Learning Terrain-Aware Edge Costs for Ant-Colony Route Planning via Genetic Programming

Asia Panizza
University of Trento
Trento, Italy
asia.panizza@studenti.unitn.it

Davide Colosimo
University of Trento
Trento, Italy
davide.colosimo@studenti.unitn.it

Filippo Marcon
University of Trento
Trento, Italy
filippo.marcon@studenti.unitn.it

Erik Nielsen
University of Trento
Trento, Italy
erik.nielsen@unitn.it

Stefano Genetti
University of Trento
Trento, Italy
stefano.genetti@unitn.it

Giovanni Iacca
University of Trento
Trento, Italy
giovanni.iacca@unitn.it

Abstract

Designing transportation routes over complex terrain requires balancing distance with constructability constraints such as slope, elevation, and obstacle avoidance. We propose a hybrid framework that combines Ant Colony Optimization (ACO) with Genetic Programming (GP) on a high-resolution geospatial mesh represented as an 8-neighbor graph enriched with terrain features. GP is used to evolve an edge-cost function via a penalty-driven fitness over multiple start-goal pairs, capturing trade-offs between competing objectives. The learned cost is then exploited by a parallel Min–Max Ant System to efficiently generate feasible routes between user-defined locations. Results show that the proposed method produces computationally efficient routes that reduce steep segments and avoid undesirable areas while maintaining limited detours.

CCS Concepts

• **Applied computing** → *Transportation*; • **Theory of computation** → **Shortest paths**; **Bio-inspired optimization**; • **Human-centered computing** → Geographic visualization.

Keywords

Ant Colony Optimization, Genetic Programming, Terrain-aware Routing, Geospatial Graphs, Route Planning

ACM Reference Format:
Asia Panizza, Davide Colosimo, Filippo Marcon, Erik Nielsen, Stefano Genetti, and Giovanni Iacca. 2026. Learning Terrain-Aware Edge Costs for Ant-Colony Route Planning via Genetic Programming. In *Genetic and Evolutionary Computation Conference (GECCO Companion '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3795101.3814665

1 Introduction

Planning and designing new transportation links (roads or railways) over uneven terrains is inherently multi-objective: minimizing distance is rarely compatible with minimizing construction effort, slope, or crossings of undesirable areas. These trade-offs often necessitate iterative manual adjustments. Recent research, however, has focused on automating this process through multi-objective optimization and integrated geospatial frameworks [7].

To address this limitation, we propose a solution that learns a terrain-aware edge-cost model and then uses it to guide a metaheuristic route planner. We represent the region of interest as a mesh graph where nodes correspond to geographic coordinates. The routing task is to connect a set of user-defined locations on this graph.

Two algorithmic components are combined in the proposed method. First, strongly typed Genetic Programming (GP) evolves a closed-form expression that maps local terrain descriptors (i.e., slope, water presence, distance) to an edge cost. Second, Ant Colony Optimization (ACO), implemented as a parallel Min–Max Ant System (MMAS) with multiple colonies, searches for low-cost routes using these learned costs. In summary, our main contributions are:

- a geospatial mesh-graph enriched with terrain features;
- a GP procedure that evolves an edge-cost expression via a penalty-based fitness over diverse routing instances;
- a parallel multi-colony MMAS configuration tailored to generating diverse, terrain-aware routing alternatives.

An overview of the overall framework is shown in Figure 1.

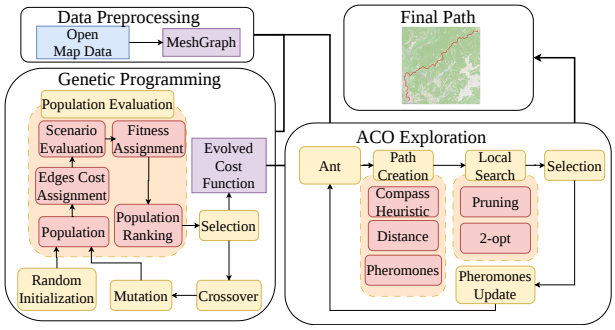


Figure 1: Overview of the proposed framework.

2 Methods

In the following, we first describe the approach for the construction of the terrain graph (Section 2.1), followed by the presentation of

GECCO Companion '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2488-6/2026/07
https://doi.org/10.1145/3795101.3814665

the GP-driven edge-cost model evolution (Section 2.2) and then of the ACO solver used to produce routes and route sets (Section 2.3).

