BEYOND THE SHORTEST PATH: A non-linear altimetric effort model for multi-objective pathfinding in digital games[1]

Paulo Renato Souza Magalhães[2]
Mateus Felipe Tymburibá Ferreira[3]

Submitted on: 11/06/2026 | Approved on: 18/06/2026

ABSTRACT

Pathfinding in digital games is usually treated as a single-objective shortest-path problem, but navigation over three-dimensional terrain must balance competing criteria: distance, exposure to danger, and the metabolic effort imposed by slope. Weighted-sum scalarization, the dominant production approach, is efficient but structurally unable to reach Pareto-optimal solutions in non-convex regions of the objective space. This work brings exact multi-objective search (NAMOA* and LazyLTMOA*) together with a novel biomechanically grounded non-linear altimetric cost model, inspired by metabolic measurements, and evaluates the combination over 3,120 scenarios from the *Dragon Age: Origins* benchmark (Moving AI Lab). Two findings stand out. First, the fraction of Pareto-optimal solutions that no weighted-sum query can reach is large and strongly map-dependent: it reaches 99.88% in extreme scenarios, and 107 of 143 maps show a median above 90%, reframing the choice between scalarization and exact search as a map-conditional decision rather than a universal trade-off. Second, the proposed effort model does not merely re-scale the linear baseline; it produces geometrically distinct paths, cutting the mean absolute slope along the chosen route by 41.7% in mountainous terrain at a modest distance cost. We further confirm that NAMOA* and LazyLTMOA* return identical frontiers, that the lazy variant is faster in 88.4% of non-trivial scenarios (median 1.08×), and that exact search becomes intractable beyond short path lengths under an interactive time budget.

Keywords: Multi-Objective Pathfinding. Pareto Frontier. NAMOA*. LazyLTMOA*. Altimetric Cost Model. Game AI. Unity.

---

[1] Scientific article elaborated for the undergraduate thesis (Trabalho de Conclusão de Curso) of the Computer Engineering program at the Nova Gameleira campus of the Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG).

[2] Undergraduate student in Computer Engineering at the Nova Gameleira campus of the Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG). paulorenatosouza02@gmail.com

[3] Thesis advisor. Faculty member of the Department of Computing at the Nova Gameleira campus of the Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG).

1 INTRODUCTION

Pathfinding is one of the most common problems in digital game development: non-player characters (NPCs) must navigate complex environments, avoid obstacles, and reach their goals with motion that is both computationally cheap and visually believable. Since the foundational work of Dijkstra (1959) and Hart *et al.* (1968), the standard approach optimizes a single objective, geometric distance, with A* adding admissible heuristics to speed up the search without changing the underlying single-objective formulation.

Modern commercial games expose the limits of that formulation. Open-world titles such as *The Elder Scrolls V: Skyrim*, *The Legend of Zelda: Breath of the Wild*, and *Red Dead Redemption 2* feature large three-dimensional terrains where intelligent navigation must weigh several criteria at once: a path should be short, but it should also avoid dangerous areas (swamps, water, enemy patrol zones) and account for the physical effort imposed by slope. A patrol unit crossing a mountain range cannot reasonably take the geometrically shortest route if that route climbs a cliff; an ambush system must trade line of sight against approach effort. These are intrinsically multi-objective problems, and they sit squarely within an established AIIDE topic, pathfinding, now extended to biomechanically realistic terrain costs.

Three concerns motivate this work. The first is a **modeling** concern: most production systems either ignore altimetry or include it through a linear penalty, which cannot represent the non-linear relationship between effort and slope in locomotion. The metabolic measurements of Minetti *et al.* (2002) show that the energy cost of walking follows a fifth-degree polynomial of slope whose minimum lies at a slight downhill, not on flat ground, a behavior no linear model can capture. The second is **algorithmic**: weighted-sum scalarization is widely used but, as Das e Dennis (1997) showed, can only recover Pareto-optimal solutions on the convex hull of the objective space, leaving solutions in non-convex regions hidden for every choice of weights. Exact methods such as NAMOA* (Mandow; Pérez-de-la-Cruz, 2005) remove this blind spot but face scalability limits, and recent work such as LazyLTMOA* (Hernández *et al.*, 2023a) promises better efficiency through lazy dominance checks whose empirical behavior in game pathfinding remains poorly documented. The third concern is **applied**: these choices must be integrated in a real engine, with standardized benchmarks that make the results reproducible.

This work therefore empirically compares weighted-sum and Pareto-frontier approaches to multi-objective pathfinding on terrains with altimetric variation, using a non-linear effort function grounded in biomechanical principles. We formulate and implement the effort model, adapt Dijkstra and A* to a weighted-sum scalarization of distance, danger, and effort, implement NAMOA* (Mandow; Pérez-de-la-Cruz, 2005; Mandow; Pérez-de-la-Cruz, 2010) and LazyLTMOA* (Hernández *et al.*, 2023a) with an informed Ideal Point heuristic, and run a systematic comparison over the standardized Moving AI Lab benchmarks inside the Unity engine.

The research question is: *how does a biomechanically grounded non-linear altimetric cost model affect the quality and diversity of solutions found by multi-objective pathfinding algorithms, compared to weighted-sum approaches?* In answering it, this work makes four contributions:

- a **non-linear altimetric effort cost model**, a polynomial approximation of the Minetti metabolic curve that preserves the properties relevant for pathfinding (non-linearity and slope asymmetry) and is cheap to evaluate at every edge;

- an **empirical characterization of the gap between weighted-sum and Pareto-frontier approaches** on a standardized benchmark with realistic three-objective terrain costs, quantifying both the fraction of optimal trade-offs invisible to scalarization (up to 99.88%) and its strong heterogeneity across maps;
- a **direct comparison of NAMOA* and LazyLTMOA*** on game-domain benchmarks, confirming correctness equivalence (identical frontiers in all completed scenarios) and measuring the lazy variant's modest but consistent runtime advantage; and
- a **reproducible Unity-based experimental pipeline** with fixed seeds, deterministic sampling, and full instrumentation, intended as infrastructure for further research.

