

Article

A Novel Field Routing Approach for Unmanned Ground Vehicles in Steep-Slope Vineyards

Mhd Ali Alshikh Khalil ^{1,2,*}  and Maria Rita Palattella ¹ 

¹ Luxembourg Institute of Science and Technology (LIST), 4362 Esch-Belval Esch-sur-Alzette, Luxembourg; mariarita.palattella@list.lu

² Department of Computer Science, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg

* Correspondence: mhdali.alshikhkhalil@list.lu

Abstract

Autonomous ground vehicles operating in steep-slope vineyards must reason jointly about three coupled factors: the order of waypoints across multiple polygon sub-fields, the points at which the route crosses sub-field boundaries, and terrain-aware travel cost in a Digital Elevation Model (DEM). Existing planners address one side at a time: Coverage Path Planning (CPP) libraries treat each field in isolation, vineyard-specific A* variants reason about terrain and rollover safety within one vineyard, and energy-aware planners optimise coverage within a single field. This paper presents a novel field-scale planner, which optimises a tour over the waypoints of all sub-fields with one Travelling Salesman Problem (TSP) whose travel costs come from three-dimensional polylines sampled from the DEM and a travel-cost model of rolling resistance, climbing work, and a differential-drive pivot-turn term. Transitions between sub-fields are routed through chains of adjacent sub-fields. Four strategies are evaluated: a greedy nearest-neighbour (NN) heuristic and three Simulated Annealing (SA) TSP variants minimising distance (SA-D), energy (SA-E), or time (SA-T). Evaluated on two Moselle sites in Luxembourg, the objective-matched variants reduce mission cost over the nearest-neighbour baseline by up to 21%, on average, and 40% on the best mission, within an integrated pipeline from public geodata to executable, terrain-aware routes.

Keywords: precision agriculture; unmanned ground vehicle; terrain-aware routing; objective-aware routing; travelling salesman problem; field routing; digital elevation model; simulated annealing; vineyard environment



Academic Editor: Sergey Y. Yurish

Received: 1 June 2026

Revised: 9 July 2026

Accepted: 9 July 2026

Published: 15 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The adoption of agricultural robots has grown steadily as platforms shift from manual operation towards autonomous control [1], and Unmanned Ground Vehicles (UGVs) are central to this transition: they carry the sensors and tooling for scouting, spraying, harvesting, and in-field logistics, reducing labour cost and enabling data-driven agricultural applications [2]. Vineyards and orchards represent a particularly active application domain because their row-structured layout, narrow alleys between vines or trees, and the continuous care cycle of perennial crops require repeated mechanised intervention.

Steep-slope viticulture poses a harder challenge: terrace walls, cadastral boundaries, and access roads fragment a single estate into many small polygon sub-fields, and on a terraced slope, most traversals are inclined, so a ground vehicle pays an energy penalty for climbing work and faces a rollover hazard that bounds the safe roll and pitch angles [3,4].

The same row geometry that makes these crops attractive for automation also restricts vehicle motion to a sparse network of alleys and headlands.

These conditions place an autonomous platform under two simultaneous constraints. First, the workspace is fragmented, so a single mission must order and route visits across several disjoint sub-fields. Second, the cost of moving between two points depends on the Digital Elevation Model (DEM) along the connecting polyline, not on planar distance. Therefore, a UGV must reason jointly about three coupled factors: the order in which it visits its inspection waypoints, the points at which its route crosses sub-field boundaries, and the terrain-aware cost of travelling between them. Existing planners address these requirements only in isolation: Coverage Path Planning (CPP) libraries treat a single field as an isolated polygon and generate swath-based sweeps to cover it [5], vineyard-specific extensions of the A* search algorithm reason about soil compaction [6] and rollover stability [3] within one vineyard, and energy-aware planners account for the robot's battery budget and trigger return-to-charge trips yet plan coverage within a single workspace [7]. What remains open is the combined system: routing a sparse set of waypoints across several sub-fields, where waypoint ordering, the choice of polygon-boundary crossing points, and the elevation-induced energy of the resulting polyline all interact.

In this paper, we address this gap by designing the field-scale planner, a routing method that optimises a tour over the inspection waypoints of all sub-fields of a vineyard with a single Travelling Salesman Problem (TSP) formulation. Its travel costs are computed on three-dimensional polylines sampled from the DEM through a travel-cost model. Transitions between sub-fields are routed through chains of adjacent sub-fields chosen to minimise the active objective, so cross-sub-field travel is not restricted to fixed alley endpoints. The waypoint tour is optimised by one of four strategies: a greedy nearest-neighbour (NN) heuristic, which serves as the baseline, and three Simulated Annealing (SA) TSP variants that minimise distance (SA-D), energy (SA-E), or time (SA-T). The remainder of this section positions the field-scale planner against the most relevant literature. Section 1.1 reviews terrain-aware field planning, and Section 1.2 reviews tour optimisation and field routing; the specific contributions are enumerated in Section 1.3.

1.1. Terrain-Aware Field Planning

CPP for agricultural vehicles is dominated by Fields2Cover [5], a Robot Operating System (ROS)-integrated library that implements state-of-the-art coverage methods and objective functions across the headland-generation, swath-generation, route-planning, and path-planning stages. Several lines of work extend this base towards three-dimensional terrain and tighter kinematic constraints. Pour Arab et al. proposed H-CCPP [8], a hybrid three-dimensional planner for wheeled agricultural robots that incorporates terrain inclination to reduce soil erosion and energy use, extending the same authors' two-dimensional One-Step Complete Coverage Path Planner (O-CCPP) [9]. Mier et al. introduced three continuous-curvature headland-turn planners for non-holonomic robots and reported that field coverage rose from 94.73% under a constant-offset baseline to 97.29% under their method [10]. Schöneegg et al. [2] targeted coverage in orchards and vineyards under explicit kinematic-vehicle constraints: they built a two-dimensional occupancy grid from a three-dimensional point cloud, extracted a topological graph with a Voronoi diagram, formulated row coverage as a rural postman problem to obtain an Euler tour over user-selected rows, and refined the high-level path with a hierarchical A* algorithm that enforces turning-radius and vehicle-dimension constraints. The recent survey of Gao et al. [1] organises this body of work into four families: classical search methods such as the A* and Dijkstra algorithms; metaheuristic bionic methods including the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and SA; sampling-based

planners such as the Rapidly Exploring Random Tree (RRT); and learning-based methods. These coverage planners share an objective complementary to ours: they cover every row of one field, whereas the field-scale planner routes a UGV through a sparse waypoint set spanning multiple sub-fields, placing its outer tour-optimisation layer in the metaheuristic family of this taxonomy.

A parallel strand augments coverage with explicit energy and terrain cost. Wei and Isler trained a data-driven model that predicts the energy of a ground robot on uneven terrain based on terrain geometry and motion direction, motivated by the observation that physics-based models calibrated on flat ground do not generalise to rough terrain [11]. Shen et al. coupled an energy-consumption model with a GA for coverage path planning on hilly farmland and reported a 17.4% energy reduction relative to a distance-optimised path [12]. Dogru and Marques proposed ECO-CPP [7], an online coverage planner that models the robot's battery budget and schedules return trips to a charging station when one pass is insufficient. Sharma et al. cast the same trade-off as a multi-objective search in TA-NSGA-II, a terrain-adaptive variant of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) that decomposes energy consumption into three objectives over total travel distance, cumulative elevation change, and turn count under a complete-coverage constraint [13]. The field-scale planner draws on this energy-modelling lineage but applies a different scalarisation: it lifts each polyline onto the DEM and reduces the travel-cost model to a single scalar energy or time objective optimised with SA over a TSP cost matrix. Instead of producing a set of trade-off routes via multi-objective optimisation, we optimise one objective at a time.

This terrain-aware reasoning is most developed in vineyard- and orchard-specific planning on steep slopes, the setting closest to the scenario under study in this work. The line originates with the AgRob V16 platform of Santos et al., who extended A* with a soil-compaction map that records traversed cells to discourage repeated passes [6] and, later, with a centre-of-mass-aware stability constraint that bounds roll and pitch to keep the robot inside its static stability margin, supported by a PointCloud-to-grid-map-and-DEM algorithm (PC2GD) that builds the occupation grid map and DEM [3]. Subsequent work addressed neighbouring sub-problems: Wei et al. treated headland turning in cluttered orchards by combining a front-end planner with a back-end optimisation step inspired by autonomous-driving stacks [14]; Hrabar et al. estimated the energy consumption of an all-terrain mobile manipulator on vineyard slopes up to 27.6°, addressing energy estimation rather than route optimisation [4]; and Ye et al. planned smooth two-dimensional paths for an orchard robot with a continuous bidirectional Quick-RRT* variant that minimises the combined path-length and heading-angle (smoothness) cost without modelling terrain slope [15]. Each of these planners operates within a single vineyard or orchard, so the field-scale planner is composable with them rather than a replacement: a Santos-style centre-of-mass-aware A* can run as the within-row controller while the field-scale planner supplies the outer tour optimisation across sub-fields on polylines sampled from the DEM. The mobility of such platforms on uneven terrain is, itself, an active modelling problem, for instance, the maximum speed at which a rocker-bogie suspension can cross a bump before a wheel lifts off the ground [16].

1.2. Tour Optimisation and Field Routing

Because the field-scale planner optimises a tour over a waypoint set under a TSP objective, exact and learned TSP solvers are relevant to its inner ordering problem. The Lin–Kernighan–Helsgaun (LKH) family and its learned augmentations represent the state of the art with respect to this sub-problem: NeuroLKH guides the LKH search with a sparse graph network and generalises across problem sizes [17], VSR-LKH couples reinforcement

learning updates with LKH and reports results on 236 TSP benchmark instances of up to 85,900 cities [18], and the POMO end-to-end policy method reports a 0.14% optimality gap on the 100-city TSP at more than an order of magnitude faster inference than prior learned heuristics [19]. These solvers optimise an abstract cost matrix and are independent of the terrain-aware travel-cost model that distinguishes the present work; the field-scale planner uses SA seeded from a nearest-neighbour route in this layer and could adopt any of them in its place.