2.1 Terrain Mesh-graph Construction

The data sources are a high-resolution GeoTIFF elevation raster for the region of interest and an OpenStreetMap (OSM) extract. A set of key locations (waypoints to connect) is provided as geographic coordinates. A bounding box encompassing all the key locations is automatically generated to focus resolution on the relevant area.

The *Mesh-graph* is constructed by sampling the geographical area defined by the bounding box using a square grid. We define the graph resolution as the desired number of sampled points on each axis of the grid.

The *nodes* of the Mesh-graph are the sampled points, and they are identified by their latitude and longitude. Each node additionally stores its elevation (sampled from the GeoTIFF elevation file), enabling computation of planar distance and elevation difference between adjacent nodes.

We then annotate nodes with feature information extracted from the OSM file. Feature layers of interest(such as water bodies) are converted into polygon masks, and nodes intersecting these polygons receive the corresponding tag. These tags are later used by the evolved cost function (see Section 2.2).

Edges follow an 8-neighbor topology, enabling both cardinal and diagonal moves. This structure reduces the directional bias to ≈8% worst-case path elongation compared to ≈41% of a 4-neighbor approach [5].

The final routes generated by ACO (see Section 2.3) are rendered to HTML (via `folium`) over an OSM basemap to enable visual inspection (see Figures 4a and 4b).

2.2 Evolving an Edge-cost Model with GP

To derive edge costs for ACO without hand-tuned weights, we evolve a closed-form expression using strongly typed GP [4] implemented in DEAP [3]. Terminals for the expression are drawn from edge metadata: edge steepness, a boolean indicator of whether at least one endpoint lies in a water polygon, and the planar distance between nodes. The primitive set includes addition, subtraction, multiplication, division, power, logarithm, and negation, all operating on real-valued arguments. Comparison operators such as "greater than or equal to" (≥), "greater than" (>), "less than or equal to" (≤), and "less than" (<) are also contained within the primitive set. We include a conditional operator to allow the expression to branch on water presence as well as on the comparison operators within the set. All edge metadata are normalized via max-normalization prior to evolution.

Instead of fixed numeric terminals, we use ephemeral constants sampled uniformly in $[0, 1]$. The GP algorithm uses double-tournament selection, one-point crossover, and uniform mutation operators affecting both tree structure and constant values. To avoid bloating, we enforce limits on maximum tree height and total node count. During each generation, every individual is compiled into an edge-cost function and used to generate a weighted adjacency matrix. Bidirectional Dijkstra's algorithm is then applied over multiple start-goal pairs, called *scenarios*, selected to span different altitudes, distances, and proximity to water. The fitness of each individual is defined as the average performance across all resulting paths, calculated by minimizing the following penalty function:

$$\text{Penalty} = \frac{d_{path}}{100} \cdot \left(100 + e^{water\%\cdot x} + e^{steepness\%\cdot x} - 2\right) \quad (1)$$

where d_{path} is the sum of the length of all edge planar distances within the path, $water\%$ is the fraction of edges in the path that intersect water areas, and $steepness\%$ is the average steepness across all path edges.

The parameter x is a region-dependent hyperparameter that controls the penalization of steepness and water crossings. It is introduced to account for differences in terrain morphology: in mountainous areas, excessive penalization of steepness can lead to overly long detours, while in flatter regions a stronger penalty is needed to discourage impractical routes. Therefore, x is empirically tuned based on the morphology of the region at hand.

2.3 Multi-colony MMAS for Routing

As mentioned, the routing task is an approximation to a Steiner tree connecting the key nodes on the mesh graph. The solver is based on the Min–Max Ant System (MMAS) [6] to reduce stagnation and is optimized for multi-core execution to handle dense 8-neighbor connectivity. To encourage diverse routing alternatives, we adopt a multi-colony approach in which separate colonies operate in parallel, each maintaining its own pheromone type. Ant transitions balance colony-specific pheromones, a dominance term, and a composite heuristic. The probability $P_{ij}^{k,c}$ of an ant k in colony c moving from node i to j and the dominance-weighted pheromone Φ_{ij}^{c} used to bias differentiation are:

$$P_{ij}^{k,c} = \frac{[\Phi_{ij}^{c}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in\mathcal{N}i^{k}}[\Phi il^{c}]^{\alpha}[\eta_{il}]^{\beta}} \qquad \Phi_{ij}^{c} = \tau_{ij}^{c}\cdot\left(\frac{\tau_{ij}^{c}}{\sum_{m\in C}\tau_{ij}^{m}}\right)^{\lambda} \quad (2)$$

where τ_{ij}^{c} is the pheromone of colony c, C is the set of colonies, and λ is the dominance exponent. This biases ants toward edges where their colony is relatively more represented.