2 BACKGROUND AND RELATED WORK

This section positions the work against the most relevant prior art in three areas: exact multi-objective search, the heuristic that makes it tractable, and terrain cost models. We assume familiarity with single-objective graph search (Dijkstra's algorithm and A* (Dijkstra, 1959; Hart *et al.*, 1968)) and with the basic vocabulary of multi-objective optimization. A path π carries a **cost vector** $\mathbf{c}(\pi) \in \mathbb{R}^n_{\geq 0}$; vector $\mathbf{u}$ *dominates* $\mathbf{v}$ ($\mathbf{u} \prec \mathbf{v}$) when $u_i \leq v_i$ for all i and $u_j < v_j$ for some j; and the **Pareto frontier** is the image of the non-dominated set. Two solution strategies compete. **Weighted-sum scalarization** collapses the vector into $c_{\text{ws}} = \sum_i w_i c_i$ so that any single-objective algorithm applies directly; it is efficient but, as Das e Dennis (1997) proved, recovers only solutions on the convex hull of the frontier. Solutions in non-convex regions, called **unsupported**, are unreachable for every weight vector. **Pareto-frontier search** instead returns the complete non-dominated set in one execution, at the cost of a frontier that can grow exponentially with graph size. Quantifying how large the unsupported set is in practice, and when it justifies exact search, is one of the gaps this work addresses.

2.1 Multi-Objective Search Algorithms

Multi-objective A* generalizes single-objective search by storing, at each vertex, a *set* of non-dominated cost vectors rather than a single g-value. Successive algorithms have refined the dominance-check machinery that controls combinatorial growth (Table 1). MOA* (Stewart; White, 1991) allowed node reopening; NAMOA* (Mandow; Pérez-de-la-Cruz, 2005; Mandow; Pérez-de-la-Cruz, 2010) removed reopening under a consistent heuristic and remains the canonical exact solver; later work reduced dominance-check cost through dimensionality reduction (Pulido *et al.*, 2015), $O(1)$ bi-objective checks (Hernández *et al.*, 2023b), balanced-tree structures (Ren *et al.*, 2022), and the simple-array, lazy strategy of LTMOA* (Hernández *et al.*, 2023a). This work focuses on NAMOA* and the lazy variant of LTMOA* (LazyLTMOA*): both compute the exact frontier, and they differ only in *when* dominance is verified, which makes them an ideal controlled pair for measuring the practical payoff of lazy checks in the game domain.

Table 1 – Representative multi-objective search algorithms. This work compares NAMOA* and LazyLTMOA*, the canonical exact solver and its most recent lazy refinement.

Algorithm	**Year**	**# Obj.**	**Worst-case cost**	**Remark**
MOA*	1991	≥ 2	$O(\lvert E\rvert\, N + \lvert V\rvert\, N^2)$	Allows node reopening
NAMOA*	2005	≥ 2	$O(\lvert E\rvert\, N + \lvert V\rvert\, N^2)$	No reopening with consistent heuristic
NAMOA*$_{\mathrm{dr}}$	2015	≥ 2	$O(\lvert E\rvert\, N + \lvert V\rvert\, N^2)$	Dimensionality reduction
BOA*	2020	$= 2$	$O(\lvert E\rvert\, N)$	$O(1)$ dominance for bi-objective
EMOA*	2022	≥ 2	$O(\lvert E\rvert\, N \log N)$	AVL-tree dominance checks
LTMOA*	2023	≥ 2	$O(\lvert E\rvert\, N + \lvert V\rvert\, N^2)$	Simple-array dominance, lazy variant

Source: Prepared by the author (2026).

In the worst-case bounds above, N denotes the number of non-dominated cost vectors stored across the search, which grows combinatorially with the number of objectives and is the term that dominates the cost. NAMOA* and LazyLTMOA* share the same asymptotic bound because they expand the same set of labels in the same order: the lazy variant changes only *when* dominance is verified, not *how many* labels are processed, so its gain is a smaller constant factor and not a lower order of growth. Our experiments confirm this directly, with both algorithms expanding identical label counts on every completed scenario (Section 4.3).

NAMOA* keeps, for each vertex v, sets $G_{\mathrm{op}}(v)$ and $G_{\mathrm{cl}}(v)$ of non-dominated cost vectors for partial paths that are, respectively, pending or already expanded. OPEN holds `(vertex, g-vector)` labels ordered by the lexicographically smallest $\mathbf{f} = \mathbf{g} + \mathbf{h}$. A label is expanded by moving its vector from G_{op} to G_{cl} and generating successors whose candidate vectors are kept only if non-dominated by $G_{\mathrm{op}} \cup G_{\mathrm{cl}}$ and by the goal-reaching solution set. With an admissible, consistent heuristic the algorithm is correct and complete and expands no label twice. LazyLTMOA* (Hernández *et al.*, 2023a) preserves this correctness while lowering constant factors in two ways: it replaces the per-vertex sets with a single flat array plus a *minimum-g vector* $\mathbf{g}_{\min}(v)$ summarizing the lower envelope, and it *defers* full set-dominance verification until a label is about to be expanded, silently skipping labels that a sibling has dominated in the meantime. The two algorithms are asymptotically equivalent and must return the same frontier on every input, a property we verify empirically (Section 4).

