# Hybrid Optimization of Horizontal Alignments in European Terrains: A Comparative Study

Ane Espeseth[1(✉)], Martin Juříček[2], Harald Michael Ludwig[3], and Tea Tušar[4,5]

[1] Center for Computing in Science Education, University of Oslo, Oslo, Norway
anekes@uio.no

[2] Faculty of Mechanical Engineering, Brno University of Technology, Brno, Czechia
200543@vutbr.cz

[3] Johannes Kepler University, Linz, Austria
ludwig@csh.ac.at

[4] Department of Intelligent Systems, Jožef Stefan Institute, Ljubljana, Slovenia
tea.tusar@ijs.si

[5] Jožef Stefan Postgraduate School, Ljubljana, Slovenia

**Abstract.** Path planning across terrain is a fundamental challenge in civil engineering, with applications ranging from transportation infrastructure to urban development. Recent advances in computational methods have enabled automated route optimization, particularly in horizontal alignment problems that balance construction costs with terrain constraints. However, standardized comparisons of optimization approaches across diverse geographical contexts remain limited, hindering the development of reliable automated planning systems. Here we show through a systematic comparative study across three European landscapes that A* significantly outperforms RRT* in initial path generation, with better computational efficiency and terrain adaptation, while PSO demonstrates superior optimization capabilities compared to CMA-ES and DE in refining these paths against roadway construction criteria. Through extensive parameter validation, we find these performance advantages remain consistent across different geographical contexts and topographical challenges, with the hybrid A*-PSO approach achieving significantly better results than applying optimization algorithms to straight-line paths alone. These findings provide a comprehensive comparison of key algorithms in infrastructure planning optimization, demonstrating the relative strengths of different approaches in horizontal alignment tasks. This comparative analysis offers practical guidance for algorithm selection while highlighting opportunities for further development through the incorporation of real-world engineering constraints.

**Keywords:** Horizontal alignment optimization · Path planning · Comparative study

A. Espeseth, M. Juíček and H. M. Ludwig—Contributed equally to this work.


P. García-Sánchez et al. (Eds.): EvoApplications 2025, LNCS 15612, pp. 119–136, 2025.
https://doi.org/10.1007/978-3-031-90062-4_8

## 1 Introduction

Horizontal Alignment Optimization (HAO) is a core optimization problem in civil engineering and transportation planning, commonly applied in the design of linear infrastructure such as roadways and railways. It involves finding an optimal path across a horizontal plane, represented by a 2D or 3D map. The goal is to minimize construction costs, factoring in land acquisition, terrain features, environmental constraints, safety, and path curvature, while ensuring compliance with geometric standards such as maximum curvature, path length, and safety requirements [41].

Aligning paths across complex terrains with multiple constraints demands high computational effort, particularly when dealing with non-differentiable cost functions. Early approaches used traditional pathfinding algorithms to generate feasible paths in controlled environments with predictable obstacles. Global optimization techniques were also introduced early on, at first in the form of numerical methods [47] and dynamic programming [46], but soon also through Genetic Algorithms (GAs) in order to address non-linear, large-scale problems in infrastructure planning [18,22]. Over time, a variety of strategies has been developed to improve efficiency, accuracy, and complexity management in HAO. Problem-specific challenges have also been identified and addressed with tailored solutions.

However, this diversity has introduced a challenge: with tailored solutions and evaluations dominating the literature, cross-comparison becomes difficult. As a result, generalization of methods across HAO contexts is limited.

To address the need for standardized comparison, we conduct a structured evaluation using simplified HAO scenarios based on publicly available maps. We employ widely recognized algorithms—A-Star (A*) [16], Rapidly-exploring Random Tree Star (RRT*) [23], Particle Swarm Optimization (PSO) [24], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [15], and Differential Evolution (DE) [43]—without domain-specific customizations to ensure generalizability and facilitate meaningful comparisons across optimization techniques in static horizontal alignment. We compare the algorithms across three structured scenarios: graph-based algorithms, optimization algorithms within a straight corridor, and a hybrid approach that initializes the optimization algorithms with the path created by A* (the best-performing graph-based algorithm). This study thereby provides an initial reference point for evaluating the effectiveness of standard optimization and graph-based algorithms in infrastructure planning.

## 2 Related Work

**The Horizontal Alignment Optimization Task.** There are several variations of HAO tasks, ranging from graph-based path optimization to 2D-corridor selection, parametric tuning of curves, and complex 3D problems combining horizontal and vertical alignment. The objective is often to optimize a single alignment with multiple constraints [22], though recent work emphasizes generating diverse, feasible alternatives to aid infrastructure planners [34].

**Graph-Based Optimization Algorithms for Corridor Selection.** Graph-based least-cost optimization algorithms have long played an essential role in infrastructure planning. They are easily adopted for minimal distance and obstacle avoidance, features of both preliminary alignment and adaptive planning tasks [1,7,38]. Among modern methods, A* and RRT* stand out due to their effectiveness in static and dynamic problems, respectively. A* is effective in predictable environments, while RRT* is adapted for complex, dynamic scenarios such as mobile robot trajectory planning [17,29]. Perhaps due to the similarities between mobile planning and horizontal planning tasks, RRT* and its variations are also widely applied in infrastructure planning [33,44,49].

**Optimization-Only Techniques.** Algorithms such as PSO, CMA-ES, and DE are robust for single-objective optimization tasks in infrastructure planning, including adjusting paths for energy efficiency and crowd safety [53], timetable synchronization [12,27], and construction constraints in challenging terrains [21]. Methods from other disciplines have also been tested. Among the most successful ones is mathematical optimization, particularly Mixed Integer Linear Programming (MILP) and its variants, and the Distance Transform (DT) algorithm, first introduced by [39] in 2006. In later years, methods like Deep Reinforcement Learning [14] have also been used with some success.

In engineering publications, optimization methods are often customized to address the unique needs of specific alignment problems. GAs in particular are frequently modified to handle project-specific requirements, creating intricate genetic representations and operations specific to the problem (see, e.g., [19,26]). These customizations, while effective in specific scenarios, reduce the generalizability of findings and limit direct comparisons between different methods.