The heuristic η_{ij} integrates the learned edge cost and a compass term and is inversely proportional to their sum:

$$\eta_{ij} = \frac{1}{C_{ij} + D(j, K_{near})} \quad (3)$$

where C_{ij} is the GP-evolved cost and $D(j, K_{near})$ is the Euclidean distance from node j to the nearest waypoints K_{near}. This compass term guides ants toward waypoints and accelerates convergence. Edges incident to target nodes have their heuristic scaled by σ.

We incorporate elitism [2] in the global pheromone update: for each iteration, only the best E ants across colonies deposit pheromones, scaled by an elitism factor γ.

Finally, each final path undergoes the following post-processing:

- Path Pruning: an algorithm that identifies and removes redundant loops or backtracking segments within the 8-neighbor mesh.
- Two-opt Heuristic: a local search that smoothens the path.

Overall, the multi-colony exploration, dominance-weighted pheromones, elitist reinforcement, and local refinement are designed to produce multiple high-quality and distinct solutions.

3 Experimental Setup

This section summarizes the test regions, data sources, routing instances, and evaluation protocol used to assess the proposed GP+ACO pipeline, including the evaluation metrics and the computational setup.

Regions and Data Sources. We focus the experimentation on two geographical areas with different characteristics:

- Trentino (Northern Italy). A mountainous alpine area characterized by sharp elevation gradients (avg. steepness 9.74%). To balance these prominent slopes, we set $x = 15$.
- North-Western Campania (Southern Italy). A hilly area between the Tyrrhenian Sea and the Apennine foothills. Given a lower avg. steepness of 4.52%, we set $x = 30$.

Routing Instances. We test the proposed approach on eight real-world routing instances within the selected regions: each instance specifies a small set of locations to be connected. All experiments use a mesh resolution of 200 nodes. The evaluated routes are:

(1) **Trentino-1**: Pergine Valsugana, Oltrecastello, Trento, Sardagna, Sopramonte, Vaneze, Vason;
(2) **Trentino-2**: Trento, Lavis, Verla, Cembra, Grauno, Capriana, C. di Fiemme;
(3) **Trentino-3** Taio, Malgolo, Sarnonico, Mondola, Caldaro, Laives;
(4) **Trentino-4** Trento, Civezzano, Miola, Sover, Casatta, Cavalese.
(5) **Campania-1**: Sessa Aurunca, Roccamonfina, Conca della Campania, Sipicciano, Teano;
(6) **Campania-2**: Sessa Aurunca, Conca della Campania, Pietramelara, Alvignano, Pontelatone, Capua;
(7) **Campania-3**: Presenzano, Ailano, Sant'Angelo d'Alife, Dragoni, Piana di Monte Verna, Pozzillo, Ciamprisco, Sessa Aurunca;
(8) **Campania-4**: Castel Volturno, Sessa Aurunca, Teano, Capua, Mondragone;

GP Evaluation Protocol and Hyperparameters. We executed GP with tournament size 3 and parsimony of 1.3, 70% crossover probability, and 25% mutation probability. Using the software framework Optuna [1] for hyperparameter tuning, we eventually ran the algorithm for 50 generations with a population size of 3500 and 60 scenarios.

ACO Evaluation Protocol and Hyperparameters. Each evolved cost function is used within our multi-colony MMAS solver. We keep the ACO configuration fixed to empirically effective values. The configuration used in the following experiments is $\alpha = 1$, $\beta = 3$, $\rho = 0.1$, $\lambda = 2$, $\gamma = 2$, $\sigma = 20$, and $q_o = 0.1$. Each evolved cost function was tested multiple times at resolution 200 with 50 ants. Runs were capped at 100 iterations; only the five best ants updated pheromones. To reduce stagnation, we reset the search process after 10 iterations with no improvements.