2.2 Ideal Point Heuristic

Both exact algorithms require a vector heuristic $\mathbf{h}(v) \in \mathbb{R}^n_{\geq 0}$ that is componentwise admissible: $h_i(v) \leq c_i^*$ for every Pareto-optimal cost-to-go c_i^*. The trivial choice $\mathbf{h} = \mathbf{0}$ degrades the search to multi-objective Dijkstra. We instead use the **Ideal Point heuristic**: for each objective i, a single-objective Dijkstra run *in reverse* from the goal yields $h_i(v) = \min\{c_i(\pi) : \pi \text{ from } v \text{ to goal}\}$, the cost-to-go on objective i alone. The resulting vector is component-wise admissible and consistent. Its pre-computation cost (n reverse Dijkstra runs per query) is amortized over the much larger frontier search and tightens the lower bound on every component, pruning OPEN substantially. This heuristic quality turns out to matter for interpreting the modest size of LazyLTMOA*'s advantage (Section 4.3).

2.3 Terrain Cost Models

Multi-objective pathfinding over uneven ground needs cost functions that reflect the physics of locomotion. The simplest altimetric model penalizes height difference linearly, $c_{\text{linear}}(u, v) = d(u, v) + k\,|h(v) - h(u)|$ (Rees, 2004). It is trivial to compute but charges ascent and descent identically, treats one steep climb as equivalent to many shallow ones, and never penalizes slope beyond its proportional contribution to elevation change. Tobler's hiking function (Tobler, 1993) improves on this by modeling walking speed as $v(m) = 6\exp(-3.5\,|m + 0.05|)$ km/h, where m is the terrain grade (the slope, defined as the ratio of vertical rise to horizontal run, dimensionless), capturing a peak speed at a slight downhill and rapid decay with slope. The biomechanically authoritative reference is the metabolic model of Minetti *et al.* (2002), who measured walking energetics directly on grades from -45% to $+45\%$ and fitted

$$C_w(m) = 280.5\,m^5 - 58.7\,m^4 - 76.8\,m^3 + 51.9\,m^2 + 19.6\,m + 2.5, \tag{1}$$

with C_w in J/(kg·m). Three of its properties matter for pathfinding: minimum cost at about -10% grade, an eccentric-to-concentric efficiency ratio near 5:1, and a disproportionate penalty for steep ascents. Evaluating Equation (1) at every edge would be wasteful; Section 3 proposes a cheaper polynomial approximation that preserves these properties. To our knowledge, no prior game-pathfinding study has combined a Minetti-grounded effort term with exact Pareto search on a standardized benchmark, which is the gap this work fills.

2.4 Benchmarks

We evaluate on the Moving AI Lab grid benchmarks (Sturtevant, 2012), the de facto standard for grid pathfinding. Maps are ASCII `.map` files (cells coded as free, obstacle, tree, swamp, or water) paired with `.scen` scenario files that list source-goal pairs grouped into difficulty buckets, each annotated with the pre-computed pure-distance `optimal_length` we use for cross-validation (Section 3.5). We use the full *Dragon Age: Origins* (DAO) set, whose maps mix open and corridor topologies at sizes from 50×50 to 1024×1024, making it representative of commercial role-playing navigation.

3 METHODOLOGY

This section describes the graph model and cost vector (Section 3.1), the proposed non-linear effort function (Section 3.2), the four implemented algorithms (Section 3.3), terrain generation and experimental design (Section 3.4), and the validation that guards against silent bugs (Section 3.5). Throughout, the emphasis is on the choices that affect the *fairness* of the comparison; low-level reproducibility details are deferred to the project artifacts.

3.1 Graph Model and Cost Vector

Each benchmark map is loaded as an 8-connected grid of walkable and obstacle cells. Diagonal moves follow the standard corner-cutting restriction: a diagonal edge is allowed only if both orthogonal neighbours it passes between are walkable, preventing characters from squeezing between corner-adjacent obstacles. Every path is evaluated against a three-dimensional cost vector $\mathbf{c}(\pi) = (D, G, E) \in \mathbb{R}^3_{\geq 0}$, combining geometric distance D, accumulated danger G, and accumulated altimetric effort E. This instantiates the multi-objective setting of Section 2.1 with $n = 3$: a path π_1 dominates π_2 when it is no worse on all three components and strictly better on at least one, the Pareto frontier is the set of non-dominated paths from start to goal, and a frontier solution is **supported** if it lies on the convex hull of that set (hence reachable by some weight vector) and **unsupported** otherwise. The three objectives are heterogeneous by construction, distance is purely geometric, danger is a property of the terrain, and effort depends on elevation change, so no two are redundant and the frontier is genuinely three-dimensional. The distance of an edge is 1 for orthogonal and $\sqrt{2}$ for diagonal moves, matching the metric behind the `optimal_length` field used for validation. The danger of a cell is $G(v) = D_{\text{terrain}}(\text{flags}(v)) + D_{\text{field}}(v)$, where the discrete term assigns fixed costs by terrain type (free $= 0$, swamp $= 5$, water $= 10$) and the continuous term is a Perlin-noise field (Section 3.4); non-negativity is enforced at evaluation to preserve shortest-path correctness. The effort component is the modeling contribution and is defined next.

3.2 Proposed Non-Linear Effort Function

The effort of edge (u, v) is

$$c_{\text{effort}}(u, v) = \alpha\, s^2 + \beta\, \max(0,\, s)^3, \quad s = \frac{h(v) - h(u)}{d_h(u, v)}, \tag{2}$$

with $\alpha = 1.0$, $\beta = 2.0$, where s is the slope, defined as the ratio of vertical displacement $h(v) - h(u)$ to horizontal displacement $d_h(u, v)$ between the two cells (the same quantity denoted m in the Minetti and Tobler models of Section 2.3). The quadratic term symmetrically penalizes steep terrain, capturing braking and impact loads. The cubic term vanishes on descents but penalizes ascents, encoding the biomechanical asymmetry of lifting body mass against gravity.