**Hybrid Strategies: Pathfinding + Optimization.** In recent years, more general-purpose optimization techniques are finding a place as *refinement tools* for initial paths in static alignment tasks. Bi-level optimization strategies use two different optimization algorithms in sequence, e.g., to find an optimal horizontal corridor to use for vertical optimization [30,50]. While graph-based least-cost algorithms are not ideal for complex objective functions, they can efficiently find promising paths which then drastically reduce the search space of more sophisticated optimization algorithms. For instance, [49] uses RRT* to initialize a DE optimization stage. In [40], a four-step hybrid approach is used which mixes Dijkstra with PSO to optimize for a complex railway alignment problem. Another paper, [32], initializes with DT and optimizes with PSO. In [45], a variant of RRT* is used alongside Ant Colony Optimization. Hybrid methods like these are gradually gaining popularity, but a 2023 review of HAO [41] still identified their exploration as a gap in the current literature.

**Solution Encoding.** In infrastructure optimization, physical constraints like path curvature are critical. While dynamic path planning often incorporates curves directly into the solution representation (see, e.g., [6,52]), HAO typically uses lightweight representations. Solutions are stored as point lists, which are transformed into smooth paths at evaluation [41] (for a comprehensive review of these methods, see [36]).

**Comparison Studies.** Despite much modern progress in optimization and pathfinding, comprehensive studies that evaluate and compare these methods in standardized static alignment contexts remain limited: in [2], a comparison was made between bi-objective optimization techniques for vertical alignment, including three scalarization methods and two GAs, and [48] compares the use of GAs, PSO and Nonlinear Optimization with Mesh Adaptive Direct Search (NOMAD) for initializing the Sequential Quadratic Programming (SQP) optimizer on a 3D problem. Research is shifting towards hybrid and multi-objective approaches that better use the strong points of each optimization technique, but consistent comparisons are still needed to make these advancements applicable across alignment problems.

# 3 Methods

## 3.1 Path Planning Algorithms

**A*.** The A* algorithm [16] is a classic and widely used algorithm for pathfinding and graph traversal that combines the properties of Dijkstra's and Uniform-Cost-Search [37]. As a heuristic search algorithm, it dynamically expands toward the goal under the guidance of a heuristic function, continually seeking the most efficient path between the start and end point. A* and its variations are particularly applicable in fields such as mobile robot navigation [5,9,10] and computer games [3,25,51].

The algorithm uses an evaluation function, $f(n)$, defined as:

$$f(n) = g(n) + h(n),$$

where $f(n)$ represents the total cost estimate for node $n$, $g(n)$ is the actual cost of the path from the start to $n$, and $h(n)$ is the heuristic estimate of the cost from $n$ to the target. The heuristic must be admissible: $h(n) \leq h^*(n)$, where $h^*(n)$ is the actual optimal path cost from node $n$ to the goal. Common heuristics include Manhattan distance, Euclidean distance, and diagonal distance. For each current node $v_n$, the cost $g(v'_n)$ for neighboring nodes $v'_n$ is calculated using the formula:

$$g(v'_n) = g(v_n) + c(v'_n),$$

where $c(v'_n)$ is the distance between the nodes. The algorithm systematically explores nodes by selecting the most promising candidate from the open list, then moves it to the closed list as part of the path. It expands neighboring nodes and updates the candidate list until the target node is added to the closed list, at which point a complete path has been found.

Adjusted slightly for HAO, the implementation accounts for elevation differences using an extended cost function, allows for diagonal movement, and operates with an extended neighborhood.

**RRT*.** RRT*, a cousin of A*, also incrementally builds a tree in a defined environment [23]. Its key feature, which adapts it to dynamic or uncertain environments, is its ability to continuously reevaluate and improve paths. If a new branch offers a shorter path to the goal, the tree is "rewired" to best incorporate it, increasing the likelihood of finding an optimal trajectory. RRT* can be easily extended with various heuristics and constraints, and has a prominent presence in the fields of robotics, such as motion planning for mobile robots [35], drones [11] and manipulators [28]. It guarantees asymptotic optimality, improving path quality as computation time increases.

In each iteration, the algorithm randomly samples a point $x_{\text{rand}}$ from the configuration space and finds the nearest existing node $x_{\text{nearest}}$ using a distance metric $d(x_1, x_2)$. It then attempts to connect a new point $x_{\text{new}}$ at a step size $\delta$ in the direction of $x_{\text{rand}}$. The best parent for $x_{\text{new}}$ is identified within a neighborhood radius, ensuring the path improves progressively. The path cost $c(x)$ is calculated as:

$$c(x) = c(x_{\text{parent}}) + d(x_{\text{parent}}, x).$$

After adding $x_{\text{new}}$, the algorithm attempts to rewire existing nearby nodes to shorten the overall path, ensuring that the tree converges toward the optimal path as $n \to \infty$. The condition for rewiring can be expressed as:

$$c(x_{\text{new}}) + d(x_{\text{new}}, x_{\text{near}}) < c(x_{\text{near}}).$$

This enables RRT* to improve paths iteratively, making it highly effective for complex path-planning tasks in high-dimensional spaces.

### 3.2 Optimization Algorithms

**PSO.** In Particle Swarm Optimization, a population of particles iteratively searches for the optimal solution by adjusting their positions within a multi-dimensional search space based on individual experiences and the collective knowledge of the swarm [13,20,24]. The key mechanisms of the PSO algorithm are described in two equations. The position $\mathbf{x}_i^{(t)}$, of particle $i$ at iteration $t$ is updated as:

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \mathbf{v}_i^{(t+1)},$$

where $\mathbf{v}_i^{(t+1)}$ is the particle's velocity at the next iteration. The velocity is updated based on four factors: the particle's current position $\mathbf{x}_i^{(t)}$, its previous velocity $\mathbf{v}_i^{(t)}$, its previous best position $\mathbf{p}_i^{(t)}$ and the global best position found by the swarm $\mathbf{g}^{(t)}$:

$$\mathbf{v}_i^{(t+1)} = w \cdot \mathbf{v}_i^{(t)} + c_1 \cdot r_1 \cdot (\mathbf{p}_i^{(t)} - \mathbf{x}_i^{(t)}) + c_2 \cdot r_2 \cdot (\mathbf{g}^{(t)} - \mathbf{x}_i^{(t)}),$$

here $w$ is the inertia weight, $c_1$ and $c_2$ are acceleration coefficients, and $r_1$ and $r_2$ are random numbers drawn from a uniform distribution.