The reformulation becomes domain-specific once the cost matrix encodes agricultural geometry and energy. Kuang et al. detected trees from aerial imagery; segmented high-density tree regions with kernel density estimation; and solved the resulting route with greedy, ACO, and Monte Carlo reinforcement learning solvers, reporting a 16.9% reduction in travel distance for the Monte Carlo solver against ACO on irregular plantations [20]. AgriPath addresses multi-objective ground planning in maize fields, quantifying energy across scenarios of increasing complexity [21]. Zheng et al. extracted tree-row endpoints to generate key waypoints and order them into a global coverage route for orchard UGVs under kinematic vehicle constraints [22]. The Soft Actor–Critic (SAC) energy-aware CPP framework of Wu et al. reaches coverage above 90% with return-to-base constraints and improves coverage by 13.4 to 19.5% over RRT, PSO, and ACO baselines [23]. The present work extends our previous conference paper on terrain-aware UGV path planning in sloped orchard environments [24], which introduced the terrain-aware travel-cost model that this article extends, and reports a 28% energy reduction and a 12% completion-time reduction for SA energy versus a nearest-neighbour baseline on a single sloped orchard. The field-scale planner differs from all of these models along two coupled axes: it combines the waypoints of multiple sub-fields into one global route rather than covering or routing within one field, and its travel cost is terrain-aware energy or time computed along three-dimensional polylines sampled from the DEM. The present article extends our previous study with new experimental results on a real Moselle vineyard and an end-to-end pipeline from public Luxembourg geodata to executable routes.

1.3. Contributions and Organisation

The evaluation of the proposed routing strategy was conducted on two vineyard sites in the Moselle wine region of Luxembourg. The core planner is independent of the source country and runs on any DEM and polygon set in a projected metric coordinate reference system. Building on our prior conference study [24], which routed a single sloped orchard with a terrain-aware energy model and simulated-annealing tour optimisation, this article makes the following contributions:

- A multi-sub-field routing formulation that combines the waypoints of all sub-fields of a vineyard into one global TSP;
- An extension of the terrain-aware travel-cost model with a load-dependent motor-efficiency term and a differential-drive pivot-turn term, yielding a scalar energy cost suitable for use inside a TSP cost matrix;
- A cross-sub-field routing module that connects sub-fields through chains of adjacent sub-fields and routes terrain-aware transitions across the service strips between sub-fields so that travel between sub-fields is not restricted to fixed alley endpoints;
- An integrated end-to-end pipeline that turns public geodata (a Digital Terrain Model (DTM), an orthophoto, and land-parcel polygons) into operational sub-fields with abstracted vine rows, ready for executable terrain-aware routes for the UGV.

Across both Moselle sites and all evaluated waypoint counts, each objective-matched SA variant attains the lowest cost on its own metric, reducing distance, energy, and time

on individual missions by up to 29.7%, 40.4%, and 28.1% respectively as described in the Results (Section 4).

The remainder of this paper is organised as follows. Section 2 documents the open pipeline that produces the planner's four data products from public geodata. Section 3 formalises the field-scale routing problem, the terrain-aware travel-cost model, and the four tour-optimisation strategies. Section 4 reports the performance evaluation of the proposed method, tested at two sites along the Moselle in Luxembourg. Section 5 discusses limitations and future work. Section 6 concludes the work, highlighting the main findings.

2. Materials and Methods

The field-scale planner consumes four site-specific data products: a DTM raster, a high-resolution orthophoto, a set of operational sub-field polygons, and a set of vine-row line strings. This section documents the open pipeline that produces those four products from public Luxembourg geodata. The planner itself is described in Section 3. Figure 1 summarises the end-to-end data flow from the public geodata to the planner: the blue nodes are the inputs, the orange nodes the processing stages, the green node represents the four planner-ready data products, and the red node corresponds to the final route. Section references on the stage nodes mark where each is described.

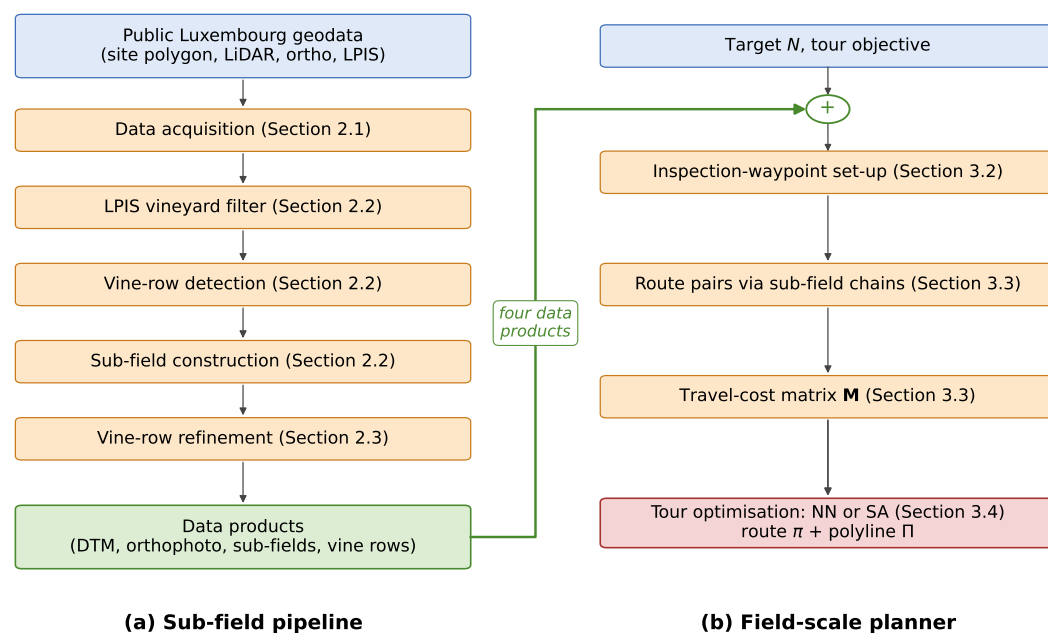


Figure 1. End-to-end flow from public Luxembourg geodata to terrain-aware routes. (a) The sub-field pipeline turns the four public inputs into the four planner-ready data products, which (b) feed the field-scale planner, as detailed in Section 3: it collects the specified waypoints into a single global set, evaluates the $N \times N$ travel-cost matrix on polylines sampled from the Digital Elevation Model (DEM), and optimises the waypoint tour. LiDAR = Light Detection and Ranging; LPIS = Land Parcel Identification System.

2.1. Study Sites and Data Acquisition

Two sites in the Luxembourg Moselle wine region are used (Figure 2): *mosel_south* south of Remich and *mosel_north* on the same hillside to the north, both on terraced slopes. The two sites were selected to contrast two sub-field topologies: the sub-fields of *mosel_north* are strung along the hillside in an open, chain-like arrangement, so a route's climbing work is dominated by the order in which the slope is traversed, whereas those of *mosel_south* span both sides of a hill in a closed, densely connected arrangement in which ascents and descents largely offset across the route. Evaluating on both rather than on a single

representative site exposes how much the terrain-aware objective gains for each site. The two layouts bound the range of topologies over which the terrain-aware gain varies, so together, they capture how the gain behaves across that range rather than at a single site; the sensitivity analysis preserves this ordering (Section 5). The planner does not depend on either site, since it runs on any DEM and polygon set in a projected metric coordinate reference system (Section 1.3).



Figure 2. The two Luxembourg Moselle test sites in the 10 cm orthophoto, with the user-drawn enclosing site boundary in yellow: (a) *mosel_south* south of Remich and (b) *mosel_north* on the same hillside to the north. The axis coordinates are in the Luxembourg Reference Frame (LUREF).

For each site, the user draws an enclosing polygon on the public Luxembourg Geoportal [25] and exports the result as a GeoJSON file in World Geodetic System 1984 (WGS84), including longitude/latitude. The polygon is reprojected to the Luxembourg Reference Frame (LUREF, EPSG:2169). Two raster products are then fetched into the site's bounding box: (a) a DTM windowed from the country-wide Lidar 2024 Cloud Optimized GeoTIFF (COG) of the cadastre service [26] via HTTP range requests and (b) a Ground Sampling Distance (GSD) orthophoto stitched from tiled Web Map Service (WMS) GetMap calls [27].

2.2. Sub-Field Polygon Pipeline

Land Parcel Identification System filter. The Luxembourg Land Parcel Identification System (LPIS) registry [28] is queried for parcels intersecting the user polygon. Each LPIS record carries a Hierarchical INSPIRE Land-Use Classification System (HILUCS) code; only parcels classified as a vineyard are kept. Each filtered parcel is then clipped to the user polygon, and any clip whose intersection area falls below a small sliver threshold is dropped to remove numerical-precision noise on boundary touches.

Vine-row detection. The pre-trained DeepLSD model of Pautrat et al. [29] runs on the orthophoto at three scales (1.0, 0.5, and 0.3) and emits a per-pixel line-attraction field, from which a classical line-segment detector [30] recovers line segments. Segments shorter than a minimum row length are dropped; the remainder are deduplicated by perpendicular distance and collinearly merged within fixed gap and angle thresholds. The result is a per-sub-field list of vine-row line strings, computed twice in the pipeline: a first pass uses raw LPIS polygons for spatial containment, and a second pass uses the operational sub-fields produced below.

Operational sub-field construction. The post-LPIS polygons are processed in three stages. First, candidate sub-fields whose row count or row-angle coherence falls below the threshold are dropped or merged into the nearest stronger neighbour within the merge-angle tolerance. Second, an open-area carve identifies regions of each candidate sub-field that lie beyond a buffer distance from any aligned row and removes any such region whose area exceeds the open-area threshold, shrinking the candidate sub-field polygons accordingly. Third, candidate sub-fields whose interior rows resolve into two or more orientation clusters are split: the rows are clustered by orientation with a Gaussian Mixture Model (GMM), the number of clusters chosen by the Bayesian Information Criterion (BIC). When two or more orientation clusters survive the minimum-group and secondary-fraction filters, the candidate sub-field is rasterised at 10 cm, and a watershed transform [31] partitions it using row-buffer seeds. Adjacent same-orientation pieces are then merged when their dominant angles agree within the orientation tolerance and they share a minimum boundary length. The final operational sub-field set, shown in Figure 3, is the one read by the routing planner.

Gaps between watershed pieces are reassigned to the nearest piece, so the split tiles the parent polygon without holes. Figures 4 and 5 show the detected vine rows draped over the DEM surface for *mosel_north* and *mosel_south*.

