Solution Approach Summary

Overview

This problem was identified as a real-world variant of the Traveling Salesman Problem. Because multiple agents are required to serve disjoint sets of locations, each agent must return to its start point, and workloads must be balanced across agents, the problem is best framed as a multi-agent Vehicle Routing Problem (VRP) with a fairness constraint.
The objective is therefore to assign all locations to agents such that each location is visited exactly once, routes are geographically efficient, and the number of locations per agent remains balanced. The entire solution was implemented end-to-end in a single notebook.

Data Preparation and Geographic Structure

The dataset consists of latitude–longitude coordinates. Initial geographic analysis showed that nearly all locations lie within a compact region, with only a very small number of extreme points. Distances from the global geographic center were computed using Haversine distance. The distribution showed 3957 total locations, with a mean distance of ~7.6 km, 75th percentile ~8.4 km, and a maximum distance of ~772 km. Only 3 locations were clearly separated from the main region. These were treated as geographic outliers and assigned to a dedicated agent, leaving 3954 main-region locations for balanced routing.

Agent Assignment and Workload Balancing

Geographic assignment was initially performed using K-Means clustering, with the number of clusters equal to the number of agents. Raw K-Means output produced extreme imbalance, with cluster sizes ranging from 1 to 223 locations per agent, making direct routing infeasible.

To correct this, a capacity-constrained rebalancing step was applied. Given 3954 locations and 40 main-region agents, the implied target workload was approximately 99 locations per agent, with acceptable bounds defined as 78 to 117.

Clusters exceeding the upper bound transferred boundary locations to nearby underloaded clusters. Boundary points were selected based on distance to cluster centroids to preserve geographic compactness. After rebalancing, all clusters satisfied the constraints, with a minimum of 78, maximum of 117, and mean of 98.85 locations per agent.

Route Optimization

Once agent assignments were finalized, routing was solved independently for each agent using Earth-curved (Haversine) distances.Two routing approaches were implemented and compared:

- Nearest Neighbor + 2-opt produced reasonable routes but required nested Python loops and quadratic local search.. The solution achieved a total travel distance of 288 km with an average of ~7 km per agent, low variance indicating fairness. Though even with optimizations, total runtime consistently exceeded 5 minutes
- Google OR-Tools (Final Approach) reformulated each agent's route as an independent TSP and solved it using OR-Tools' C++-backed routing engine. Integer distance matrices, Guided Local Search, and strict per-agent time limits were used to balance solution quality and runtime. Using OR-Tools, the solution achieved a total distance of 269.93 km with an average of 6.92 km per agent, low dispersion (std ≈ 2.48 km) indicating balanced workloads, an improved max–min ratio of 4.52, and a significantly reduced runtime of approximately 40 seconds.
- OR-Tools was selected as the final approach because it produced equal or better routes while reducing total runtime to under one minute, making the solution scalable and production-ready.

Results, Visualization, and Conclusion

Final evaluation (excluding the dedicated outlier agent) showed a total distance of ~270 km, mean distance of ~7 km per agent, and standard deviation of ~2.48 km, indicating good fairness. Routes were visualized using both static plots and interactive maps, with each agent rendered in a distinct color. The final solution combines geographic clustering, explicit workload balancing, and efficient routing using an industry-grade solver, satisfying all problem constraints in a realistic and scalable manner.