**CMA-ES.** The Covariance Matrix Adaptation Evolution Strategy is an evolutionary optimization algorithm that generates new solutions by sampling from an evolving probability distribution [15]. A key feature of CMA-ES is its ability to learn correlations between variables, enabling efficient search even for complex and non-separable functions. CMA-ES adapts two components during optimization: the covariance matrix and the global step size. The covariance matrix is updated based on evolutionary paths and the differences between the best individuals of consecutive generations. The global step size is adjusted by monitoring the length of consecutive steps: it increases when steps are larger than expected and decreases if the steps are smaller. The algorithm begins with initialization, followed by population generation and fitness evaluation. After sorting and selecting the best individuals, the covariance matrix and step size are updated, and the process is repeated until a stopping criterion is met.

**DE.** Differential Evolution is an iterative algorithm that creates new candidate solutions by blending information from randomly selected individuals within the current population [43]. Candidates are then compared to existing solutions, and if they show better performance, they replace them. Its key operations are *mutation*, *crossover* and *selection*.

*Mutation (differential variant):* At each iteration $t$, new solutions $\mathbf{v}$ are created by combining randomly selected individuals $\mathbf{x}$ from the population:

$$\mathbf{v}_i^{(t+1)} = \mathbf{x}_{r_1}^{(t)} + F \cdot (\mathbf{x}_{r_2}^{(t)} - \mathbf{x}_{r_3}^{(t)}),$$

where $i = 1, 2, \ldots, N$, and $r_1, r_2$, and $r_3$ are distinct random indices. Here, $F$ is the mutation factor that scales the differential solution.

*Crossover (exponential variant):* Trial solutions $\mathbf{u}_i^{(t+1)} = (u_{i,1}^{(t+1)}, \ldots, u_{i,D}^{(t+1)})$ are created by mixing elements from $\mathbf{v}$ and $\mathbf{x}$. For each element $j$ in the solution:

$$u_{i,j}^{(t+1)} = \begin{cases} v_{i,j}^{(t+1)} & \text{if } \mathrm{rand} \leq CR \text{ or } j = j_{\mathrm{rand}}, \\ x_{i,j}^{(t)} & \text{otherwise,} \end{cases}$$

where $CR$ is the crossover constant, rand is a random value between 0 and 1, and $j_{\mathrm{rand}}$ is a random index, ensuring at least one element from $V$ is included.

*Selection:* The better solution between the trial solution and the current solution is retained, ensuring elitism where $f(\cdot)$ is the fitness function:

$$\mathbf{x}_i^{(t+1)} = \begin{cases} \mathbf{u}_i^{(t+1)} & \text{if } f(\mathbf{u}_i^{(t+1)}) < f(\mathbf{x}_i^{(t)}), \\ \mathbf{x}_i^{(t)} & \text{otherwise,} \end{cases}$$

where $f(\cdot)$ is the fitness function.

### 3.3 Experiment Design

The Python code used in this work is available on Github[1].

**Geographical Information Model.** To base this study on real-world data, we used digital elevation maps from the OpenDEM project, produced using Copernicus data and information funded by the European Union – EU-DEM layers [31]. Our experiments focused on regions in Slovenia, Austria, and Italy, selected for their diverse topographical features[2]. In Slovenia, we examined the area between Deskle and Ljubljana; in Austria, the section between Lienz and Villach; and in Italy, the path between Terni and L'Aquila. The elevation data for these regions was downloaded in GeoTIFF format from OpenDEM, then normalized and converted into 2D array format for analysis.

**Path Planning Comparison.** In the first stage, we compared the A* and RRT* algorithms for finding paths across all maps. The goal was to identify a foundational planning algorithm for subsequent path optimization. The comparison of these algorithms is then made in the context of path length (Euclidean distance), time computation and elevation differences along the path. The RRT* algorithm typically works with constraints or collision objects that it must avoid, and in our case, we introduced an altitude constraint set to 800 m to improve computational efficiency by limiting the search space. For our *baseline* comparison we connected the start and goal points with a line segment, which is then optimized in the next phase based on the objective function.

**Path Optimization.** In the second stage, paths produced by the selected path planning algorithm and the baseline approach are further refined because they initially fail to meet all horizontal alignment constraints—such as a clothoidal shape and minimum turning radius. To address these limitations, optimization algorithms like PSO, CMA-ES, and DE are applied to this constrained optimization problem. The steps of this stage are illustrated in Fig. 1.

Starting with an initial path $P_{\text{init}}$, we first simplify it through downsampling with the Douglas-Peucker method[3], yielding a more manageable path $P$ with $n$ points. The downsampling precision is controlled by the parameter $\varepsilon$. Since both $P_{\text{init}}$ and $P$ follow the map grid and are rugged, we further smooth $P$ using splines of clothoid curves, with their smoothness controlled by the parameter $\tau$. The clothoid curves are defined by control points, which can be placed on perpendicular cutting planes along the path $P$. The length of cutting planes is determined by the Cutting Plane Factor (CPF) parameter.

---

1 https://github.com/Steigner/HorizAligns-Hybrid-Optimization.

2 The specific maps were **Slovenia**: N255E460, N255E465; **Austria**: N260E450, N260E455, N260E460; **Italy**: N210E455, N210E460, N215E460, N215E455, N215E450, N210E450.

3 The Douglas-Peucker method is a widely used algorithm for simplifying curves, polygons or paths by eliminating points that minimally affect the overall shape [8].

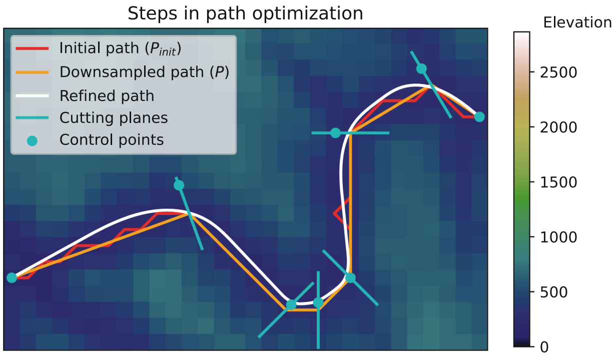


**Fig. 1.** The path optimization process consists of two steps: 1) The initial path $P_{\text{init}}$ is simplified to generate a downsampled path $P$, which establishes cutting planes—line segments designated for the placement of clothoid control points. 2) Optimization algorithms search for the optimal positioning of these control points along the cutting planes, yielding an optimized path trajectory that satisfies the given constraint.

