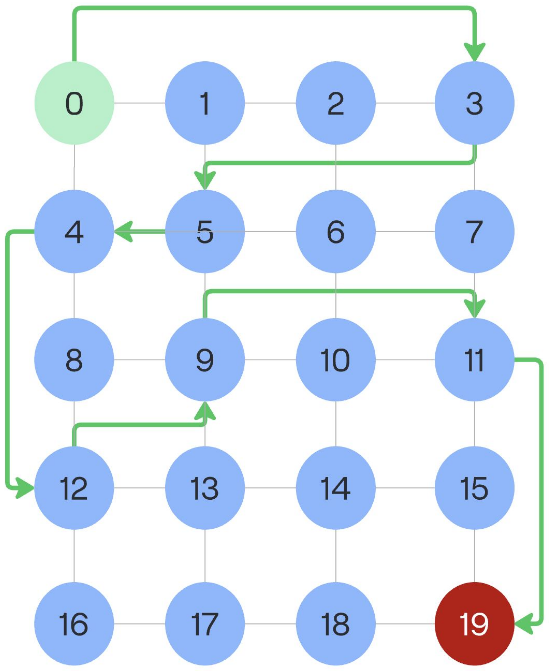
Problem Definition: Bicycle Routing

- Let ***V*** be the set of nodes and ***E*** be the set of edges.
- Decision variable: **P** = (v_1, v_2, ..., v_k) be a route with v_1 = 0 and v_{20} = 19.
- For each edge *(i,j)* $\in$ ***E***:
  - $d_{ij} \in [50,200]$: distance between nodes i and j (in meters)
  - a_{ij}: scenic beauty score (1 = worst, 5 = best)
  - b_{ij}: roughness score (1 = very rough, 5 = smooth)
  - s_{ij}: safety score (1 = very safe, 5 = dangerous)
  - l_{ij}: slope score (1 = gentle, 5 = steep)
- Objectives:
  - Minimize $\Sigma\,(d_{ij})$
  - Maximize $\Sigma(a_{ij}) + \Sigma(b_{ij}) + \Sigma(6 - s_{ij}) + \Sigma(6 - l_{ij})] * (1 / (4.|P|))$
- Dataset: 300 samples selected from a pre-generated set of random routes, each consisting of 7 to 10 nodes.



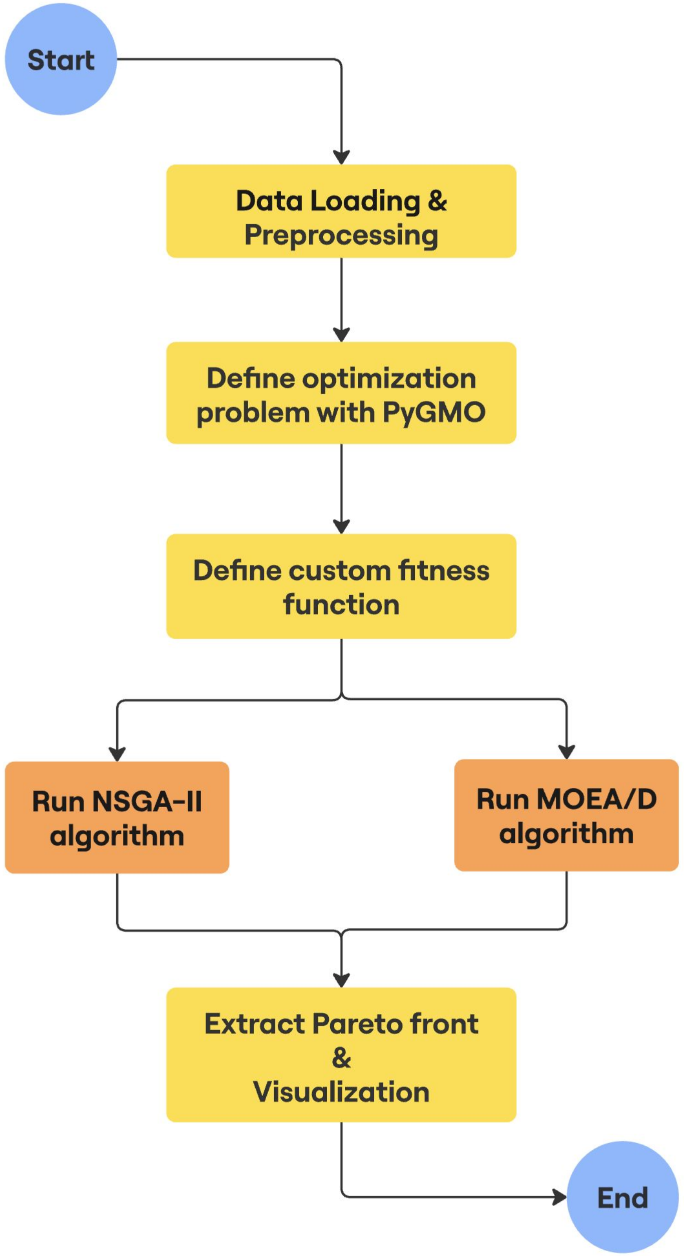
Path	d	a	b	s	l
[0, 3, 8, 5, 14, 16, 2, 9, 11, 19]	1125	20	24	22	25
[0, 3, 5, 14, 17, 8, 6, 7, 12, 19]	981	30	25	29	24