The values $\alpha = 1.0$ and $\beta = 2.0$ were chosen, rather than fitted, to reproduce the qualitative shape of the Minetti curve (1) over the slope range of game terrain. The ratio $\beta/\alpha = 2$ makes the cubic ascent penalty dominate the quadratic term once the grade exceeds $s = 0.5$, which reproduces the disproportionate cost of steep climbs that is the most pathfinding-relevant feature of the metabolic data, while keeping the two terms comparable at the gentle slopes that make up most of the terrain. Figure 6 shows that, under this ratio, the resulting paths track the contour-following behaviour expected from the exact model. A systematic sensitivity analysis of the $\alpha{:}\beta$ ratio, and a calibration against human-trial energetics, are left as future work (Section 5); the present study fixes a single physically motivated ratio so that the comparison isolates the effect of non-linearity itself rather than of parameter tuning.

Equation (2) is a deliberate approximation of the Minetti curve (1), not a literal implementation, for three reasons. It is cheaper, requiring three multiplications and two additions per edge against six and five for the exact fifth-degree polynomial, which matters across the 10^4 to 10^7 edge evaluations per scenario. It preserves the pathfinding-relevant properties (non-linearity, ascent-descent asymmetry, disproportionate penalty for steep climbs). And because only the relative ordering of paths affects the Pareto frontier, which is invariant under positive affine transformations of any single component, the choice of (α, β) is one of relative emphasis rather than physical fidelity. Within the slope range relevant for game terrain ($|s| \leq 0.5$, inside Minetti's measured ± 0.45 interval), the approximation keeps the qualitative shape of the metabolic curve: strictly monotone in $|s|$ for descents and strongly convex for ascents.

3.3 Implemented Algorithms

Four algorithms share a common `IPathfinder` interface and operate on the same grid, scenarios, and edge costs. The two **weighted-sum baselines** optimize the scalar $c_{\text{ws}}(u, v) = w_1 D + w_2 G + w_3 E$ over a binary-heap priority queue: **Dijkstra-WS** stops when the goal is dequeued, and **A*-WS** adds the admissible distance heuristic $h_{\text{ws}}(v) = w_1 h_{\text{oct}}(v, t)$ using octile distance. The danger and effort components contribute nothing to the heuristic, since no constant-time admissible bound is available for terrain not yet visited; when $w_1 = 0$ the heuristic collapses to zero and A*-WS correctly degenerates to Dijkstra-WS. Because both optimize the same scalar with an admissible heuristic, they return identical-cost paths (A* merely expands fewer nodes), so we report them as a single weighted-sum reference. Five weight configurations are evaluated per scenario (Table 2), giving ten weighted-sum points per scenario.

The five configurations were chosen to probe the corners and the centre of the objective simplex: the three single-objective vertices (`dist_only`, `danger_only`, `effort_only`) bound the reachable region, the symmetric `balanced` point samples its interior, and `dist_effort` isolates the distance-effort trade-off with danger neutralized, the pair most directly tied to the altimetric contribution. This discrete set is deliberately small: its purpose is to locate the weighted-sum solutions relative to the exact frontier, not to densely sample the simplex. A dense weight sweep, discretizing the simplex into a fine grid to confirm that the unsupported solutions remain unreachable for *every* weight rather than these five, would further reinforce Claim A and is identified as future work (Section 5).

Table 2 – Weight configurations used for the weighted-sum algorithms.

Configuration	w_1	w_2	w_3	**Meaning**
`dist_only`	1.00	0.00	0.00	Pure shortest path
`danger_only`	0.00	1.00	0.00	Safest path
`effort_only`	0.00	0.00	1.00	Least effort
`balanced`	0.33	0.33	0.34	Symmetric trade-off
`dist_effort`	0.50	0.00	0.50	Short and flat (ignore danger)

Source: Prepared by the author (2026).

The two **exact solvers** follow the references of Section 2.1. NAMOA* (Mandow; Pérez-de-la-Cruz, 2005; Mandow; Pérez-de-la-Cruz, 2010) keeps per-vertex $G_{\text{op}}/G_{\text{cl}}$ arrays and applies three pruning rules at each expansion: heuristic pruning (discard labels whose $\mathbf{f}$-vector is dominated by the solution set), local pruning (discard a generated vector dominated by $G_{\text{op}}(w) \cup G_{\text{cl}}(w)$), and insertion pruning (when inserting a surviving vector, remove from $G_{\text{op}}(w)$ any vector it dominates). LazyLTMOA* (Hernández *et al.*, 2023a) instead stores a single flat array per vertex plus an incrementally updated $\mathbf{g}_{\min}(v)$, runs only the cheap component-wise check against $\mathbf{g}_{\min}(w)$ on the generator side, and verifies full set dominance only at extraction time, deferring exhaustive checks for labels that will be transitively dominated before reaching the front of the queue. Both use the same Ideal Point heuristic, computed by three reverse Dijkstra runs per query.