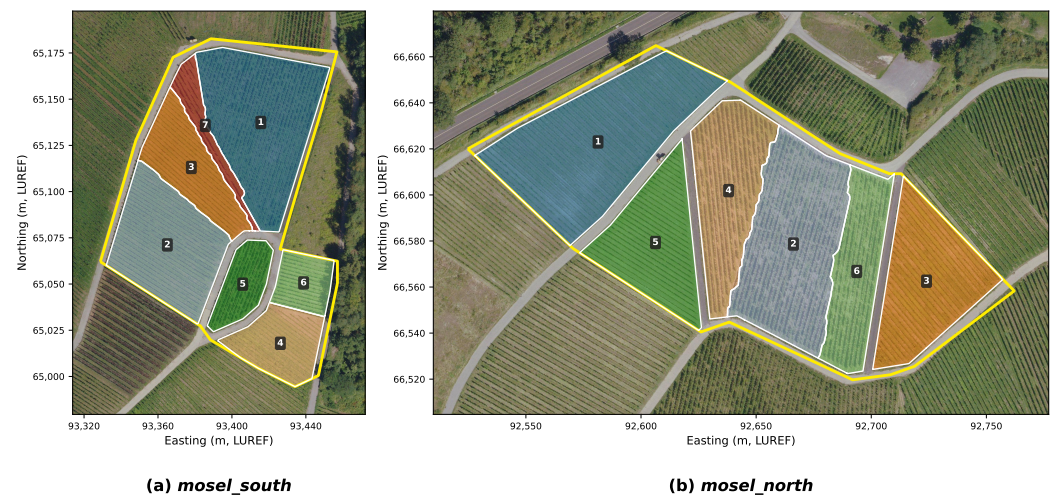


Figure 3. Operational sub-field regions read by the routing planner, overlaid on the orthophoto with the enclosing site boundary in yellow; each colour is one sub-field. *mosel_south* (a) has seven sub-fields, and *mosel_north* (b) has six. The number on each sub-field is its sub-field identifier.

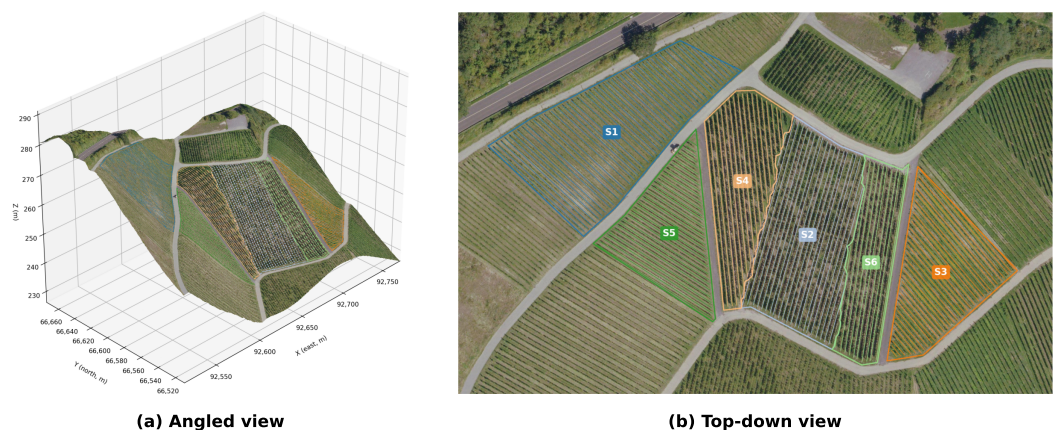


Figure 4. *mosel_north*: Orthophoto and detected vine rows draped over the DEM, including angled (a) and top-down (b) views. S1–S6 mark the six operational sub-fields, numbered as in Figure 3b.

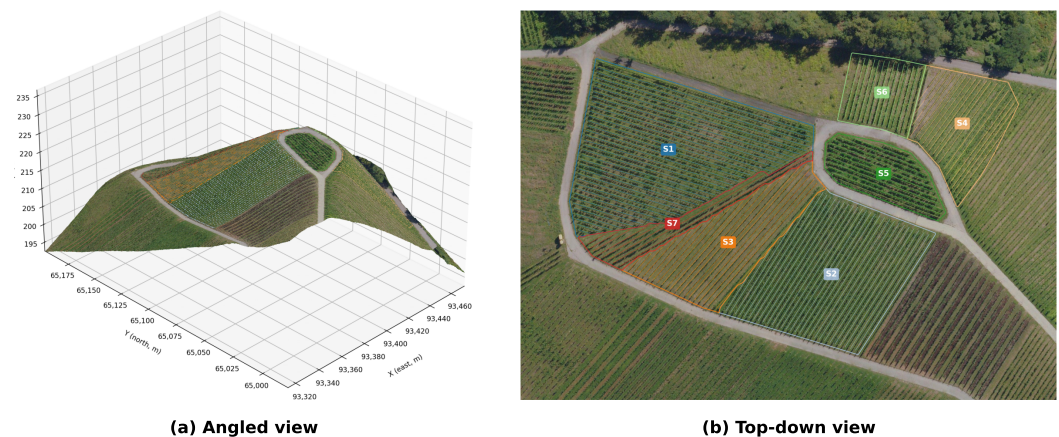


Figure 5. *mosel_south*: Orthophoto and detected vine rows draped over the DEM, including angled (a) and top-down (b) views. S1–S7 mark the seven operational sub-fields, numbered as in Figure 3a.

2.3. Vine-Row Refinement for Routing

The raw DeepLSD rows reflect canopy line segments at native image resolution and frequently include partial detections near sub-field boundaries. The routing planner consumes a refined set built per sub-field by a 1D peak detector on the canopy score. For each sub-field, the script (a) computes a per-pixel vine-canopy score (a Gaussian top-hat filter on the Excess Green index [32]), (b) bins the score by perpendicular distance to the sub-field centroid along the axis perpendicular to the dominant row angle, (c) smooths the resulting 1D profile with a Gaussian kernel, and (d) runs peak detection under prominence and minimum-spacing constraints (the same peak analysis used for anomalous images [33]). Each peak emits one straight row line at the sub-field's dominant angle at the peak's perpendicular distance, clipped to the sub-field polygon. Two boundary-aware filters trim the result: rows whose perpendicular position falls within a small margin of the sub-field's perpendicular extent are dropped, and the endpoints of all remaining rows are drawn inward. The design thresholds for all pipeline stages are collected in Table 1.

Table 1. Pipeline design parameters that drive stages (data fetch → sub-field → row).

Stage	Parameter	Default
Data fetch (raster)	DTM native resolution	50 cm
Data fetch (raster)	Orthophoto GSD	10 cm
Data fetch (LPIS)	HILUCS class filter	vineyard
Sub-field filter	Well-rowed minimum	5 rows
Sub-field filter	Minimum coherence	0.65
Sub-field filter	Open-area minimum	300 m ²
Sub-field filter	Row buffer for clip	1.0 m
Sub-field split	Secondary cluster fraction	0.30
Sub-field split	Minimum piece area	400 m ²
Sub-field merge	Orientation tolerance	5.0°
Sub-field merge	Minimum shared border	4.0 m
Row detection	Minimum row length	3.0 m
Row refinement	Peak prominence fraction	0.10

The detector runs with the Table 1 defaults, and depending on the orthophoto, these thresholds are tuned so the detected sub-fields and rows match the observed vine rows. The tuned output recovers the regular planting geometry, extracting 295 vine rows across the 13 sub-fields of the two sites (Figures 3–5) at a median row spacing of 2.00 m, so the

detected rows track the vineyard's periodic structure. The reliability of this stage depends on canopy visibility in the orthophoto and on the presence of the LPIS vineyard labels.

3. Field-Scale Planner

Given the four data products of Section 2 and a set of N inspection waypoints, the field-scale planner returns a terrain-aware route that visits every waypoint. Figure 1b sketches its structure: it collects the specified waypoints into a single global set, evaluates the $N \times N$ travel-cost matrix ($\mathbf{M}$) on three-dimensional polylines sampled from the DEM, and optimises the waypoint tour under one of the strategies described later in this section.

3.1. Routing-Problem Formulation

Let $\mathcal{S} = \{S_1, \dots, S_K\}$ denote the operational sub-fields produced by the pipeline of Section 2.2, each with a polygon boundary and a discrete grid of inter-row alleys at the sub-field's dominant row angle. Let $z : \mathbb{R}^2 \rightarrow \mathbb{R}$ denote the DEM, that is, the continuous surface obtained by bilinear interpolation of the DTM raster of Section 2.1. Given a set of N inspection waypoints ($\mathcal{W} = \{w_1, \dots, w_N\}$) across the sub-fields, the planner produces an ordered sequence ($\pi = (\pi_1, \dots, \pi_N)$) over $\mathcal{W}$, together with a concatenated polyline ($\Pi \subset \mathbb{R}^3$) that visits every w_{π_k} in order and respects two routing constraints: travel within a sub-field (same-sub-field routing) that follows the network of inter-row alleys and perimeter headlands and travel between sub-fields (cross-sub-field routing) that follows chains of adjacent sub-fields, crossing the service strips between them by short bridges. The planner minimises one of three scalar objectives over π :

$$C(\pi) = \sum_{k=1}^{N-1} \mathbf{M}[\pi_k, \pi_{k+1}] \quad (1)$$

Here, the travel-cost matrix ($\mathbf{M}$) is the distance matrix ($\mathbf{M}_D$), the energy matrix ($\mathbf{M}_E$), or the time matrix ($\mathbf{M}_T$) defined below. The algorithm returns the cheapest route it finds under the objective, and the first and last entries of π define the inspection waypoint at which the mission starts and the inspection waypoint at which it ends.

3.2. Inspection-Waypoint Set-Up

The N inspection waypoints are specified as geographic coordinates (WGS84 latitude/longitude) and reprojected to LUREF (EPSG:2169). Each waypoint is then converted to the planner's internal description: it is located in its host sub-field (S_k) and, in S_k 's local rotated frame, in its host alley (a), whose non-empty, polygon-clipped horizontal range is $[x_{lo}^{k,a}, x_{hi}^{k,a}]$ and whose centre line is y_a^k . This yields the tuple $(k, a, x_{lo}^{k,a}, x_{hi}^{k,a}, y_a^k)$, together with the waypoint's along-alley position (x), clamped to $[x_{lo} + \delta, x_{hi} - \delta]$ with a margin of $\delta = \min(1, 0.1(x_{hi} - x_{lo}))$ m so that it stays clear of the alley endpoint. Each w_i is tagged with its home sub-field and lifted to DEM coordinates by the per-sub-field rotation back into LUREF.