Optimizing the path trajectory means finding the positions of clothoid control points that minimize the objective function:

$$\sum_{i=1}^{n-1} \|P_{i+1} - P_i\| \cdot (1 + w_g + w_t),$$

where $\|P_{i+1} - P_i\|$ represents the Euclidean distance between consecutive points in the path $P = \{P_1, P_2, \ldots, P_n\}$. The weights $w_g$ and $w_t$ penalize steep terrain gradients, the mitigation of which would demand additional construction costs, and expensive digging of tunnels, respectively:

$$w_g = \begin{cases} 2, & \text{if } G > 8\,\% \text{ or } G < -8\,\% \\ 0, & \text{otherwise} \end{cases}, \quad w_t = \begin{cases} 5, & \text{if } T > 800\,\text{m} \\ 0, & \text{otherwise} \end{cases}.$$

The constants used in these formulations were determined by a domain expert. Additionally, to adhere to safety constraints, the solution requires that the turning radius $R > 100\,\text{m}$ along the entire path. We verify this using the differential curvature method [42].

## 4 Results

**Graph-Based Algorithms.** Our investigation revealed a striking hierarchy of efficiency among path-planning approaches, with implications for real-world infrastructure optimization. The comparison between A* and RRT* algorithms proved particularly illuminating: A* demonstrated remarkable computational efficiency, executing approximately 78 times faster than RRT* while simultaneously discovering paths that more naturally conformed to terrain features

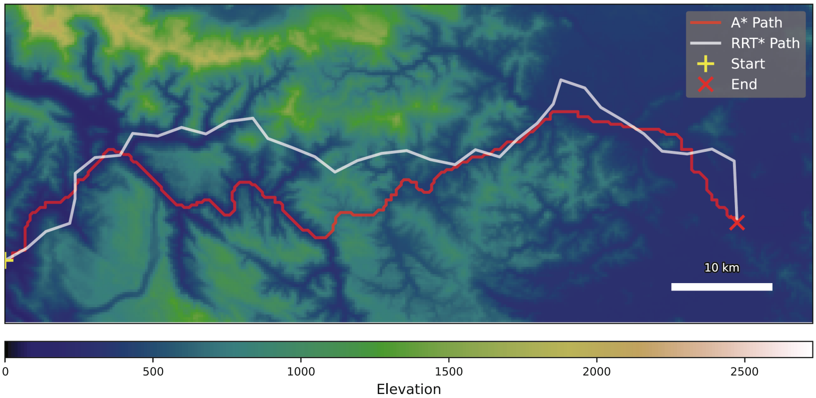


**Fig. 2.** Comparison of paths generated by A* and RRT* algorithms on the Slovenian map from Deskle to Ljubljana city center, overlaid on elevation data (meters). The A* path (red) demonstrates more direct routing compared to the RRT* path (white). (Color figure online)

(Fig. 2). This dramatic performance difference stems from A*'s ability to leverage terrain-aware heuristics better than RRT*, enabling it to prioritize paths through valleys and avoid unnecessary elevation changes. Furthermore, A* exhibits favorable computational scaling with terrain resolution. Our experiments show that, on average, a 10-fold increase in resolution results in only a 6.2-fold increase in computation time, from 0.8 s to 5 s, demonstrating efficient performance.

**Hyperparameter Search.** This initial finding guided our hybrid optimization strategy. Rather than pursuing parallel development of both algorithms, we leveraged A*'s superior performance as a foundation for a deeper exploration of path refinement. Using Weights & Biases' Bayesian optimization framework [4], we conducted an extensive hyperparameter search across 128 preliminary runs on the Slovenian map, systematically exploring the parameter space for three distinct optimization algorithms: PSO, CMA-ES, and DE.

To validate whether these parameters, optimized for the Slovenian terrain, would generalize to different geographical contexts, we performed a comprehensive grid search comprising 246 runs across three distinct geographical regions (Slovenia, Austria, and Italy), three algorithms, three random seeds, and multiple parameter combinations (Table 1). Notably, the parameter values that performed well in Slovenia proved robust across all three terrains, suggesting that our optimization approach captures fundamental aspects of the path-planning problem rather than terrain-specific features. The results of this exploration were unequivocal: PSO emerged as the consistently superior approach across all tested scenarios, maintaining its performance advantage even in substantially different geographical contexts.

**Table 1.** Parameter ranges explored during Bayesian optimization using A*-initialized paths.

| Parameter | Sweep Range | Most Promising Values | Selected Values |
|---|---|---|---|
| Cutting Plane Factor | [1, 5] | 1, 3, 5 | 1 |
| $\varepsilon$ | [0.0, 1.0] | 0.6, 0.8, 1.0 | 1.0 |
| $\tau$ | [0.1, 1.0] | 0.2, 0.4, 0.75 | 0.4 |
| Population Size | [20, 100] | 60 | 60 |
| Generation Size | [5, 50] | 50 | 50 |

**Table 2.** Fitness mean for the best CPF and across CPFs for the three stand-alone optimization algorithms on each map. Apart from *, which used CPF 3, all top runs used CPF 1.

| Map | Best CFP ($n = 3$) | | | All CFPs ($n = 9$) | | |
|---|---|---|---|---|---|---|
| | PSO | CMA-ES | DE | PSO | CMA-ES | DE |
| Slovenia | 487 | 490 | **482** | 501 | 593 | 604 |
| Italy | **883** | 895 | 896 | 984 | 1012 | 1121 |
| Austria | 1365* | **1190*** | 1443 | 1428 | 1419 | 1581 |
| Overall | 912 | **858** | 940 | 971 | 1008 | 1102 |

**Optimization on a Straight Corridor.** We next created a baseline for the three algorithms PSO, CMA-ES, and DE by optimizing over the straight path from *start* to *goal*. To enable a fair comparison with the hybrid strategy, we used the hyperparameters identified in the previous section, but allowed a re-tuning of the CPF, as the optimal path needed to deviate quite far from the straight line. A grid search over CPF values 1, 3 and 5 nevertheless favoured CPF 1, i.e., the paths tightly following the original line, as can be seen in Table 2. The overall best-performing algorithm was PSO, but the top algorithm differed for each map, showing how the HAO task can have strong sensitivity to the choice of algorithm when it operates on its own. Figure 3 demonstrates this descrepancy between the best paths for each CPF on the Austrian map.