3.4 Terrain Generation and Experimental Design

Heightmaps and danger fields are generated procedurally to give controlled, reproducible conditions over the structurally diverse DAO maps. The heightmap is $h(x, y) = A \cdot \mathrm{Perlin}(x/S + o,\, y/S + o)$ with amplitude $A = 30$, scale $S = 20$, and offset derived from a fixed seed of 42; the continuous danger field uses an independent Perlin process (seed 1337) so the danger gradient is statistically independent of elevation, clamped to preserve non-negativity. The three Perlin parameters control distinct aspects of the terrain: the amplitude A sets the maximum elevation range and therefore the steepness of the slopes the effort model responds to; the scale S sets the spatial wavelength of the features, with larger values producing broad, gentle hills and smaller values producing rugged, high-frequency relief; and the seed fixes the particular realization, making every map reproducible. The chosen values ($A = 30$, $S = 20$) yield slopes within Minetti's measured ± 0.45 range across most edges, keeping the effort model in its validated regime. Replicating the experiments on terrain with deliberately extreme features, such as cliffs or plateaus, would test the model at the boundary of that regime and is left as future work (Section 5). For each map, 20 scenarios are drawn from its `.scen` file by stratified sampling: scenarios are sorted by `optimal_length`, partitioned into 10 quantiles, and 2 are taken per quantile, giving uniform difficulty coverage and $156 \times 20 = 3{,}120$ scenarios in total. For aggregation, scenarios are bucketed by $\lfloor$`optimal_length`$/4\rfloor$ and grouped into five macro-tiers (`easy` 0–9, `medium` 10–24, `hard` 25–49, `extreme` 50–99, `intractable` 100+). The single execution cap is a per-algorithm wall-clock timeout of 30 seconds; frontier-size and node caps are intentionally disabled so that the timeout alone exposes scalability limits. For each algorithm-scenario pair we record status (`ok`, `timeout`, or `skipped_unreachable`), frontier size, the cost components of each solution, expanded and generated label counts, wall-clock time, and frontier hypervolume. The hypervolume is computed against a per-scenario reference point set to 1.1 times the maximum value observed on each objective across the frontier, placing the reference just beyond the worst attained value on every axis so that all solutions contribute positive volume. Because the reference point is scenario-specific, hypervolume values are used here only as an internal cross-check that NAMOA* and LazyLTMOA* return identical frontiers, not as an absolute quantity compared across scenarios.

All experiments ran on a single Windows 11 workstation (Ryzen 5 7600X, 16 GB DDR5) with the algorithms implemented in C# under Unity 6000.3.6f1, using the Burst Compiler and `NativeArray` containers to keep the inner loops free of garbage-collection pressure and a custom binary heap following Lague's `IHeapItem` pattern. Each scenario was executed under twelve algorithm-configuration combinations (ten weighted-sum runs plus two exact runs), for $3{,}120 \times 12 = 37{,}440$ executions in roughly 8.4 hours. Absolute timings are therefore platform-specific, but the *relative* comparisons of interest, NAMOA* against LazyLTMOA* and the linear against the non-linear cost model, run on the same machine and scenarios and are robust to hardware. The result CSV, the cost-model comparison JSON, the Unity source, the figure scripts, and the maps form the full reproducibility set, available in the project repository.[4]

3.5 Validation of the Implementation

Three layers of validation guard the pipeline. First, a **distance ground truth**: for every scenario the Dijkstra-WS `dist_only` run must return a path whose cost equals the `optimal_length` field within floating-point tolerance, implicitly validating the parser, the neighbour-baking routine, the distance formula, and the priority queue. Second, a **weighted-sum cross-check**: Dijkstra-WS and A*-WS at the same weights must return equal-cost paths. Third, a **Pareto frontier cross-check**: in every scenario where both exact solvers complete, their frontiers must match exactly in size, hypervolume, and component-wise content after sorting, the strongest available correctness test given their substantially different internal machinery. This last property held in all 945 scenarios where both produced a result, with frontier-size and hypervolume differences of exactly zero, confirming that the lazy optimizations lose no solutions.

4 RESULTS AND DISCUSSION

This section presents and interprets the findings, organized around four claims. Claim A addresses the gap between weighted-sum and Pareto-frontier approaches; Claim B characterizes the scalability of exact search; Claim C quantifies the runtime relationship between LazyLTMOA* and NAMOA*; and Claim D shows that the proposed effort model produces qualitatively distinct paths.

Across the 3,120 scenarios, NAMOA* completed within the 30-second budget on 945 scenarios, timed out on 2,156, and was skipped on 19 where the goal was provably unreachable. LazyLTMOA* was run only when NAMOA* completed, so its completion count is also 945. In all 945 of these scenarios the two algorithms returned identical Pareto frontiers, coinciding in frontier size, hypervolume, and exact component-wise content. This confirms the correctness equivalence anticipated in Section 2.1 and underpins the interpretation of Claims B and C, where the two algorithms behave identically in every respect except per-expansion cost.

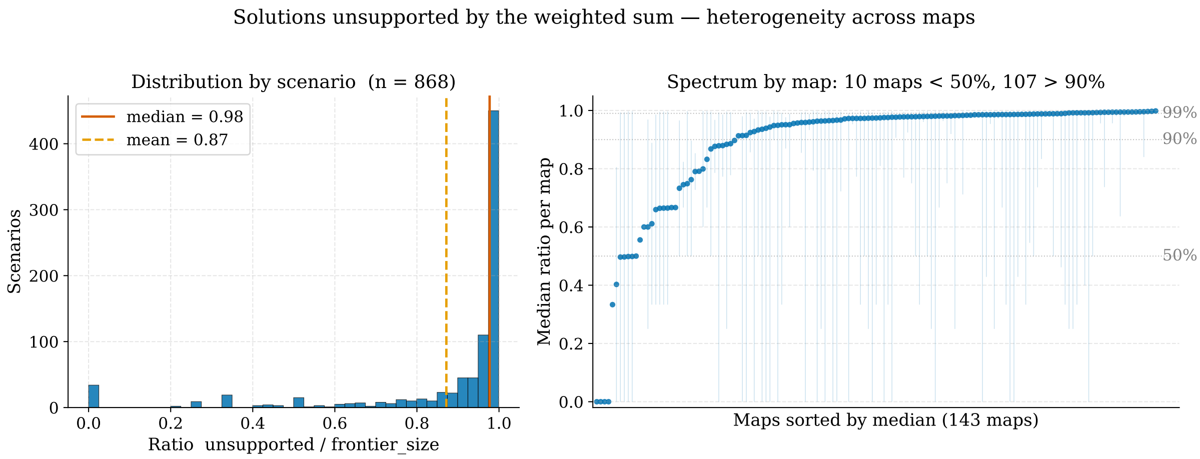
4.1 Claim A: The Hidden Half of the Trade-off Space