Algorithmic Approach

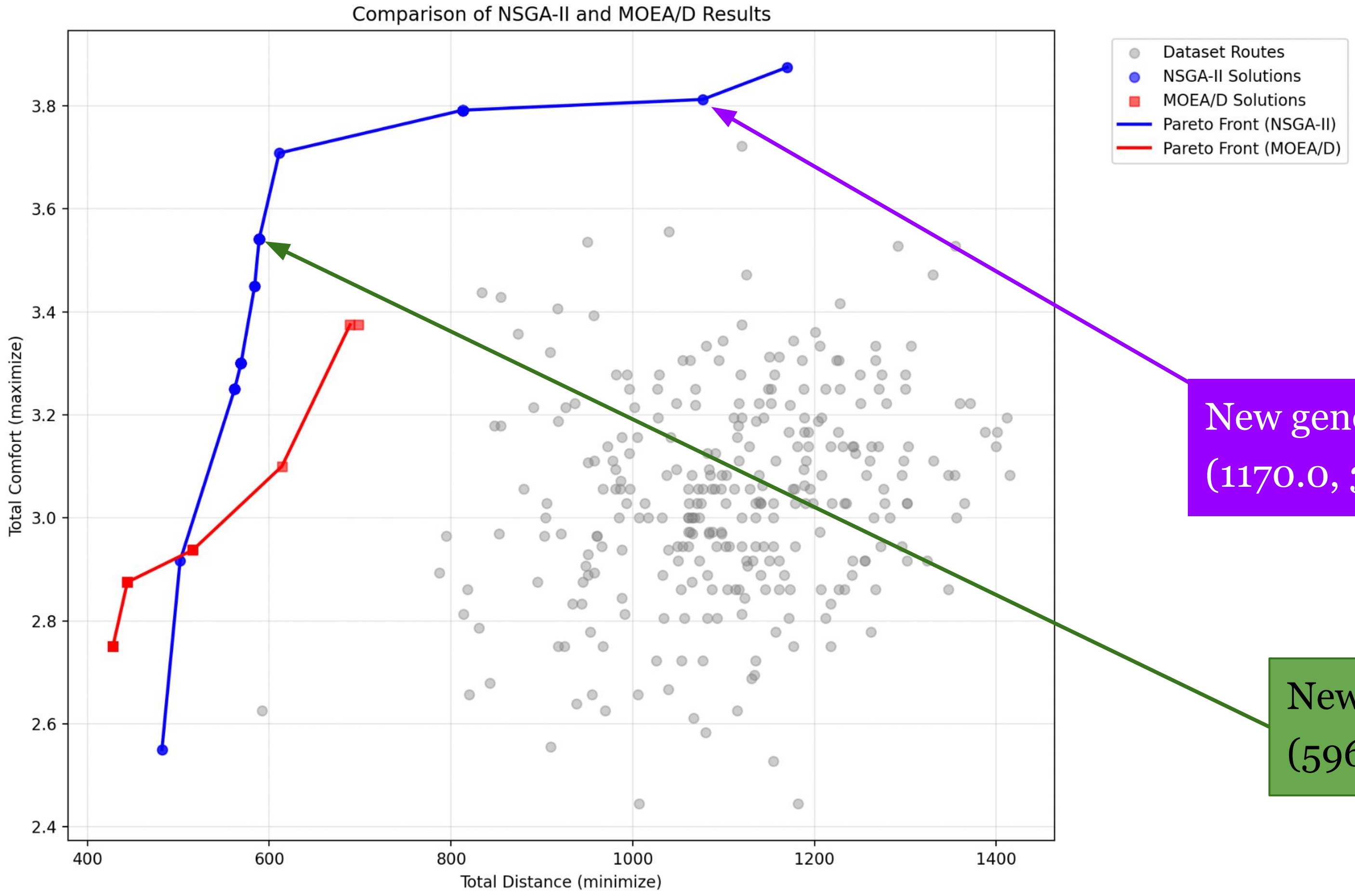
- **NSGA-II (Non-dominated Sorting Genetic Algorithm II):** Uses non-dominated sorting and crowding distance to maintain a diverse set of solutions.

- **MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition):** Decomposes the multi-objective problem into many single-objective sub-problems using weight vectors and solve it in parallel.

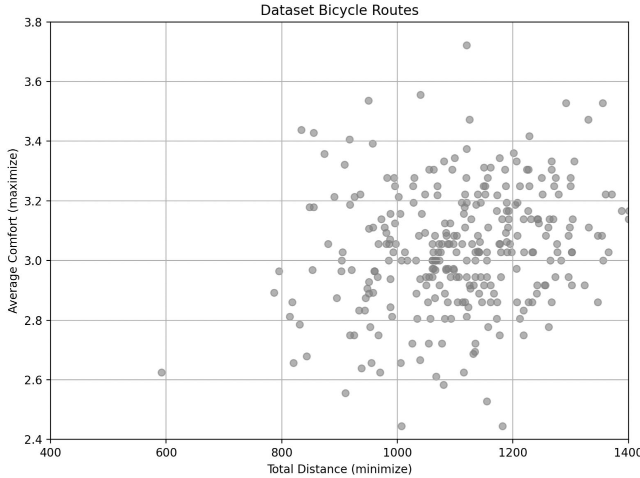
*We used the default value for mutation and crossover for each algorithm.



Results



Original data points



New generated route: [0, 10, 6, 9, 14, 11, 3, 12, 19]
(1170.0, 3.875)

New generated route: [0, 2, 10, 8, 3, 1, 19]
(596, 3.54)

Discussion

- We used PyGMO to efficiently identify optimal trade-offs for bicycle routing using four comfort and distance metrics.
- The new generated Pareto points with NSGA-II and MOEA/D are more efficient than Pareto points form our original dataset, showing the efficiency of the methods.
- Our approach can be applied to other datasets, with simple formatting.

Limitation and Future Work:

- Equal metric weights: currently, we ignore user preferences but will personalize weights to reflect individual priorities.
- Synthetic network: synthetic network may not capture real-world complexity, we will test on real-world data such as OpenStreetMap.