**Hybrid Optimization Results.** Next, we compared the performance of PSO, CMA-ES and DE when optimizing the A* path. The optimization convergence plots (Fig. 4) reveal several key insights. First, PSO consistently outperformed both CMA-ES and DE across all geographical contexts, achieving better solutions with fewer generations. This superiority manifested not just in final fitness values but also in the reliability of convergence, as evidenced by the tighter interquartile ranges in the PSO trials. Second, the relative difficulty of optimization varied significantly across regions, with the Slovenian terrain proving most amenable to optimization while the Italian landscape presented the greatest challenge. This variation appears to be driven by topographical constraints:

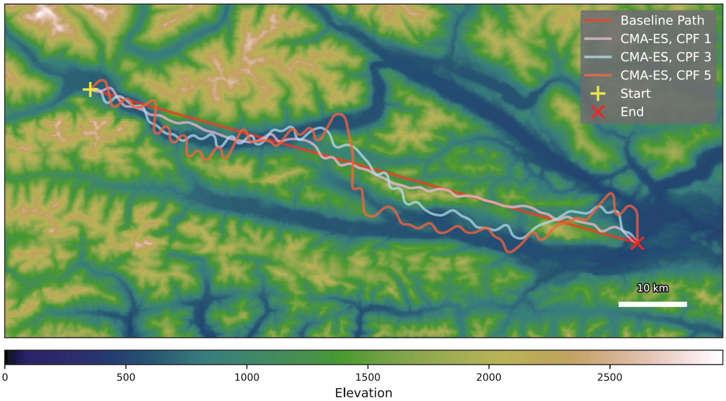


**Fig. 3.** Comparison of CMA-ES paths optimized with three different CPFs on the Austrian map. As CPF increases, curves become more extreme, leading to lower fitness. For the Austrian map, CPF 3 is just large enough to benefit from the nearby valleys, and outperforms CPF 1.

the Italian test case includes a path where approximately half the optimal route lies in a very narrow corridor of low altitude. In this region, even slight deviations from the optimal path result in substantial fitness penalties, leading to a more challenging optimization landscape. Notably, in these topographically constrained sections, the optimized path closely follows the initial A* solution, suggesting that the initial path-planning phase had already identified near-optimal routing through these challenging areas. Third, all algorithms showed rapid initial improvement followed by diminishing returns, but PSO maintained progress longer than its competitors. When compared to the baseline results, the hybrid optimization consistently outperforms the baseline optimization, as demonstrated in Table 3.

**Implications.** The practical implications of these findings are visualized in Fig. 5, where we can observe how the optimized paths deviate from their A*-initialized predecessors. The optimized routes demonstrate smoother transitions and better adaptation to terrain features while maintaining feasible construction constraints. This improvement is particularly evident in areas where the original A* path made sharp turns or traversed challenging elevation changes. Since the optimization process adjusts the cutting planes, additional turns can be introduced along the optimized path.

Our results suggest that a two-phase approach—initial path planning with A* followed by PSO-based refinement—represents a robust strategy for real-world infrastructure planning. This combination effectively balances computational efficiency with solution quality, providing a practical framework for addressing complex routing challenges in varied geographical contexts.

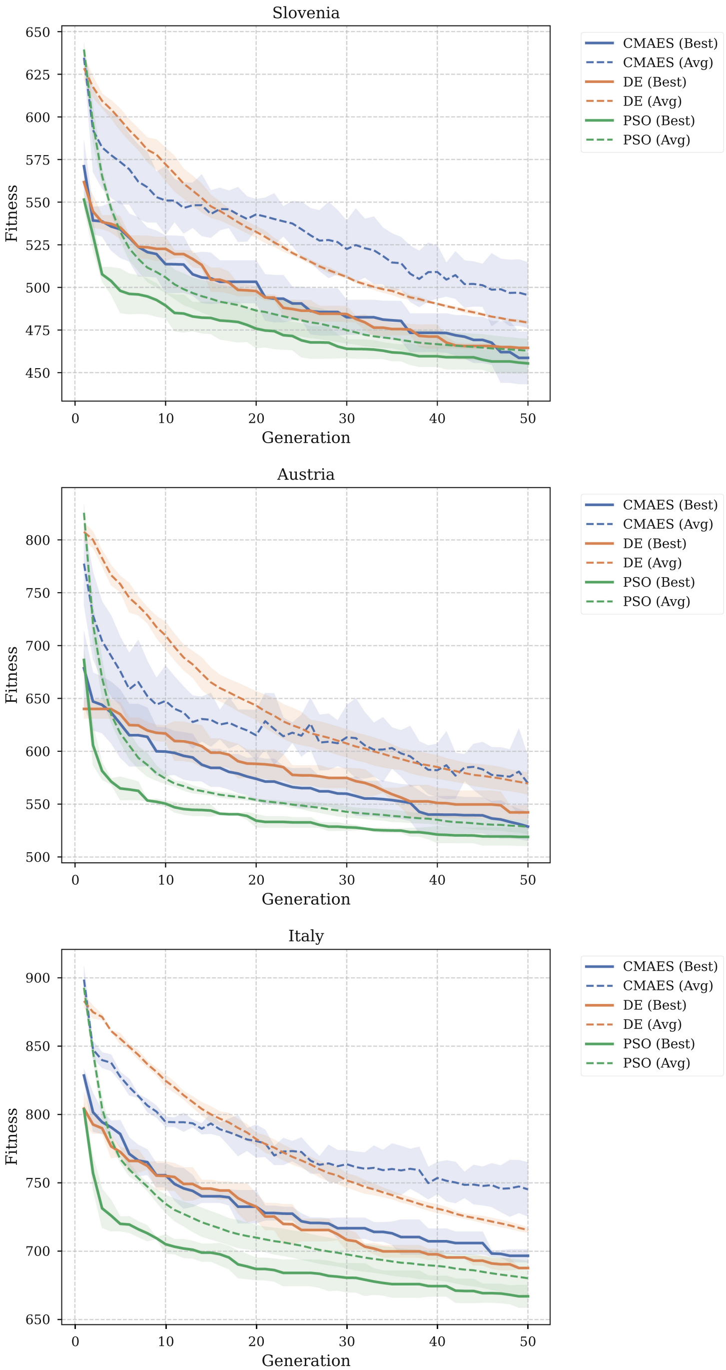


**Fig. 4.** Fitness progressions for optimization algorithms across different geographical maps. Solid lines represent best fitness values per generation, while dashed lines show average fitness. Shaded areas indicate the inter-quartile range across three independent runs.