Statement. Weighted-sum scalarization hides a large and scenario-dependent fraction of Pareto-optimal solutions, regardless of the chosen weights. The phenomenon is structural rather than residual: in most non-trivial scenarios the majority of optimal trade-offs are unsupported.

Evidence. For each scenario in which NAMOA* completed, we compute the **unsupported ratio**, the fraction of Pareto-optimal solutions not on the convex hull of the frontier and therefore unreachable by any weighted-sum query. Figure 1 aggregates this ratio over the 868 scenarios with frontier size ≥ 2, the only ones for which it is non-trivially defined. The per-scenario distribution has median 0.98 and mean 0.87, and the heterogeneity across maps is wide: of the 143 distinct maps, ten show a median ratio below 0.5 while 107 show a median above 0.9. The high mean is thus a property of most maps, not an artifact of a few outliers. In individual extreme scenarios the ratio reaches 99.88%: `rmtst03` bucket 9, for instance, returns 4,971 Pareto-optimal solutions of which 4,965 are unsupported.

[4] Repository: <https://github.com/PaulitoRenatito/TCC_Pathfinding>

Figure 1 – **In the median scenario, 98% of Pareto-optimal solutions are unreachable by any weighted-sum query, and 107 of 143 maps exceed a 90% median, so scalarization's blind spot is the common case rather than the exception.** Left: per-scenario distribution of the unsupported ratio over 868 scenarios with $|F| \geq 2$ (mean and median marked). Right: per-map median ratio, one dot per map, 143 maps sorted by median, with vertical lines showing per-map spread.



Source: Prepared by the author (2026).

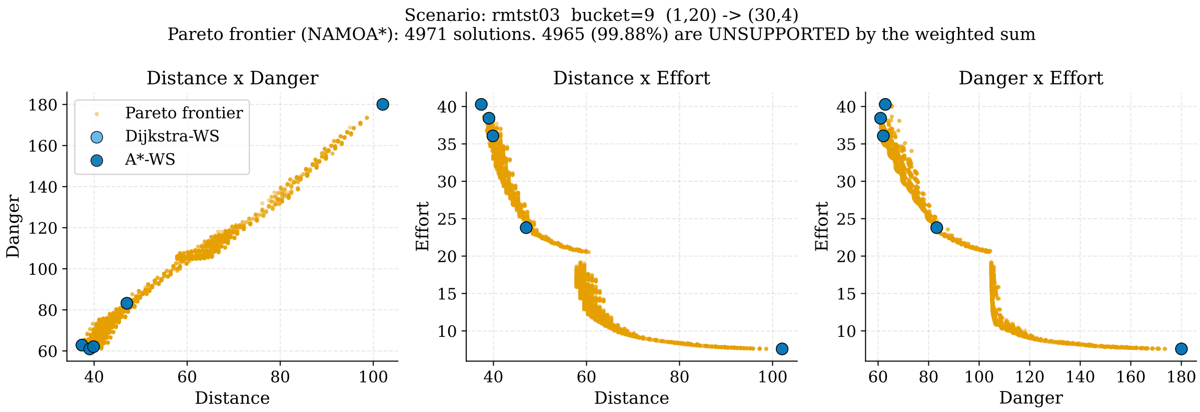
Interpretation. The result has two consequences for practice. A designer who tunes weight vectors expecting to explore the trade-off space is in fact navigating only its convex hull, typically a small minority of the actual optimal solutions. And because the unsupported ratio is strongly map-dependent, no single recommendation holds: on the ten maps with median below 0.5, weighted-sum is a reasonable approximation, whereas on the 107 above 0.9 it is structurally unable to expose the trade-offs that matter. This reframes the choice between weighted-sum and Pareto frontier as a *map-conditional* decision rather than a universal one.

It is worth being explicit about *when* computing the full frontier is the right goal, since for real-time, in-game routing a single path is needed and wall-clock time is the binding constraint. The setting this work targets is the *offline, design-time* one: the frontier is computed once, away from the interactive loop, so that a level designer (or an automated tool) can inspect the full set of optimal trade-offs and choose among them, or so that a population of distinct NPC behaviours can be drawn from different regions of the same frontier. In that setting completeness, not latency, is what matters, and the cost of exact search is paid once rather than per frame. The aim of this study is accordingly to characterize how the algorithm families behave under a non-linear effort cost, not to advocate running exact multi-objective search inside the game loop.

Computing the full frontier, however, does not by itself resolve which solution to use: selecting a single path from a frontier of thousands remains an open problem. Several strategies are available and worth contrasting in future work. One is to *prune* the frontier to a small, evenly distributed subset, for instance by ε-dominance, which partitions the objective space into a grid and keeps one representative per cell, yielding a bounded approximation whose size the designer controls. Another is to use a weighted-sum criterion *inside* the exact search as a tie-break or ranking rule, guiding the algorithm toward a designer-specified region of the frontier without sacrificing exactness. Characterizing which of these selection strategies best serves game design is a natural continuation of this work (Section 5).