3.3. Travel-Cost Matrix

The $N \times N$ matrix ($\mathbf{M}$) is filled by one ordered waypoint pair (i, j) at a time, in two passes: a first pass builds a three-dimensional polyline between w_i and w_j , and a second assigns it a cost under each objective. The polyline is built in one of two ways, according to whether the two waypoints share a sub-field.

Same-sub-field routing. When both waypoints lie in one sub-field (S_k), the polyline follows the network of inter-row alleys and perimeter headlands: from w_i , the router runs along the host alley to an endpoint, walks the polygon perimeter (the headland) to the nearest endpoint of w_j 's alley, then runs along that alley to w_j . Both endpoints of each alley

are tried, and the cheaper headland walk under the active objective is kept. The polyline is sampled at 0.5 m spacing and lifted to three dimensions by reading the DEM elevation at each sample. Writing $P_S(u, v)$ for the path along the alleys and headland of a sub-field (S) between two of its points (u and v), the same-sub-field polyline is expressed as follows:

$$\Pi_{ij} = P_{S_k}(w_i, w_j) \quad (2)$$

Cross-sub-field routing. When the waypoints lie in different sub-fields (S_a and S_b), the planner routes through a chain of adjacent sub-fields rather than one direct crossing. Adjacency, the relation between two sub-fields whose boundaries are directly connected, defines a graph over the site; on it, the planner takes a shortest chain ($S_a \rightarrow C_1 \rightarrow \dots \rightarrow C_r \rightarrow S_b$) and assembles the polyline as a path along the alleys and headland of S_a from w_i to an exit point ($\mathbf{e}_a$) on the boundary (∂S_a) of S_a , a headland walk around each intermediate sub-field ($C_1, \dots, C_r$) between successive crossing points, short bridges across the service strips between consecutive sub-fields, and a path along the alleys and headland of S_b from an entry point ($\mathbf{e}_b$) on the boundary (∂S_b) of S_b to w_j ; the route follows each intermediate headland. Each hop crosses the shared service strip between two adjacent sub-fields, on which candidate crossing points are sampled (its endpoints, plus the projected source or destination alley endpoints on the first and last hops) and snapped onto both boundaries; the chain, the alley endpoints, and the per-hop crossing points are then optimised jointly by enumerating the candidate combinations and keeping the one with the lowest cost under the active objective on the three-dimensional polyline. Using the following notation, with $\oplus$ for polyline concatenation and P_S for the within-sub-field path of Equation (2), this assembly is expressed as follows:

$$\Pi_{ij} = P_{S_a}(w_i, \mathbf{e}_a) \oplus W(\mathbf{e}_a, \mathbf{e}_b) \oplus P_{S_b}(\mathbf{e}_b, w_j) \quad (3)$$

where $W(\mathbf{e}_a, \mathbf{e}_b)$ is the chain walk from $\mathbf{e}_a$ to $\mathbf{e}_b$, i.e., the intermediate headland walks joined by bridges across the service strips between consecutive sub-fields.

When S_a and S_b share no border, directly or through a chain of adjacent sub-fields, the planner uses a direct boundary-to-boundary crossing through the service strip: a path within S_a from w_i to $\mathbf{e}_a$, a straight line from $\mathbf{e}_a$ to $\mathbf{e}_b$, and a path within S_b from $\mathbf{e}_b$ to w_j . The band of ∂S_a facing S_b is sampled as N_c exit candidates, and the band of ∂S_b facing S_a is sampled as N_c entry candidates, together with two additional candidate crossings: the centroid-line intersection with each boundary and the closest pair of points between the two boundaries. The minimum-cost pair ($\mathbf{e}_a, \mathbf{e}_b$) is retained.

Figure 6 shows an example field-scale route over 10 inspection waypoints on *mosel_north*, in which consecutive waypoints are linked by the same-sub-field and cross-sub-field routing described above. In panel (a), waypoints are numbered by their position in the visit order and coloured by sub-field; path segments inside a sub-field are drawn in that sub-field's colour, cross-sub-field transitions as gold dashed lines, and shared sub-field boundaries in red. Panel (b) profiles the same route along its length: the upper plot shows the DEM elevation, with the route coloured by segment kind (alley or headland), each cross-sub-field crossing marked by a diamond, and the waypoints marked in visit order above the profile; the lower plot shows the cumulative expended energy, which rises steeply on the climb to the high sub-field.

The cost pass then evaluates the polyline under the three objectives, populating the energy matrix ($\mathbf{M}_E$), the time matrix ($\mathbf{M}_T$), and the distance matrix ($\mathbf{M}_D$).

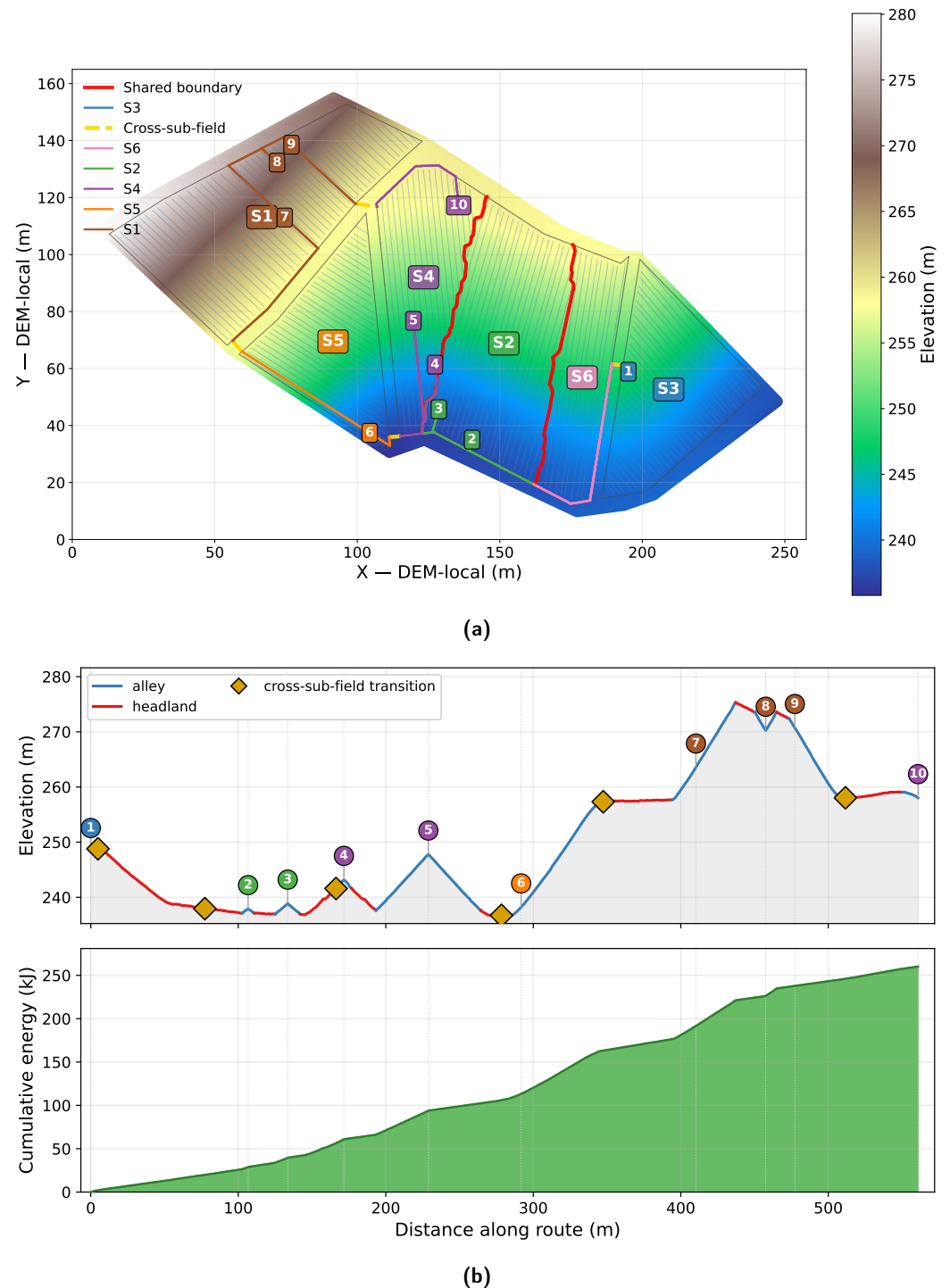


Figure 6. Example field-scale route over 10 inspection waypoints on *mosel_north*: (a) the route at the DEM-shaded site; (b) the elevation and cumulative energy along the route.

Energy. Of the three objectives, energy is the most heavily affected by terrain: on a sloped site, the work done against gravity is a leading cost of motion, yet it is invisible to a planar distance metric. We therefore evaluate an energy model developed in our prior work [24] on the three-dimensional samples of each polyline. The consumed energy is the sum of a translational term, charged by the distance travelled, and a turning term, charged by the angle of each pivot turn:

$$E_{\text{total}} = E_{\text{trans}} + E_{\text{turn}} \quad (4)$$

The translational term (E_{trans}) is integrated over the polyline, which is resampled at 0.5 m horizontal spacing so that each short segment follows the three-dimensional terrain. For one such segment, with horizontal projection (d_{2D}) and elevation change (Δh), the planner computes the three-dimensional length and slope angle:

$$d_{3D} = \sqrt{d_{2D}^2 + \Delta h^2}, \quad \theta = \arctan\left(\frac{|\Delta h|}{d_{2D}}\right) \quad (5)$$

It then decomposes the per-segment translational energy into three terms. The rolling resistance is expressed as follows:

$$E_{\text{rolling}} = \mu m g d_{3D} \cos \theta = \mu m g d_{2D} \quad (6)$$

where μ is the rolling-resistance coefficient, m is the UGV mass, and g is the gravitational acceleration. The climbing work is expressed as follows:

$$E_{\text{climbing}} = \begin{cases} m g \Delta h, & \Delta h > 0 \\ 0, & \Delta h \leq 0 \end{cases} \quad (7)$$

with the climbing work incurred only for a positive elevation change ($\Delta h > 0$), that is, on uphill segments. Defining the load factor as $\ell = 1 + |\Delta h|/d_{3D}$, which equals unity on level ground and rises with slope, the load-dependent motor efficiency is expressed as follows:

$$\eta(\Delta h) = \begin{cases} \max(\eta_{\min}, \eta_0 - \beta(\ell - 1)), & \Delta h > 0 \\ \eta_0, & \Delta h \leq 0 \end{cases} \quad (8)$$