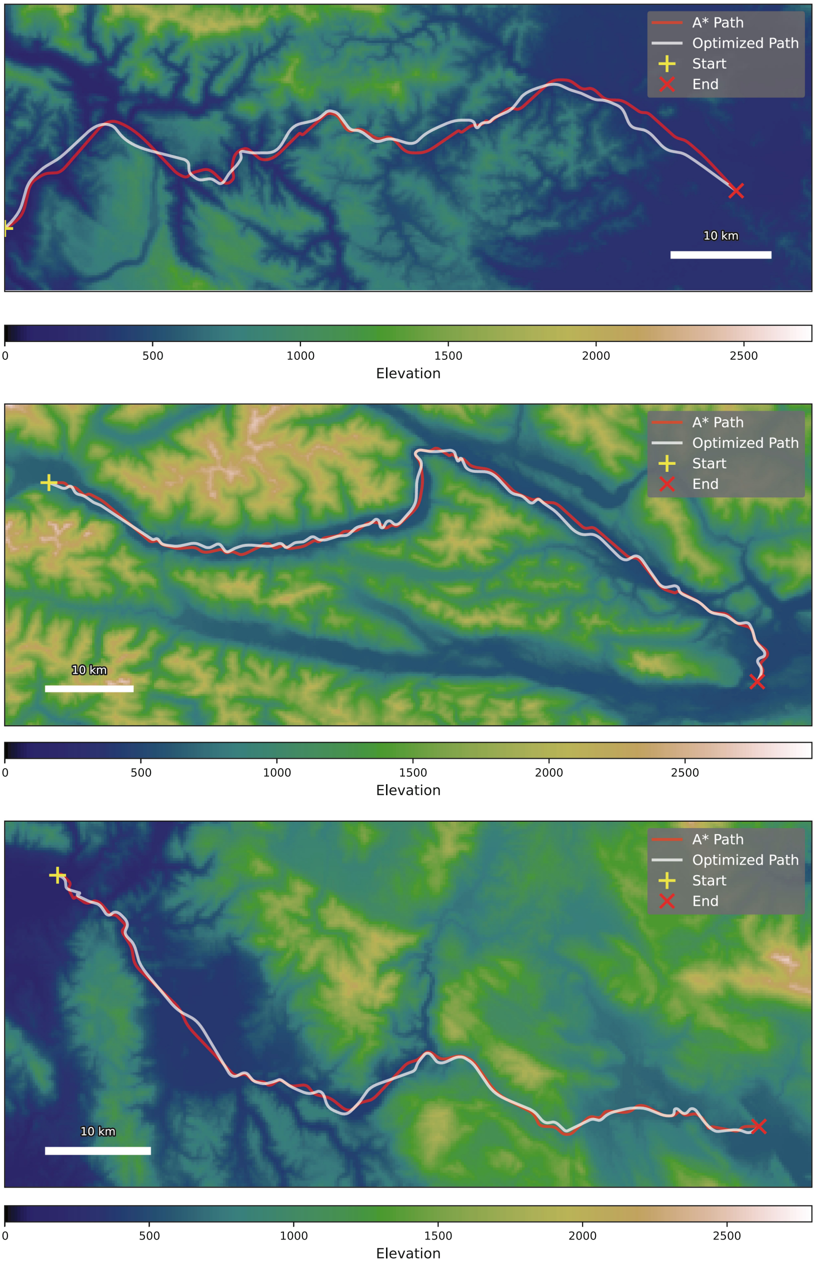


**Fig. 5.** Comparison of initial A* paths (red) and final PSO optimized paths (white) across three different geographical regions, showing how optimization improves path smoothness and terrain adaptation while maintaining construction constraints. (Color figure online)

**Table 3.** Comparison of the final mean fitness values from optimization on a A* path versus on the straight corridor. T-tests for the overall scores of the two approaches showed them to be highly significant ($p \lll 0.01$ for distinct maps, $p < 0.05$ overall).

| Map | Algorithm | Hybrid | Baseline |
|---|---|---|---|
| Slovenia | PSO | **455** ± 5.9 | 487 ± 13.0 |
| | CMA-ES | **459** ± 15.3 | 491 ± 4.8 |
| | DE | **465** ± 0.5 | 482 ± 7.5 |
| | Overall | **460** ± 9.1 | 487 ± 8.8 |
| Austria | PSO | **519** ± 8.5 | 1365 ± 49.7 |
| | CMA-ES | **529** ± 13.6 | 1190 ± 105 |
| | DE | **542** ± 6.6 | 1443 ± 6.8 |
| | Overall | **530** ± 13.2 | 1351 ± 150 |
| Italy | PSO | **667** ± 8.2 | 884 ± 5.7 |
| | CMA-ES | **697** ± 4.7 | 895 ± 11.3 |
| | DE | **688** ± 0.8 | 896 ± 10.5 |
| | Overall | **684** ± 14.0 | 892 ± 10.2 |
| Overall | | **558** ± 96 | 910 ± 369 |

## 5 Conclusions

Our comparison of path-planning and optimization algorithms for horizontal alignment has yielded significant insights for infrastructure planning. The clear superiority of A* over RRT* for initial route planning suggests that graph-based approaches may be more suitable for static environments. Furthermore, our systematic evaluation across diverse European terrains revealed that PSO consistently outperforms both CMA-ES and DE, with parameter settings that proved robust across different geographical contexts.

However, important limitations must be acknowledged. Our study focused on relatively small geographical regions and simplified cost functions compared to real infrastructure projects. The computational requirements of our approach may increase significantly for larger-scale projects or when incorporating more complex terrain features and constructions constraints.

Despite these limitations, our two-phase approach—combining A* for initial path generation with PSO for refinement—offers a promising direction for practical infrastructure planning, effectively balancing computational efficiency with solution quality. Future work could explore the method's scalability to larger geographical areas, incorporate multi-objective optimization for competing priorities such as environmental impact and social factors, and explore more diverse landscapes.

**Acknowledgments.** Special thanks to the Species Society and the organizers of the Species Summer School 2024 for connecting us thus making this research possible.