Metrics and Output Analysis. We summarize solution quality using (i) average route length (computed over the terrain mesh and, when applicable, as a 3D polyline using elevation, and expressed in km) and (ii) average steepness (%) along the selected edges. In addition, we qualitatively inspect the generated routes by rendering them on an OSM basemap, enabling comparison against existing roads and visual identification of undesirable crossings (e.g., water).

Computational Setup. All experiments were conducted on a ProLiant DL560 Gen10 server, with 4 Xeon6252N processors for a total of 96 cores. Effective single-job requests were set at 50 cores and 30GB RAM. All experiments were executed with a parallel Python implementation using multi-core processing. Due to the computational cost of repeated GP evaluations and ACO runs, hyperparameter tuning and ablations were performed under limited budgets, rather than through exhaustive search. We make our code publicly available on GitHub[1].

4 Experimental Results

This section reports (i) the outcome of the GP search for terrain-aware edge-cost functions and (ii) the routing performance obtained when these learned costs are used to guide the ACO solver. We first summarize the best-performing GP individuals and the corresponding algorithm configurations, then evaluate the resulting routes on multiple real-world instances in the selected regions.

4.1 Evolved Cost Functions

We selected the two best-performing individuals for each of the two regions. The selected individuals are denoted as cost functions CF1-CF4. Table 1 reports the configurations that evolved them, and their corresponding achieved fitness values.

Table 1: Performance summary of the best-performing cost functions, with the related experimental configurations producing those functions.

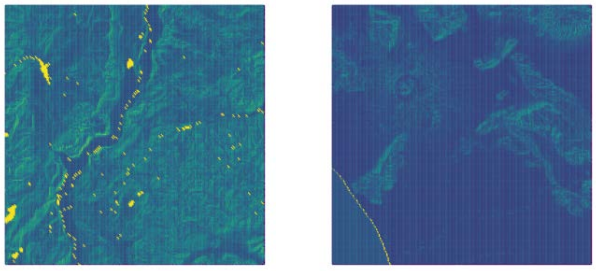
Cost Function	Configuration				Fitness
	Region	Pop. Size	No. Scenarios	No. Generations	
CF1	Trentino	3500	60	50	86.97
CF2	Trentino	3500	60	50	86.97
CF3	Campania	3500	60	50	97.73
CF4	Campania	3500	60	50	97.73

To validate the GP search, we visualize the evolved functions as a Cost Landscape (Figure 2). This is a geographic heatmap where the abstract GP expression is mapped back onto the mesh to represent the spatial distribution of edge costs. It acts as an interpretability tool, allowing designers to visually verify that high-cost zones correctly align with steep gradients and water bodies.

4.2 Optimized Routes

We evaluated the routing performance of the four top-performing evolved cost functions from Table 1 across the eight routing instances detailed in Section 3. The results indicate that these optimized functions effectively guide the solver to minimize steep gradients without incurring significant distance penalties, thereby producing routes that closely parallel existing infrastructure. As illustrated by the comparative metrics in Figure 3 and the representative route visualizations in Figure 4, the proposed methodology demonstrates considerable robustness and adaptability across diverse and complex topographical scenarios.

[1] https://github.com/DIOL-UniTN/aco-gp-communication.git



(a) Edge-cost distribution for CF1 in the Trentino region. **(b) Edge-cost distribution for CF3 in the Campania region.**

Figure 2: Examples of "Cost Landscapes" for two selected evolved cost functions. Higher costs mark steep or constrained areas; lower costs follow favorable terrain.

5 Discussion and Limitations

The proposed framework reduces manual trial-and-error in route planning by learning a transparent, terrain-aware cost model to guide a metaheuristic solver based on ACO. It rapidly generates constructability-aware alternatives prior to expensive field surveys. However, several limitations remain.

Practical Adoption. While valuable for decision support and surfacing trade-offs, the framework does not replace downstream engineering. Deployment-grade use must integrate uncaptured factors like geology, land ownership, and environmental regulations.

Hyperparameter Sensitivity. Interacting ACO parameters complicate fine-tuning and performance attribution. We used sequential tuning to avoid a costly exhaustive search, but more systematic optimization would be needed to find optimal parameters.