Here, η_0 is the nominal motor efficiency, β is the efficiency derating per unit increase in ℓ , and $\eta_{\min}$ is the efficiency floor. The per-segment translational energy is then expressed as follows:

$$E_{\text{trans}} = \frac{E_{\text{rolling}} + E_{\text{climbing}}}{\eta(\Delta h)} \quad (9)$$

The turning term (E_{turn}) accounts for in-place pivot turns at polyline vertices where the heading change ($|\Delta\phi|$) reaches or exceeds a threshold ($\Delta\phi_{\text{piv}}$). Under a differential-drive pivot, both drive wheels skid sideways across the ground, and the friction torque to be overcome is $\tau = \mu mgL/2$, where L is the wheel separation. Dividing the mechanical work ($\tau |\Delta\phi|$) by the motor efficiency yields the pivot-turn energy:

$$E_{\text{turn}}(\Delta\phi) = \frac{\mu m g L}{2 \eta_0} |\Delta\phi| \quad (10)$$

Heading changes below $\Delta\phi_{\text{piv}}$ are absorbed into continuous-curvature turning at full velocity and incur no additional energy. The connecting polyline (Π_{ij}) of Equations (2) and (3) is resampled into G_{ij} segments of 0.5 m indexed by $s = 1, \dots, G_{ij}$, and Q_{ij} is the number of pivot vertices along it, that is, the vertices where the heading change ($|\Delta\phi_p|$) reaches or exceeds $\Delta\phi_{\text{piv}}$, indexed by $p = 1, \dots, Q_{ij}$. The energy cost between inspection waypoints w_i and w_j accumulates the two components of Equation (4) along Π_{ij} :

$$\mathbf{M}_E[i, j] = \sum_{s=1}^{G_{ij}} E_{\text{trans}}^{(s)} + \sum_{p=1}^{Q_{ij}} E_{\text{turn}}(|\Delta\phi_p|) \quad (11)$$

Figure 7 plots the two terms that build up Equation (11). Panel (a) shows the per-segment translational energy of Equations (6)–(9) over 1 m of travel as a function of slope angle (θ)

for five rolling-resistance coefficients (μ). The curves are flat on the downhill side because the climbing work does not apply when $\Delta h \leq 0$, and they diverge above $\theta = 0$ as μ scales the rolling term while the climbing term adds the uphill-only contribution. This uphill–downhill asymmetry propagates into Equation (11) and makes the travel-cost matrix asymmetric. Panel (b) shows the per-pivot turn energy of Equation (10) against the heading change ($|\Delta\phi|$) for the same five μ values: below the pivot threshold of $\Delta\phi_{\text{piv}} = 60^\circ$, the heading change is absorbed into continuous-curvature turning and pays nothing extra, and at or above the threshold, the cost grows linearly in $|\Delta\phi|$ with a slope set by μ .

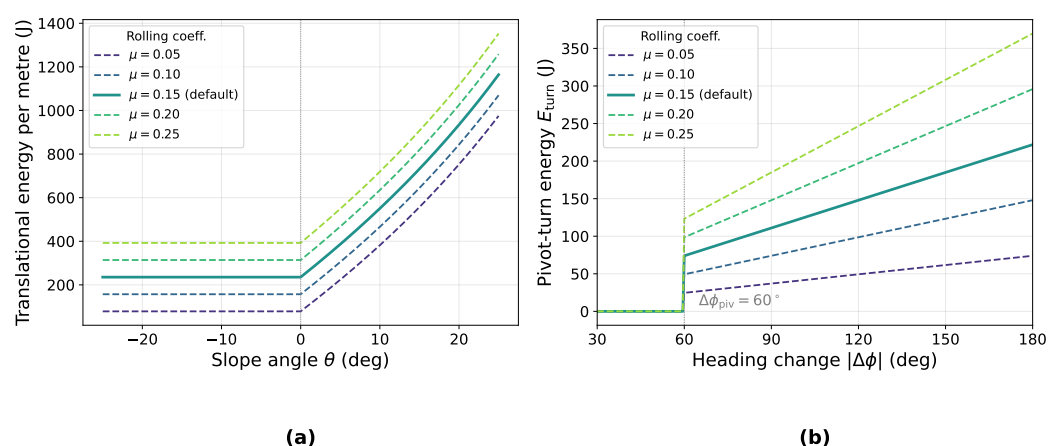


Figure 7. The two components of the energy model: the translational energy (E_{trans}) (a) and the pivot-turn energy (E_{turn}) (b), each plotted for five rolling-resistance coefficients (μ).

Time and distance. Distance and time are the two kinematic objectives: unlike energy, they are independent of travel direction on a slope, so their matrices are symmetric. The time matrix decomposes analogously into translational and rotational components. The translational time of a segment of three-dimensional length d_{3D} at constant nominal UGV velocity (v) is d_{3D}/v , and the rotational time at a pivot vertex with a heading change of $|\Delta\phi| \geq \Delta\phi_{\text{piv}}$ at constant pivot angular velocity (ω_{max}) is $|\Delta\phi|/\omega_{\text{max}}$. The time cell entry is expressed as follows:

$$\mathbf{M}_T[i, j] = \frac{1}{v} \sum_{s=1}^{G_{ij}} d_{3D}^{(s)} + \frac{1}{\omega_{\text{max}}} \sum_{p=1}^{Q_{ij}} |\Delta\phi_p| \quad (12)$$

The distance objective uses the matrix of three-dimensional polyline lengths, which carries no rotational term:

$$\mathbf{M}_D[i, j] = \sum_{s=1}^{G_{ij}} d_{3D}^{(s)} \quad (13)$$

3.4. Tour Optimisation

The travel-cost matrix and the tour form a two-stage decomposition rather than a single monolithic optimisation. Each entry of $\mathbf{M}$ is the lowest cost of routing between its two waypoints under the active objective, so the per-edge crossing selection already minimises the objective that the tour ordering minimises over $\mathbf{M}$; the two stages are coupled through this shared objective rather than solved as decoupled geometric and combinatorial steps. Because each edge is built once and reused, consecutive edges are treated as independent, so an edge's crossing points are not re-coordinated with its neighbours in the tour.

With $\mathbf{M}$ defined, the tour optimiser selects an ordering of the N inspection waypoints using one of four strategies.

The Nearest-neighbour (NN) strategy starts from w_1 and, at each step, selects the unvisited inspection waypoint with the smallest $\mathbf{M}[\pi_{\text{cur}}, \cdot]$ and appends it to π . This is the baseline greedy heuristic and serves as the warm start for the three SA strategies described below.

Simulated Annealing (SA-D, SA-E, and SA-T) includes three variants, all seeded with the nearest-neighbour ordering. At each iteration, the strategy proposes either a random 2-swap (exchange the positions of two inspection waypoints) or a 2-opt reversal (reverse a contiguous sub-route) with equal probability, computes the change in cost ($\Delta = C(\pi') - C(\pi)$) under the active matrix ($\mathbf{M}$), and accepts the candidate with the Metropolis acceptance probability:

$$P_{\text{accept}} = \min(1, \exp(-\Delta/T)) \quad (14)$$

Here, T follows a geometric cooling schedule ($T_{k+1} = \alpha T_k$) over a fixed budget of K_{max} iterations. The initial temperature (T_0) is set to three times the mean travel cost of $\mathbf{M}$, which scales the Metropolis acceptance consistently across the three objectives. The single best ordering observed across all iterations is returned. The matrix is $\mathbf{M}_D$ for SA-D, $\mathbf{M}_E$ for SA-E, and $\mathbf{M}_T$ for SA-T.

The output of the tour optimiser is the ordered sequence (π) and the concatenated polyline (Π), the latter of which is assembled by chaining the per-pair polylines of the travel-cost matrix in the order dictated by π ; the visit order drawn in Figure 6 is one such sequence.

Optimisation structure and complexity. The travel cost of an edge between two waypoints is evaluated on demand when the total cost of a route is computed, and its construction cost depends on the relationship between the two waypoints' sub-fields. Within one sub-field, the edge is a direct in-field path built in $O(P)$, where P is its number of samples, proportional to the path length. Between two sub-fields that share a border, it adds a single crossing selected on that shared border, as well as $O(P)$. Between non-adjacent sub-fields, it follows a chain of adjacent sub-fields and enumerates the combinations of per-hop crossing points: with a bounded number (c) of candidate crossings per hop and a chain of L sub-fields, this is $O(c^{L-1})$ combinations, each $O(P)$, where the chain length (L) is, at most, the number of sub-fields (n_s). This crossing enumeration is exponential in the chain length in the worst case but bounded in practice by the short chains of a real field, and it is independent of the waypoint count (N). Evaluating the total travel cost of one tour sums its $N - 1$ edges on demand at $O(N)$ edge evaluations. Simulated annealing evaluates K_{max} orderings of $N - 1$ edges each, computing every edge cost on demand once and reusing it thereafter, so the annealing search performs $O(K_{\text{max}} N)$ edge evaluations (linear in N). The exponential $O(2^N N^2)$ of an exact TSP solver is thereby avoided. The $O(K_{\text{max}} N)$ bound counts edge evaluations, not edge construction: if the search evaluates the full cost matrix, the number of constructed edges is $O(N^2)$ in the waypoint count (N).

4. Performance Evaluation

4.1. Run-Time Configuration

To evaluate the achievable performance of the proposed routing method, we use the parameters listed in Table 2. The UGV mass, wheel separation, cruise velocity and pivot-turn rate are those of MAXXII, the differential-drive UGV in our laboratory, also consistent with our prior work [24]; the rolling-resistance coefficient and the nominal motor efficiency ($\eta_0 = 0.75$) are representative values (η_0 is typical for geared DC traction

motors). The energy-model derivation that uses these constants is given in Section 3. The evaluation covers two Moselle sites (*mosel_south* and *mosel_north*), five waypoint counts $N \in \{5, 10, 15, 20, 25\}$, and ten different sets of waypoints. For every combination of site, waypoint count (N), and waypoint set, four strategies are run: the nearest-neighbour heuristic (NN) assumed as baseline and three SA TSP variants whose objective is distance (SA-D), energy (SA-E), or time (SA-T), using the corresponding travel-cost matrix ($\mathbf{M}_D$, $\mathbf{M}_E$, or $\mathbf{M}_T$) defined in Section 3. Each waypoint set is drawn at random: for every waypoint, an inter-row alley is picked at random from the global pool across the site's sub-fields, then a random position along it, clear of the endpoints by δ , so the waypoints scatter over the drivable inter-row network. The ten sets per site and waypoint count (N) use ten fixed seeds, so they are reproducible, and the identical set is evaluated by all four strategies.