Tea Tušar acknowledges financial support from the Slovenian Research and Innovation Agency (research core funding No. P2-0209, and project No. N2-0254 "Constrained Multiobjective Optimization Based on Problem Landscape Analysis"). Martin Jurícek was supported by the project GACR No. 24-12474S "Benchmarking derivative-free global optimization methods".

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Aguiar, M.O., et al.: Optimizing forest road planning in a sustainable forest management area in the Brazilian Amazon. J. Environ. Manage. **288**, 112332 (2021). https://doi.org/10.1016/j.jenvman.2021.112332
2. Akhmet, A., Hare, W., Lucet, Y.: Bi-objective optimization for road vertical alignment design. Comput. Oper. Res. **143**, 105764 (2022). https://doi.org/10.1016/j.cor.2022.105764
3. Álvaro, V., Ortega, J., Serrano, J.E., Acuña, L.C., Martinez-Santos, J.C.: Path planning for non-playable characters in arcade video games using the wavefront algorithm. In: 2020 IEEE Games, Multimedia, Animation and Multiple Realities Conference (GMAX), pp. 1–5 (2020). https://doi.org/10.1109/GMAX49668.2020.9256835
4. Biewald, L.: Experiment tracking with weights and biases (2020). https://www.wandb.com/, software available from wandb.com
5. Chatzisavvas, A., Dossis, M., Dasygenis, M.: Optimizing mobile robot navigation based on A-star algorithm for obstacle avoidance in smart agriculture. Electronics **13**(11) (2024). https://doi.org/10.3390/electronics13112057
6. Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J.: Path planning for autonomous vehicles in unknown semi-structured environments. Int. J. Robot. Res. **29**(5), 485–501 (2010). https://doi.org/10.1177/0278364909359210
7. Douglas, D.H.: Least-cost path in GIS using an accumulated cost surface and slopelines. Cartographica: the International Journal for Geographic Information and Geovisualization **31**(3), 37–51 (1994). https://doi.org/10.3138/D327-0323-2JUT-016M
8. Douglas, D.H., Peucker, T.K.: Algorithms for reducing the number of points needed to represent a digitized line or its simplified version. Cartographica **10**(2), 112–122 (1973). https://doi.org/10.3138/FM57-6770-U75U-7727
9. Ducho, F., et al.: Path planning with modified a star algorithm for a mobile robot. Proc. Eng. **96**, 59–69 (2014). https://doi.org/10.1016/j.proeng.2014.12.098
10. Erke, S., Bin, D., Yiming, N., Qi, Z., Liang, X., Dawei, Z.: An improved A-star based path planning algorithm for autonomous land vehicles. Int. J. Adv. Rob. Syst. **17**(5), 1729881420962263 (2020). https://doi.org/10.1177/1729881420962263
11. Flores-Caballero, G., Rodríguez-Molina, A., Aldape-Pérez, M., Villarreal-Cervantes, M.G.: Optimized path-planning in continuous spaces for unmanned aerial vehicles using meta-heuristics. IEEE Access **8**, 176774–176788 (2020). https://doi.org/10.1109/ACCESS.2020.3026666
12. Fournier, D., Martinez, T., Fages, F., Mulard, D.: Metro Energy Optimization through Rescheduling: Mathematical Models and Heuristic Algorithm Compared to MILP and CMA-ES. Ph.D. thesis, Inria Saclay Ile de France (2016)

13. Gad, A.G.: Particle swarm optimization algorithm and its applications: a systematic review. Arch. Comput. Methods Eng. **29**(5), 2531–2561 (2022). https://doi.org/10.1007/s11831-021-09694-4
14. Gao, T., et al.: A deep reinforcement learning approach to mountain railway alignment optimization. Comput.-Aided Civil Infrastruct. Eng. **37**, 73–92 (2021). https://doi.org/10.1111/mice.12694
15. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evol. Comput. **9**(2), 159–195 (2001). https://doi.org/10.1162/106365601750190398
16. Hart, P., Nilsson, N., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. **4**, 100–107 (1968). https://doi.org/10.1109/TSSC.1968.300136
17. Jiang, F., Ma, L., Broyd, T., Li, J., Jia, J., Luo, H.: Systematic framework for sustainable urban road alignment planning. Transp. Res. Part D: Transp. Environ. **120**, 103796 (2023). https://doi.org/10.1016/j.trd.2023.103796
18. Jong, J.C., Jha, M.K., Schonfeld, P.: Preliminary highway design with genetic algorithms and geographic information systems. Comput.-Aided Civil Infrastruct. Eng. **15**(4), 261–271 (2000). https://doi.org/10.1111/0885-9507.00190
19. Jong, J.C., Schonfeld, P.: An evolutionary model for simultaneously optimizing three-dimensional highway alignments. Transport. Res. Part B: Methodol. **37**(2), 107–128 (2003). https://doi.org/10.1016/S0191-2615(01)00047-9
20. Juíček, M., Parák, R., Kdela, J.: Evolutionary computation techniques for path planning problems in industrial robotics: a state-of-the-art review. Computation **11**, 245 (2023). https://doi.org/10.3390/computation11120245
21. Kaleybar, H.J., Davoodi, M., Brenna, M., Zaninelli, D.: Applications of genetic algorithm and its variants in rail vehicle systems: A bibliometric analysis and comprehensive review. IEEE Access, pp. 68972–68993 (2023). https://doi.org/10.1109/ACCESS.2023.3292790
22. Kang, M.W., Schonfeld, P.: Artificial intelligence in highway location and alignment optimization: applications of genetic algorithms in searching, evaluating, and optimizing highway location and alignments. World Sci. (2020). https://doi.org/10.1142/11059
23. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robot. Res. **30**(7), 846–894 (2011). https://doi.org/10.1177/0278364911406761
24. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 – International Conference on Neural Networks. vol. 4, pp. 1942–1948 vol.4 (1995)
25. Khantanapoka, K., Chinnasarn, K.: Pathfinding of 2D & 3D game real-time strategy with depth direction A* algorithm for multi-layer. In: 2009 Eighth International Symposium on Natural Language Processing, pp. 184–188 (2009). https://doi.org/10.1109/SNLP.2009.5340922
26. Kim, E., Jha, M.K., Son, B.: Improving the computational efficiency of highway alignment optimization models through a stepwise genetic algorithms approach. Transport. Res. Part B: Methodol. **39**(4), 339–360 (2005). https://doi.org/10.1016/j.trb.2004.06.001
27. Kwan, C.M., Chang, C.S.: Timetable synchronization of mass rapid transit system using multiobjective evolutionary approach. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **38**(5), 636–648 (2008). https://doi.org/10.1109/TSMCC.2008.923872