Computational Scalability. The proposed pipeline is computationally intensive, requiring repeated shortest-path computations on large graphs during GP and ACO evaluation. The memory overhead scales quadratically with the graph resolution R, following $O(R^2 \cdot N)$, where N represents the population size (number of GP individuals or ants). To mitigate these demands, distributed execution or multi-fidelity surrogate strategies may be necessary.

6 Conclusions and Future Work

We presented a hybrid routing framework that learns terrain-aware edge costs with GP and uses them within a multi-colony MMAS solver on a mesh graph. By evolving a compact, explicit cost expression from terrain descriptors, the proposed system reduces reliance on hand-tuned weighting schemes while retaining a transparent model that can be inspected and adapted by practitioners. Empirically, the learned costs steer ACO toward routes that better respect terrain constraints (e.g., avoiding steep segments and undesirable areas) while keeping detours limited.

To the best of our knowledge, this work is the first to integrate GP-evolved, feature-based edge-cost models directly into an ACO/MMAS routing framework for terrain-aware transportation planning on real-world geospatial meshes. Future work includes more systematic hyperparameter optimization, broader benchmarking across regions with different geomorphology, and extending the feature set (e.g., land use, protected areas, and existing infrastructure) to better reflect real engineering constraints.

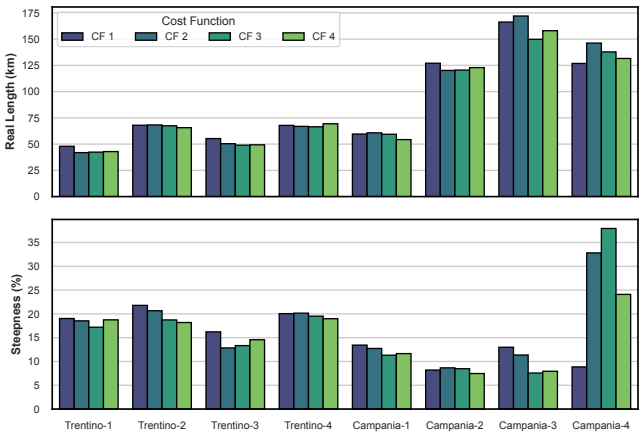
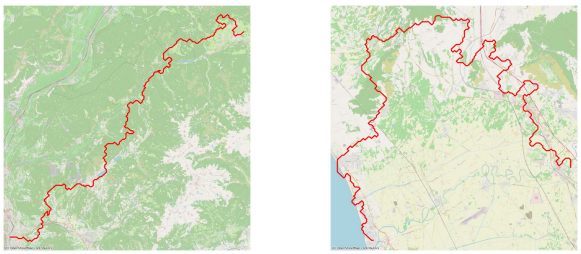


Figure 3: Average route length (top) and average steepness (bottom) on each test case with the Cost Functions (CF) corresponding to the solutions reported in Table 1.



(a) Trentino-2 with CF1 **(b) Campania-1 with CF3**

Figure 4: Representative best routes produced by the proposed framework on two selected instances.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. 2623–2631 pages. https://doi.org/10.1145/3292500.3330701

[2] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26, 1 (1996), 29–41. https://doi.org/10.1109/3477.484436

[3] Fortin, Félix-Antoine and De Rainville, François-Michel and Gardner, Marc-André Gardner and Parizeau, Marc and Gagné, Christian. 2012. DEAP: Evolutionary algorithms made easy. *The Journal of Machine Learning Research* 13, 1 (2012), 2171–2175. https://doi.org/10.5555/2503308.2503311

[4] David J Montana. 1995. Strongly typed genetic programming. *Evolutionary computation* 3, 2 (1995), 199–230. https://doi.org/10.1162/evco.1995.3.2.199

[5] Alex Nash and Sven Koenig. 2013. Any-angle path planning. *AI Magazine* 34, 4 (2013), 85–107. https://doi.org/10.1609/aimag.v34i4.2512

[6] Stützle, Thomas and Hoos, Holger H. 2000. MAX–MIN ant system. *Future Generation Computer Systems* 16, 8 (2000), 889–914. https://doi.org/10.1016/S0167-739X(00)00043-1

[7] Linlin Zhao, Zhansheng Liu, and Jasper Mbachu. 2019. Highway Alignment Optimization: An Integrated BIM and GIS Approach. *ISPRS International Journal of Geo-Information* 8, 4 (2019), 28 pages. doi:10.3390/ijgi8040172