Figure 2 makes the geometric reason visible for `rmtst03` bucket 9. The pairwise projections of the frontier show pronounced non-convex regions; the weighted-sum points all cluster on the convex hull, leaving a wide “belly” of unsupported solutions that no scalar combination of objectives can reach.

Figure 2 – **Of the 4,971 Pareto-optimal solutions in this scenario, 4,965 (99.88%) are unsupported, so the ten weighted-sum points recover only the six convex-hull corners and miss the entire interior of the frontier.** Three pairwise projections for `rmtst03` bucket 9: small dots are Pareto-optimal solutions, large dots the weighted-sum points under the five weight configurations.

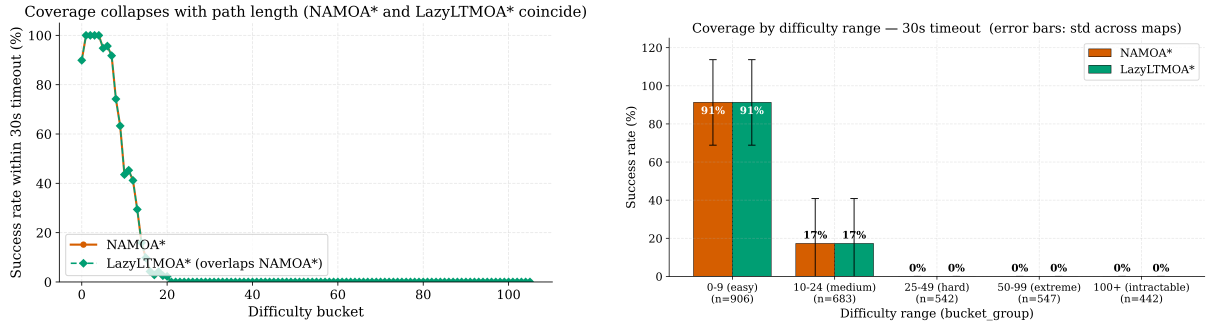


Source: Prepared by the author (2026).

4.2 Claim B: Exact Multi-Objective Search Does Not Scale

Statement. The success rate of exact frontier computation declines sharply with scenario difficulty. Above a moderate optimal-path length, computing the full frontier becomes intractable within any plausible real-time budget. Because NAMOA* and LazyLTMOA* explore the graph identically, this ceiling applies equally to both.

Evidence. Grouping the scenarios into five macro-tiers by optimal length, completion within 30 seconds falls from 91% on `easy` scenarios (0–9, $n = 906$) to 17% on `medium` (10–24, $n = 683$) and to 0% on `hard`, `extreme`, and `intractable` tiers ($n = 542, 547, 442$). NAMOA* and LazyLTMOA* are indistinguishable at this resolution: they time out on exactly the same scenarios, so their coverage curves coincide (Figure 3a), and the per-tier bars are identical (Figure 3b).



(a) Per-bucket coverage; the two curves coincide.

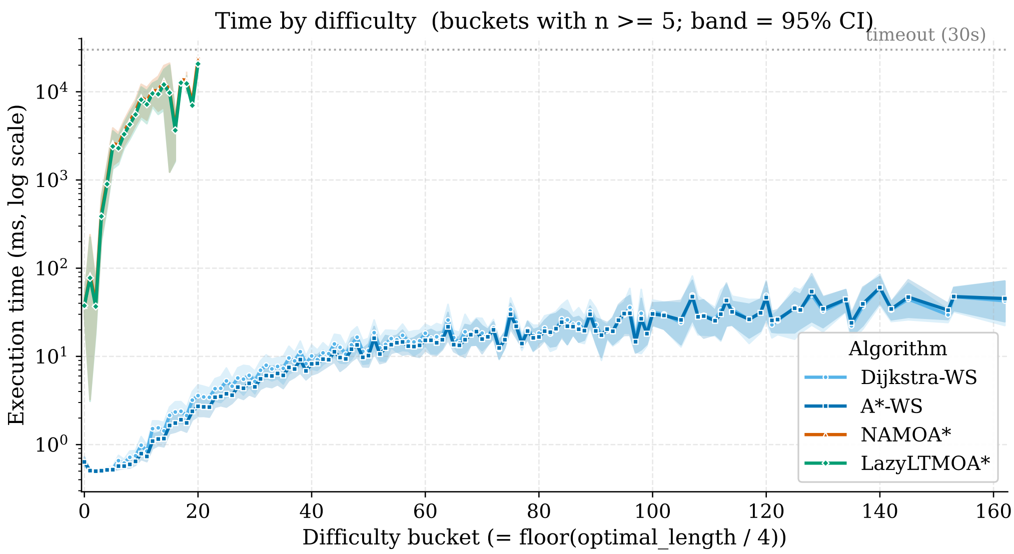
(b) Aggregated by difficulty tier.

Figure 3 – **Coverage collapses from 91% on easy scenarios to 0% beyond an optimal length of about 25 cells, marking a hard scalability ceiling that NAMOA* and LazyLTMOA* hit identically.** Left: completion rate per difficulty bucket, with the two algorithms overlapping exactly. Right: aggregation over five tiers; error bars are ±1 standard deviation of the per-map rate and may extend past 100% because the distribution is bounded but the bars are symmetric.

Source: Prepared by the author (2026).

Conditional on completion, runtime also grows quickly. Figure 4 shows the per-bucket distribution of execution time: by bucket 10 the median is already above 5 seconds, and by bucket 15 the upper quartile reaches the 30-second timeout.

Figure 4 – **Median runtime crosses 5 seconds by bucket 10 and the upper quartile reaches the 30-second timeout by bucket 15, confirming that the Pareto frontier grows combinatorially with path length.** Per-bucket distribution of wall-clock time on completed scenarios (log scale); the dashed line marks the timeout.



