Online Algorithm Summaries

CZL–ORB (Candidates-Based)

- Precompute global utility-to-risk bounds $\rho_{\min}, \rho_{\max}$ across one or more candidates CSVs via `czl_thresholds`.
- Load a candidates problem from a `*_candidates.csv`, obtaining epochs and total budget Δ_0 (or override).
- For each epoch t with remaining budget Δ_t:
  - Compute $z = 1 - \Delta_t/\Delta_0$ and the threshold $\Psi_{\text{czl}}(t) = ((\rho_{\max} e)/\rho_{\min})^z(\rho_{\min}/e)$ (implemented in `czl_psi`).
  - Feasible set: candidates with $R(\tau) \le \Delta_t$ and $\rho(\tau) \ge \Psi_{\text{czl}}(t)$; if empty, return `None` for this epoch.
  - Otherwise pick the feasible candidate with highest ρ = utility/risk, deduct its risk from Δ_t, and proceed.
- Write selected rows (skipping `None`) to `*_online_CZL-ORB.csv`

  under `results/data/online solutions/candidates`.

BAT–ORB (Candidates-Based)

- Precompute global $\delta_{\min}$ (minimum positive risk) across one or more candidates CSVs via `bat_threshold`.
- Load a candidates problem from a `*_candidates.csv`, obtaining epochs and total budget Δ_0 (or override).
- For each epoch t with remaining budget Δ_t:
  - Compute $\Psi_{\text{bat}}(t) = (\Delta_0/\Delta_t) \cdot \ln\big(1 + \Delta_0/\delta_{\min}\big)$.
  - Feasible set: candidates with $R(\tau) \le \Delta_t$ and $\rho(\tau) \ge \Psi_{\text{bat}}(t)$; if empty, return `None` for this epoch.
  - Otherwise pick the feasible candidate with highest ρ, deduct its risk from Δ_t, and proceed.
- Write selected rows (skipping `None`) to `*_online_BAT-ORB.csv`

  under `results/data/online solutions/candidates`.

ITM–ORB (Graph-Based, Skeleton)

- Inputs: edge metrics CSV (`*_edge_values.csv`), decision timeline CSV, optional graph pickle, total budget Δ_0, and an optional utility/risk threshold ψ.
- Load the graph problem (edges DF, timeline DF, optional NetworkX graph) and ordered decision nodes from the timeline.
- Iterate epochs t: start v_s = decision_nodes[t], goal v_g = decision_nodes[t+1], with remaining budget Δ_t.
  - Solve MILP with binary x_{ij}^{ν} and continuous $\delta_t \in [0, \Delta_t]$:
    * Flow conservation (7a) from v_s to v_g.
    * Risk constraint $\sum x\, r \le \delta_t$ (7b).
    * Utility threshold $\sum x\, u \ge \psi\, \delta_t$ if $\psi > 0$ (7c).
    * One speed per edge: $\sum_{\nu} x_{ij}^{\nu} \le 1$ (7d).
    * Maximize $\sum x\, u$.

- Collect selected edge-speed tuples, δ_t, objective; deduct used risk from Δ_t and continue until budget or timeline ends.

- Outputs: per-epoch dictionaries (epoch/start/goal, selected edges, δ_t, objective, remaining budget). Extend as needed for full path constraints.