Table 2. Run-time constants used in all tour-optimisation experiments.

UGV Parameters			Mission Parameters		
Parameter	Symbol	Value	Parameter	Symbol	Value
Mass	m	120 kg	Iteration budget	$K_{\max}$	100,000
Wheel separation	L	0.6 m	Initial temperature	T_0	per mission
Rolling-resistance coeff.	μ	0.15	Cooling factor	α	0.9999
Nominal motor efficiency	η_0	0.75	Site 1		<i>mosel_south</i>
Efficiency derating	β	0.30	Site 2		<i>mosel_north</i>
Efficiency floor	$\eta_{\min}$	0.40	Waypoint count	N	$\{5, 10, 15, 20, 25\}$
Gravitational accel.	g	9.81 m/s ²	Waypoint sets		10
Nominal cruise velocity	v	1.0 m/s			
Pivot angular velocity	$\omega_{\max}$	15°/s			

Over the $K_{\max} = 100,000$ iterations, the geometric schedule ($\alpha = 0.9999$) cools the temperature to 4.5×10^{-5} of its initial value, annealing the search from broad exploration to near-greedy refinement. Every run settles within 0.1% of its final cost, well before the cap, from about 600 iterations at $N = 5$ to around 50,000 at $N = 25$, corresponding to roughly half of $K_{\max}$, after which the tour no longer changes.

4.2. Tour-Optimiser Comparison Across Waypoint Counts

For both Moselle sites, we sweep the global tour-optimisation stage over 10 different sets of waypoints per waypoint count ($N \in \{5, 10, 15, 20, 25\}$), comparing the NN heuristic against three SA TSP variants minimising distance (SA-D), energy (SA-E), and time (SA-T). Table 3 reports the per-set mean of each metric. NN is consistently the worst strategy at every N on every metric. Within the SA family, the objective-matched variant wins its own metric column in every row block. Averaged over the ten sets, the objective-matched variant's largest mean reduction from NN reaches 17.5% in distance (*mosel_north*, $N = 25$), 21.3% in energy (*mosel_north*, $N = 15$), and 16.5% in time (*mosel_north*, $N = 25$) (Figure 8). These per-set means understate the largest gains: across all 100 missions per metric (2 sites $\times$ 5 waypoint counts $\times$ 10 sets), the best single mission improves on NN by 29.7% in distance, 40.4% in energy, and 28.1% in time.

Figure 8 plots, for each metric, the per-set percentage difference between every SA variant and the nearest-neighbour baseline. Rows of the figure correspond to the three metrics (distance/energy/time, top to bottom), and the columns correspond to the two sites, labelled (a) *mosel_south* on the left and (b) *mosel_north* on the right. Each marker is the mean across the 10 tests, and the vertical bars are ± 1 standard deviation; the dashed line at $\Delta = 0$ is the NN baseline, so negative values denote a lower cost than NN. On every metric, the objective-matched variant (SA-D for distance, SA-E for energy, and SA-T for time) attains the largest reduction relative to NN at every N at both sites, and on distance and time, this reduction grows with the waypoint count.

Table 3. Mean tour distance, energy, and time across 10 random waypoint seeds for the two Moselle sites as a function of waypoint count (N) and sequencing algorithm. Bold marks the best (lowest) value in each row block per metric column. NN = nearest-neighbour heuristic; SA-D/SA-E/SA-T = simulated-annealing TSP minimising distance/energy/time, respectively.

Site	N	Distance (m)				Energy (kJ)				Time (s)			
		NN	SA-D	SA-E	SA-T	NN	SA-D	SA-E	SA-T	NN	SA-D	SA-E	SA-T
Mosel south	5	324.5	310.4	310.8	311.0	131.6	119.9	111.8	117.2	392.0	376.7	376.3	373.3
	10	590.6	534.8	538.8	541.4	234.8	212.2	203.8	217.2	760.1	703.1	702.8	695.8
	15	801.4	702.0	707.5	709.8	323.8	284.7	276.5	290.9	1053.8	959.5	963.5	946.3
	20	992.6	861.7	867.7	880.3	399.8	350.8	342.7	362.6	1341.6	1206.8	1209.1	1189.4
	25	1180.4	996.3	1000.9	1015.2	485.8	411.5	401.8	415.2	1627.7	1434.9	1444.8	1401.3
Mosel north	5	341.2	304.6	310.5	305.5	132.5	125.9	101.3	126.2	435.6	385.9	394.5	384.4
	10	614.0	536.4	548.6	541.1	246.2	218.4	197.2	224.1	807.7	716.6	735.5	711.3
	15	834.7	706.3	733.6	713.9	346.1	287.5	272.2	305.9	1125.7	984.2	1019.3	965.3
	20	1046.2	861.9	889.4	880.7	431.1	363.0	341.2	373.3	1430.2	1225.7	1267.2	1206.4
	25	1247.1	1023.7	1053.8	1039.4	522.9	439.0	415.4	457.3	1735.9	1486.2	1521.9	1445.1

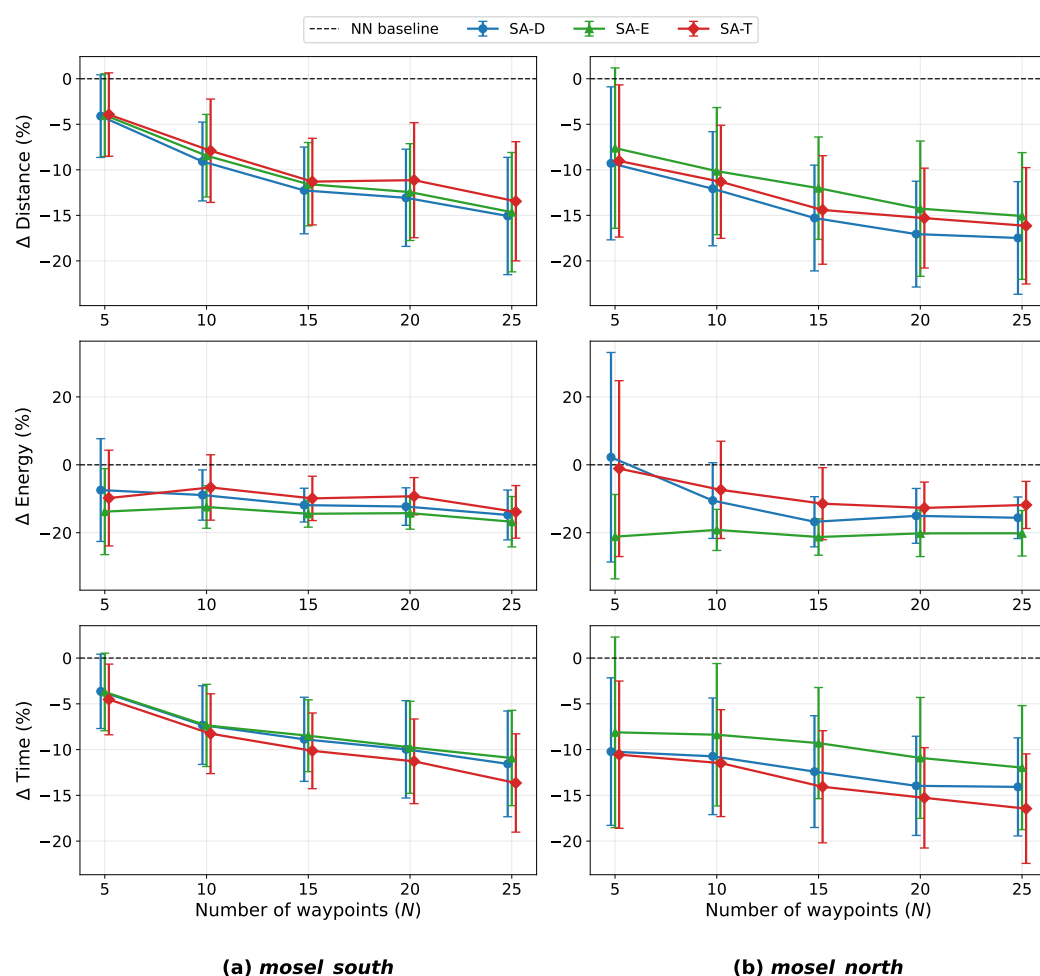


Figure 8. Per-set metric difference between each SA variant and the NN baseline as a function of waypoint count (N) for (a) *mosel_south* and (b) *mosel_north*; negative values indicate a lower cost than NN.

4.3. Benchmark Against Metaheuristic Tour Optimisers

To confirm that the reported energy reductions follow from the terrain-aware cost model rather than from the choice of solver, we benchmark SA against two established metaheuristics named in the literature review—the genetic algorithm (GA) and ant colony optimisation (ACO)—on the energy objective. The three optimisers minimise energy on the same cost matrix, from the same nearest-neighbour warm start, and under the same

evaluation budget. Table 4 shows that they reach nearly the same tour energy of around 1%, on average, and that all three fall far below the nearest-neighbour baseline at every waypoint count. The energy reductions are therefore robust to the choice of metaheuristic, and SA, which attains the lowest energy in most cases, is retained as the default optimiser.

Table 4. Metaheuristic benchmark on the energy objective: mean tour energy (kJ) across ten waypoint seeds for the nearest-neighbour heuristic (NN) and the simulated annealing (SA), genetic algorithm (GA), and ant colony optimisation (ACO) optimisers. Bold marks the lowest value in each row.

Site	N	NN	SA	GA	ACO
Mosel south	5	131.6	111.8	111.8	111.8
	10	234.8	203.8	206.0	204.6
	15	323.8	276.5	279.4	277.6
	20	399.8	342.7	351.2	341.9
	25	485.8	401.8	406.3	403.1
Mosel north	5	132.5	101.3	101.3	101.3
	10	246.2	197.2	201.1	197.2
	15	346.1	272.2	275.2	272.2
	20	431.1	341.2	348.2	346.1
	25	522.9	415.4	422.2	420.4

5. Discussion

Global tour optimisation outperforms greedy routing, and the margin widens with mission size. On both sites and at every waypoint count, the NN heuristic produced the most expensive route on all three metrics (Table 3). Its excess over the best SA variant grew with the waypoint count (N): on *mosel_south*, the NN distance rose from 4.5% above SA-D at $N = 5$ to 18.5% at $N = 25$. Therefore, combining every sub-field's waypoints into a single TSP pays off more as the mission grows, which is the regime in which greedy or per-sub-field routing degrades fastest.

