}} = \mathbf{z}^{i+1}$.
 - 7: **end for**
-

Algorithm EC.9 The CT algorithm for solving SCTPs.

-
- 1: **Input:** toll increment $\Delta > 0$, threshold value $\varepsilon > 0$.
 - 2: Set $\tilde{z} = \mathbf{0}$ and $\mathcal{A}_{\text{act}} = \tilde{\mathcal{A}}$.
 - 3: **while** $\mathcal{A}_{\text{act}} \neq \emptyset$ **do**
 - 4: Solve a $p \in \mathcal{P}^*(\tilde{z})$ via iGP, terminating when $\delta(p; \tilde{z}) < \varepsilon$. Set $x = \bar{\Lambda}p$.
 - 5: Find $a' = \arg \max_{a \in \mathcal{A}_{\text{act}}} u'_{\text{time},a}(x_a) \cdot x_a$.
 - 6: If $x_{a'} > x_{\text{so},a'}$, set $\tilde{z}_{a'} = \tilde{z}_{a'} + \Delta$; otherwise, remove a' from $\mathcal{A}_{\text{act}}$.
 - 7: **end while**
-

References

- Beck A, Teboulle M, 2003 *Mirror descent and nonlinear projected subgradient methods for convex optimization*. *Operations Research Letters* 31(3):167–175.
- Chen G, Teboulle M, 1993 *Convergence analysis of a proximal-like minimization algorithm using Bregman functions*. *SIAM Journal on Optimization* 3(3):538–543.
- Harks T, Kleinert I, Klimm M, Möhring RH, 2015 *Computing network tolls with support constraints*. *Networks* 65(3):262–285.
- Josefsson M, Patriksson M, 2003 *Pitfalls of the sensitivity analysis of Tobin and Friesz*. Technical report, Chalmers University of Technology.
- Kinderlehrer D, Stampacchia G, 2000 *An introduction to variational inequalities and their applications* (SIAM).
- Lawphongpanich S, Hearn DW, 2004 *An MPEC approach to second-best toll pricing*. *Mathematical Programming* 101(1):33–55.
- Marcotte P, Wu JH, 1995 *On the convergence of projection methods: Application to the decomposition of affine variational inequalities*. *Journal of Optimization Theory and Applications* 85(2):347–362.
- Shaban A, Cheng CA, Hatch N, Boots B, 2019 *Truncated back-propagation for bilevel optimization*. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 1723–1732 (PMLR).
- Sheffi Y, 1985 *Urban transportation networks* (Prentice-Hall).
- Tobin RL, Friesz TL, 1988 *Sensitivity analysis for equilibrium network flow*. *Transportation Science* 22(4):242–250.
- Yang H, Bell MGH, 2007 *Sensitivity analysis of network traffic equilibrium revisited: The corrected approach*. *Proceedings of the 4th IMA International Conference on Mathematics in Transport*.