28. Liu, Z., et al.: An optimal motion planning method of 7-DOF robotic arm for upper limb movement assistance. In: 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 277–282 (2019). https://doi.org/10.1109/AIM.2019.8868594
29. Loganathan, A., Ahmad, N.S.: A systematic review on recent advances in autonomous mobile robot navigation. Eng. Sci. Technol. Int. J. **40**, 101343 (2023). https://doi.org/10.1016/j.jestch.2023.101343
30. Mondal, S., Lucet, Y., Hare, W.: Optimizing horizontal alignment of roads in a specified corridor. Comput. Oper. Res. **64**, 130–138 (2015). https://doi.org/10.1016/j.cor.2015.05.018
31. Open digital elevation model (OpenDEM) the portal for sharing the 3rd dimension. https://www.opendem.info/index.html
32. Pu, H., et al.: Mountain railway alignment optimization using stepwise and hybrid particle swarm optimization incorporating genetic operators. Appl. Soft Comput. (2019). https://doi.org/10.1016/j.asoc.2019.01.051
33. Pu, H., Wan, X., Song, T., Schonfeld, P., Peng, L.: A 3D-RRT-star algorithm for optimizing constrained mountain railway alignments. Eng. Appl. Artif. Intell. **130**, 107770 (2024). https://doi.org/10.1016/j.engappai.2023.107770
34. Pushak, Y., Hare, W., Lucet, Y.: Multiple-path selection for new highway alignments using discrete algorithms. Eur. J. Oper. Res. **248**(2), 415–427 (2016). https://doi.org/10.1016/j.ejor.2015.07.039
35. Rapalski, A., Dudzik, S.: Energy consumption analysis of the selected navigation algorithms for wheeled mobile robots. Energies **16**(3) (2023). https://doi.org/10.3390/en16031532
36. Ravankar, A., Ravankar, A.A., Kobayashi, Y., Hoshino, Y., Peng, C.C.: Path smoothing techniques in robot navigation: state-of-the-art, current and future challenges. Sensors **18**(9), 3170 (2018). https://doi.org/10.3390/s18093170
37. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice Hall Press, USA, 3rd edn. (2009). https://doi.org/10.5555/1671238
38. Saha, A.K., Arora, M.K., Gupta, R.P., Virdi, M., Csaplovics, E.: GIS-based route planning in landslide-prone areas. Int. J. Geogr. Inf. Sci. **19**(10), 1149–1175 (2005). https://doi.org/10.1080/13658810500105887
39. Smith, M.: Determination of gradient and curvature constrained optimal paths. Comput. Aided Civ. Infrastruct. Eng. **21**, 24–38 (2006). https://doi.org/10.1111/j.1467-8667.2005.00414.x
40. Song, T., et al.: Mountain railway alignment optimization integrating layouts of large-scale auxiliary construction projects. Comput.-Aided Civil Infrastruct. Eng. **38**(4), 433–453 (2023). https://doi.org/10.1111/mice.12839
41. Song, T., Schonfeld, P., Pu, H.: A review of alignment optimization research for roads, railways and rail transit lines. IEEE Trans. Intell. Transp. Syst. **24**, 4738–4757 (2023). https://doi.org/10.1109/TITS.2023.3235685
42. Spivak, M.: A Comprehensive Introduction to Differential Geometry. No. 1 in A Comprehensive Introduction to Differential Geometry, Publish or Perish, Incorporated (1999)
43. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328
44. Sushma, M., Maji, A.: A modified motion planning algorithm for horizontal highway alignment development. Comput.-Aided Civil Infrastruct. Eng. **35**(8), 818–831 (2020). https://doi.org/10.1111/mice.12534

45. Sushma, M., Roy, S., Maji, A.: Optimizing points of intersection for highway and railway alignment—Using path planner method and ant algorithm-based approach. In: Recent Advances in Transportation Systems Engineering and Management: Select Proceedings of CTSEM 2021, pp. 127–143. Springer (2022). https://doi.org/10.1007/978-981-19-2273-2_9
46. Trietsch, D.: A family of methods for preliminary highway alignment. Transp. Sci. **21**(1), 17–25 (1987). https://doi.org/10.1287/trsc.21.1.17
47. Turner, A.K.: Route selection. Photogramm. Eng. Remote. Sens. **44**(12), 1561–1576 (1978)
48. Vázquez-Méndez, M.E., Casal, G., Santamarina, D., Castro, A.: A 3D model for optimizing infrastructure costs in road design. Comput.-Aided Civil Infrastruct. Eng. **33**(5), 423–439 (2018). https://doi.org/10.1111/mice.12350
49. Yang, D., He, Q., Yi, S.: Underground metro interstation horizontal-alignment optimization with an augmented rapidly exploring random-tree connect algorithm. J. Transport. Eng., Part A: Syst. **146**(11), 04020129 (2020). https://doi.org/10.1061/JTEPBS.0000454
50. Yang, D., He, Q., Yi, S.: Bilevel optimization of intercity railway alignment. Transp. Res. Rec. **2675**(11), 985–1002 (2021). https://doi.org/10.1177/03611981211023756
51. Yao, J., Lin, C., Xie, X., Wang, A.J., Hung, C.C.: Path planning for virtual human motion using improved A* star algorithm. In: 2010 Seventh International Conference on Information Technology: New Generations, pp. 1154–1158 (2010). https://doi.org/10.1109/ITNG.2010.53
52. Zhang, T., et al.: 3D constrained hybrid A*: Improved vehicle path planning algorithm for cost-effective road alignment design. Autom. Constr. **166**, 105645 (2024). https://doi.org/10.1016/j.autcon.2024.105645
53. Zhong, J., Li, D., Cai, W., Chen, W.N., Shi, Y.: Automatic crowd navigation path planning in public scenes through multiobjective differential evolution. IEEE Trans. Comput. Social Syst. **11**(1), 905–918 (2022). https://doi.org/10.1109/TCSS.2022.3217417