Energy is the highest-payoff objective, and sub-field topology sets the size of the payoff. Tour optimisation under the wrong objective cost little in terms of distance and time (a few percent) but much more in terms of energy, as ordering by distance or time inflated energy substantially, most of all on *mosel_north* (Figure 8b). The cause is the asymmetry of the energy cost: climbing work is charged only on ascents (Equation (7)), and the motor-efficiency penalty applies only to climbs (Equation (8)), so M_E is asymmetric, and the order of ascents and descents changes the total. The SA-E variant exploits this by reordering the route to reduce cumulative climbing. How much climbing it can remove depends on how the sub-fields connect: *mosel_north* is an open, chain-like layout whose traversal order along the hillside dominates the climbing work, so the energy objective gains a large advantage, whereas *mosel_south* is a closed, densely connected layout in which ascents and descents offset across the structure, leaving little climbing to eliminate. Therefore, across the two sites evaluated here, the energy objective gained more on the open, slope-aligned layout than on the closed, densely connected one. This advantage is concentrated at low waypoint counts: on *mosel_north*, the energy penalty for tour optimisation under distance or time falls from 25–28% at $N = 5$ to 6–11% at $N = 25$ (Figure 8b) because the total route energy grows with N faster than the climbing work that SA-E removes.

Choosing the routing objective in practice. The three objectives are not interchangeable, and the appropriate choice depends on the binding operational constraint. Optimising under the wrong objective has little effect on distance and time but a large effect on energy (Figure 8), so energy is the objective that must be selected explicitly to realise its benefit. On a battery-limited UGV working steep, open, slope-aligned terraces, where climbing work dominates the route cost, SA-E is preferred and captures energy reductions that the

distance- and time-optimised tours leave unrealised. When mission completion time is the binding constraint, for example, a time-critical spray treatment or a limited operating-time window, SA-T should be used, since it minimises both travel time and cumulative pivot-turn time. SA-D is preferred when minimising wheel travel is valuable in its own right, for example, to limit tyre wear and soil compaction, or when the energy model's rolling-resistance and motor-efficiency parameters are uncertain, since distance does not depend on those assumptions.

Incorporating a battery budget into the route. Onboard battery capacity is a binding operational limit for a field UGV, and sloped terrain tightens it: the climbing work that dominates the route cost on ascents (Equation (7)) drains the battery fastest exactly where the mission is most demanding, so a single charge need not cover a full multi-sub-field tour. The planner exposes this limit directly because each evaluated entry of the energy matrix ($\mathbf{M}_E$) reports a waypoint-to-waypoint cost in the same units as usable battery capacity. Writing E_{batt} for that capacity and $q = (q_1, \dots, q_m)$ for the ordered waypoints of a trip that departs from and returns to a charging point at q_1 , the trip is battery-feasible when

$$\sum_{k=1}^m \mathbf{M}_E[q_k, q_{k+1}] \leq E_{\text{batt}} \quad q_{m+1} = q_1 \quad (15)$$

When a mission exceeds this budget, the single TSP becomes an energy-constrained, multi-trip Vehicle Routing Problem (VRP): the inspection-waypoint set is partitioned into battery-feasible trips, each bounded by Equation (15) and anchored at a charging or battery-swap station, and the number of trips is minimised jointly with their orderings. The same simulated-annealing search can optimise this variant, since the budget is computed from the same $\mathbf{M}_E$ entries the planner evaluates during routing. Because SA-E minimises cumulative energy, it enlarges the mission reachable on one charge and lowers the number of recharge stops, so the energy objective and the battery constraint reinforce each other.

Enforcing a rollover-stability constraint. Rollover is a safety-critical risk on the steep, slope-aligned terraces this planner targets: a UGV that traverses a grade beyond its static stability angle, the tip-over threshold fixed by its track width and centre-of-mass height, can roll over laterally. The current planner treats stability only implicitly by confining the route to the established alley and headland network and does not bound the slope the route traverses. The constraint can be incorporated into the optimisation from the DEM the planner already samples: each segment ($s = 1, \dots, G_{ij}$) of the connecting polyline (Π_{ij}) carries a local ground slope (σ_s), the steepest inclination of the DEM at that point, so a stability limit (θ_{stab}) masks the travel-cost matrix by pruning any waypoint pair whose connecting polyline exceeds it:

$$\widetilde{\mathbf{M}}[i, j] = \begin{cases} \mathbf{M}[i, j], & \sigma_s \leq \theta_{\text{stab}} \text{ for all } s = 1, \dots, G_{ij} \\ \infty, & \text{otherwise.} \end{cases} \quad (16)$$

Therefore, the tour optimiser never selects a route across terrain steeper than the safe limit. Bounding the local ground slope (σ_s) by θ_{stab} is conservative, since the side slope that drives lateral tip-over never exceeds it. Because unsafe edges are removed before optimisation rather than checked afterward, stability becomes a hard constraint of the routing problem rather than a property left to trajectory-level control. Like the battery budget of Equation (15), this stability mask is left as a future-work extension and is not applied in the experiments of Section 4.

Limitations and future work. The reported time and energy are model predictions rather than measurements: the time model assumes a constant nominal velocity (Table 2), and the energy model uses an assumed rolling-resistance coefficient and motor efficiency

that were not measured on the vehicle. A field campaign recording UGV telemetry along the planned routes would confirm the absolute time and energy and calibrate $\mathbf{M}_E$ and $\mathbf{M}_T$. The router confines the vehicle to the network of inter-row alleys and perimeter headlands and to point-pivot turns, so continuous-curvature tracking is not enforced at the trajectory level. The sub-field extraction and vine-row detection pipeline has not been quantitatively evaluated against manually annotated ground truth; its verification is qualitative, against the provided orthophoto. The reported experiments route a single vehicle under no resource budget; beyond the battery-constrained extension set out above, the most substantial direction is the joint optimisation of a multi-UGV fleet: rather than planning each vehicle independently, the inspection-waypoint set would be partitioned across vehicles, with their routes planned together under a shared fleet objective, for example, minimising the makespan or the total fleet energy, so that load balancing and inter-vehicle coordination are optimised jointly with the per-vehicle routes.

Sensitivity to model assumptions. Because all four strategies are evaluated under the same cost model, each reported reduction is the gap between the NN baseline and an SA tour computed with identical constants, so the percentages are far more stable than the absolute energy they are derived from. The mass (m) and gravitational acceleration (g) enter every entry of $\mathbf{M}_E$ as a common factor (Equations (6)–(10)) and cancel exactly in the energy-reduction ratio. To quantify the remaining parameters, we re-ran the $N = 15$ missions with the NN and SA-E strategies while varying the rolling coefficient (μ) and the nominal efficiency (η_0) (Table 5). The SA-D and SA-T tours are unaffected by this sweep, since neither μ nor η_0 enters $\mathbf{M}_D$ or $\mathbf{M}_T$ (Equations (12) and (13)). The rolling coefficient is the dominant parameter: it multiplies the rolling and pivot-turn terms but not the climbing term, so as μ falls, the climbing work dominates and the energy reduction grows, from 18.5% at $\mu = 0.30$ to 23.4% at $\mu = 0.10$ on *mosel_north*, whereas as μ rises, the energy-optimal tour converges to the distance-optimal one. The nominal efficiency (η_0) scales the translational energy almost uniformly and nearly cancels as well: over $\eta_0 \in [0.60, 0.90]$, the reduction moves by, at most, 0.5 percentage points. Across the full range, the qualitative findings are preserved, i.e., SA-E minimises energy and the reduction is larger on the open *mosel_north* layout (18.5–23.4%) than on the closed *mosel_south* layout (13.4–15.2%); only the absolute magnitude, set mainly by μ , depends on the calibration that the planned field campaign would supply. The constant-velocity assumption affects only the time model, where $\mathbf{M}_T$ reduces to a fixed combination of three-dimensional path length and total turning; a slope-dependent velocity would redistribute the time costs and is the assumption most likely to shift the SA-T results.

Table 5. Sensitivity of the SA-E energy reduction over the NN baseline (per-set mean, %, $N = 15$) to the rolling-resistance coefficient (μ) and the nominal motor efficiency (η_0). The baseline ($\mu = 0.15$, $\eta_0 = 0.75$) reproduces the reported reductions.

Site	Rolling coeff. (μ) at $\eta_0 = 0.75$				Efficiency (η_0) at $\mu = 0.15$	
	0.10	0.15	0.20	0.30	0.60	0.90
<i>mosel_north</i>	23.4	21.3	20.0	18.5	21.6	21.1
<i>mosel_south</i>	15.2	14.4	13.9	13.4	14.5	14.4

6. Conclusions

This paper addressed the routing of a UGV across multiple disjoint vineyard sub-fields, a system in which waypoint ordering, the choice of sub-field-boundary crossing points, and terrain-aware travel cost are coupled and cannot be optimised independently. We presented the field-scale planner, which optimises a tour over the inspection waypoints of all sub-fields with a single TSP whose travel costs are distance, energy, or time evaluated

on three-dimensional polylines sampled from the DEM; reduces an energy-modelling decomposition of rolling resistance, climbing work, and a differential-drive pivot-turn term to a scalar travel cost; and routes cross-sub-field transitions through chains of adjacent sub-fields instead of fixed alley endpoints. The waypoint tour is optimised by one of four strategies—a greedy NN heuristic and three SA variants matched to the distance (SA-D), energy (SA-E), and time (SA-T) objectives, all driven by an integrated pipeline from public Luxembourg geodata to executable routes on two test sites in the Moselle wine region. Across both Luxembourg Moselle sites and every evaluated waypoint count, each objective-matched SA variant attained the lowest cost on its own metric, and the best single mission improved on the NN baseline by up to 29.7% in distance, 40.4% in energy, and 28.1% in time. Across the two evaluated sites, the energy-planning advantage was larger on the open, slope-aligned layout, where the traversal order along the hillside dominates the climbing work, than on the closed, densely connected one, where ascents and descents offset across the structure. Because the reported energy and time are model predictions rather than measurements, the primary direction for future work is a field campaign that calibrates the energy and time models against vehicle telemetry recorded along the planned routes, followed by a joint routing of a multi-UGV fleet under a shared objective so that load balancing is optimised, together with the per-vehicle routes.

Author Contributions: Conceptualization, M.A.A.K.; methodology, M.A.A.K.; software, M.A.A.K.; validation, M.A.A.K.; formal analysis, M.A.A.K.; investigation, M.A.A.K.; data curation, M.A.A.K.; writing—original draft preparation, M.A.A.K.; writing—review and editing, M.R.P.; visualization, M.A.A.K.; supervision, M.R.P.; project administration, M.R.P.; funding acquisition, M.R.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Luxembourg National Research Fund (FNR) under AFR Individual 2023 Call Project No. 18013466/AgriROS.

Data Availability Statement: The inputs for the Moselle test sites (DTM, sub-field polygons, and vine rows) are derived from public geodata: the Luxembourg Land Parcel Identification System (LPIS) and the Luxembourg 2024 LiDAR Digital Terrain Model.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACO	Ant Colony Optimization
BIC	Bayesian Information Criterion
COG	Cloud-Optimized GeoTIFF
CPP	Coverage Path Planning
DEM	Digital Elevation Model
DTM	Digital Terrain Model
GA	Genetic Algorithm
GMM	Gaussian Mixture Model
GSD	Ground Sampling Distance
HILUCS	Hierarchical INSPIRE Land-Use Classification System
LiDAR	Light Detection and Ranging
LKH	Lin–Kernighan–Helsgaun
LPIS	Land Parcel Identification System
LUREF	Luxembourg Reference Frame
NN	Nearest-Neighbour
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PSO	Particle Swarm Optimization
ROS	Robot Operating System

RRT	Rapidly Exploring Random Tree
SA	Simulated Annealing
SAC	Soft Actor–Critic
TSP	Travelling Salesman Problem
UGV	Unmanned Ground Vehicle
VRP	Vehicle Routing Problem
WGS84	World Geodetic System 1984
WMS	Web Map Service

References

1. Gao, Y.; Jiang, Q.; Wang, M.; Dong, X. Advances in Path-Planning Algorithms for Agricultural Robots. *J. Field Robot.* **2026**, *43*, 89–145. [\[CrossRef\]](#)
2. Schöneegg, T.; Tuna, T.; Yang, F.; Waibel, G.; Mattamala, M.; Hutter, M. Global Path Planning for Autonomous Vehicles in Orchards and Vineyards. In Proceedings of the 13th International Workshop on Robot Motion and Control (RoMoCo), Poznan, Poland, 2–4 July 2024; pp. 103–110. [\[CrossRef\]](#)
3. Santos, L.; Santos, F.N.; Mendes, J.; Costa, P.; Lima, J.; Reis, R.; Shinde, P. Path Planning Aware of Robot’s Center of Mass for Steep Slope Vineyards. *Robotica* **2020**, *38*, 684–698. [\[CrossRef\]](#)
4. Hrabar, I.; Vasiljević, G.; Kovačić, Z. Estimation of the Energy Consumption of an All-Terrain Mobile Manipulator for Operations in Steep Vineyards. *Electronics* **2022**, *11*, 217. [\[CrossRef\]](#)
5. Mier, G.; Valente, J.a.; de Bruin, S. Fields2Cover: An open-source coverage path planning library for unmanned agricultural vehicles. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2166–2172. [\[CrossRef\]](#)
6. Santos, L.; Ferraz, N.; dos Santos, F.N.; Mendes, J.; Morais, R.; Costa, P.; Reis, R. Path planning aware of soil compaction for steep slope vineyards. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018; pp. 250–255. [\[CrossRef\]](#)
7. Dogru, S.; Marques, L. ECO-CPP: Energy constrained online coverage path planning. *Robot. Auton. Syst.* **2022**, *157*, 104242. [\[CrossRef\]](#)
8. Pour Arab, D.; Spisser, M.; Essert, C. 3D hybrid path planning for optimized coverage of agricultural fields: A novel approach for wheeled robots. *J. Field Robot.* **2025**, *42*, 455–473. [\[CrossRef\]](#)
9. Pour Arab, D.; Spisser, M.; Essert, C. Complete coverage path planning for wheeled agricultural robots. *J. Field Robot.* **2023**, *40*, 1460–1503. [\[CrossRef\]](#)
10. Mier, G.; Fennema, R.; Valente, J.A.; de Bruin, S. Continuous Curvature Path Planning for Headland Coverage with Agricultural Robots. *J. Field Robot.* **2025**, *42*, 641–656. [\[CrossRef\]](#)
11. Wei, M.; Isler, V. Predicting Energy Consumption of Ground Robots on Uneven Terrains. *IEEE Robot. Autom. Lett.* **2022**, *7*, 594–601. [\[CrossRef\]](#)
12. Shen, M.; Wang, S.; Wang, S.; Su, Y. Simulation Study on Coverage Path Planning of Autonomous Tasks in Hilly Farmland Based on Energy Consumption Model. *Math. Probl. Eng.* **2020**, *2020*, 4535734. [\[CrossRef\]](#)
13. Sharma, M.; Voruganti, H.K. Energy-Efficient Coverage Path Planning for Mobile Robots on Uneven Terrain via the TA-NSGA-II Approach. *Arab. J. Sci. Eng.* **2025**. [\[CrossRef\]](#)
14. Wei, P.; Peng, C.; Lu, W.; Zhu, Y.; Vougioukas, S.; Fei, Z.; Ge, Z. Efficient and Safe Trajectory Planning for Autonomous Agricultural Vehicle Headland Turning in Cluttered Orchard Environments. *IEEE Robot. Autom. Lett.* **2025**, *10*, 2574–2581. [\[CrossRef\]](#)
15. Ye, L.; Li, J.; Li, P. Improving Path Planning for Mobile Robots in Complex Orchard Environments: The Continuous Bidirectional Quick-RRT* Algorithm. *Front. Plant Sci.* **2024**, *15*, 1337638. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Di Massa, G.; Malfi, P.; Pagano, S.; Rocca, E.; Savino, S. Analysis, Modeling, and Simulation of a Rocker–Bogie System Overcoming a Harmonic Bump. *Machines* **2026**, *14*, 103. [\[CrossRef\]](#)
17. Xin, L.; Song, W.; Cao, Z.; Zhang, J. NeuroLKH: Combining Deep Learning Model with Lin-Kernighan-Helsgaun Heuristic for Solving the Traveling Salesman Problem. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Online, 6–14 December 2021.
18. Zheng, J.; He, K.; Zhou, J.; Jin, Y.; Li, C.M. Reinforced Lin-Kernighan-Helsgaun Algorithms for the Traveling Salesman Problems. *Knowl.-Based Syst.* **2023**, *260*, 110144. [\[CrossRef\]](#)
19. Kwon, Y.D.; Choo, J.; Kim, B.; Yoon, I.; Gwon, Y.; Min, S. POMO: Policy Optimization with Multiple Optima for Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020.
20. Kuang, W.; Ho, H.W.; Zhou, Y. CPP-DIP: Multi-objective Coverage Path Planning for MAVs in Dispersed and Irregular Plantations. *Robot. Auton. Syst.* **2026**, *195*, 105193. [\[CrossRef\]](#)

21. Yang, C.; Zheng, D.; Chen, S.; Mansurova, M.; Belgibaev, B.; Zhao, B. AgriPath: A robust multi-objective path planning framework for agricultural robots in dynamic field environments. *Front. Plant Sci.* **2025**, *16*, 1687747. [[CrossRef](#)] [[PubMed](#)]
22. Zheng, Y.; Zhou, J.; Zheng, P.; Sun, J.; Naz, S.T.; Wang, Y. Key-waypoint-guided global coverage path planning system for unmanned ground vehicles in orchards. *Comput. Electron. Agric.* **2026**, *248*, 111779. [[CrossRef](#)]
23. Wu, B.; Ding, Z.; Ostigaard, L.; Huang, J. Reinforcement Learning-Based Energy-Aware Coverage Path Planning for Precision Agriculture. In Proceedings of the International Conference on Research in Adaptive and Convergent Systems (RACS '25), Ho Chi Minh, Vietnam, 16–19 November 2025. [[CrossRef](#)]
24. Alshikh Khalil, M.A.; Palattella, M.R. Terrain-Aware Path Planning for Unmanned Ground Vehicles in Orchard Environments. In *Proceedings of the 6th IFSA Winter Conference on Automation, Robotics and Communications for Industry 4.0/5.0/6.0 (ARCI' 2026)*, Salzburg, Austria, 25–27 February 2026; Yurish, S.Y., Ed.; IFSA: Barcelona, Spain, 2026; pp. 45–49. [[CrossRef](#)]
25. Administration du Cadastre et de la Topographie. Geoportail.lu: The Official Geographic Information Portal of the Grand Duchy of Luxembourg. 2026. Available online: <https://map.geoportail.lu> (accessed on 26 May 2026).
26. Administration du Cadastre et de la Topographie. BD-L-Lidar2024: 3D Survey of the Luxembourg Territory, Digital Terrain Model. Available online: <https://data.public.lu/en/datasets/lidar-2024-releve-3d-du-territoire-luxembourgeois/> (accessed on 26 May 2026).
27. Administration du Cadastre et de la Topographie. Official Orthophoto of the Grand Duchy of Luxembourg, edition 2023 (10 cm GSD). Available online: <https://data.public.lu/en/datasets/orthophoto-officielle-du-grand-duche-de-luxembourg-edition-2023/> (accessed on 26 May 2026).
28. Administration des Services Techniques de l'Agriculture. Land Parcel Identification System (LPIS), Reference Data of Agricultural Parcels, Luxembourg, Edition 2025. Available online: <https://data.public.lu/en/datasets/67af4e7c501c9523e9284226/> (accessed on 26 May 2026).
29. Pautrat, R.; Barath, D.; Larsson, V.; Oswald, M.R.; Pollefeys, M. DeepLSD: Line Segment Detection and Refinement with Deep Image Gradients. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
30. von Gioi, R.G.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: A Fast Line Segment Detector with a False Detection Control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 722–732. [[CrossRef](#)] [[PubMed](#)]
31. Vincent, L.; Soille, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598. [[CrossRef](#)]
32. Woebbecke, D.M.; Meyer, G.E.; Von Bargen, K.; Mortensen, D.A. Color Indices for Weed Identification Under Various Soil, Residue, and Lighting Conditions. *Trans. ASAE* **1995**, *38*, 259–269. [[CrossRef](#)]
33. Nuanmeesri, S.; Poomhiran, L. Dendro-AutoCount Enhanced Using Pith Localization and Peak Analysis Method for Anomalous Images. *Mathematics* **2026**, *14*, 94. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.