Source: Prepared by the author (2026).

Interpretation. The combinatorial growth of the frontier with path length, together with the cost of dominance maintenance at every expansion, places a hard ceiling on the scenarios solvable exactly under interactive budgets. The ceiling is dictated by the frontier itself, not by the algorithm: the lazy refinements of Claim C lower the constant factor but cannot change the asymptotics, which is why both algorithms collapse at the same difficulty. For deployment, exact methods are suitable for short tactical paths (room- or sector-scale navigation) but not for full open-world routing without approximation, partitioning, or precomputation.

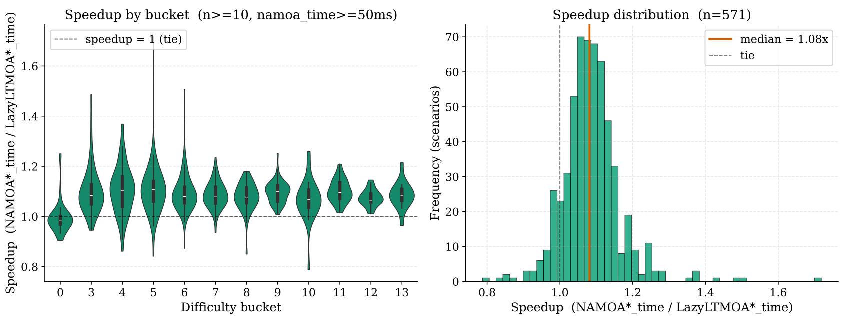
4.3 Claim C: LazyLTMOA*'s Advantage Comes From Cheaper Expansions, Not Fewer

Statement. On scenarios where both complete, LazyLTMOA* is faster than NAMOA* on the large majority of non-trivial cases, but the speedup is modest, and it arises entirely from a lower cost per expansion rather than from exploring less of the graph.

Evidence. Restricting attention to scenarios where NAMOA*'s time exceeded 50 ms (to avoid timing noise on trivial cases), LazyLTMOA* is faster in 88.4% of the comparable scenarios, with a median speedup of 1.08× and a maximum of 1.72× (Figure 5). The distribution is asymmetric: the lower tail, where Lazy is slower, is short and concentrated near 1.0, while the upper tail extends past 1.5×.

To confirm that this advantage is not an artifact of measurement noise, we applied the Wilcoxon signed-rank test to the paired execution times of NAMOA* and LazyLTMOA* over the 945 scenarios where both completed. The test is appropriate here because the data are naturally paired (the same scenario run by both algorithms) and the runtime distribution is strongly right-skewed rather than normal. The time reduction of LazyLTMOA* is highly significant ($p < 10^{-90}$; median paired difference of 7 ms, 95% bootstrap confidence interval $[4, 11]$ ms; effect size $r = 0.78$), so the consistent 1.08× advantage reflects a real, if small, difference rather than noise. The effect is concentrated in the harder scenarios, where the longer linear scans of the dominance structure that the $\mathbf{g}_{\min}$ filter replaces occur most often.

Figure 5 – **LazyLTMOA* is faster than NAMOA* in 88.4% of non-trivial scenarios, with a median speedup of 1.08×, a consistent if modest advantage in this domain.** Distribution of the per-scenario speedup ratio $t_{\text{NAMOA*}}/t_{\text{Lazy}}$, restricted to scenarios with $t_{\text{NAMOA*}} > 50$ ms; values above 1 favour LazyLTMOA*.



Source: Prepared by the author (2026).

Crucially, the speedup does not come from exploring fewer states. NAMOA* and LazyLTMOA* expand and generate *exactly* the same number of labels in all 945 completed scenarios: the expanded-label counts are identical in every row of the result set, as expected from two exact algorithms using the same heuristic and the same lexicographic expansion order. The lazy variant therefore traverses the same search graph; its advantage is purely a smaller constant per expansion, achieved by deferring full set-dominance checks and replacing the per-vertex $G_{\text{op}}/G_{\text{cl}}$ structures with a flat array and a $\mathbf{g}_{\min}$ summary.

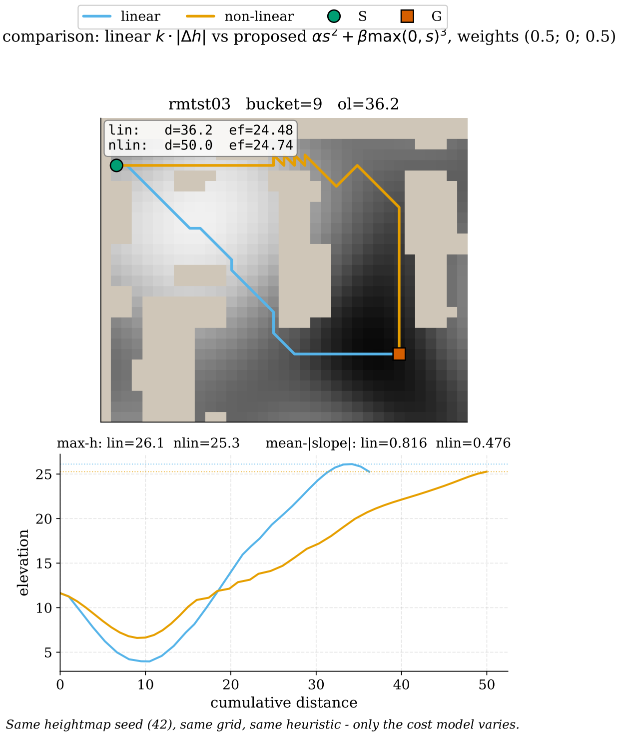
Interpretation. This refines the qualitative claim of Hernández *et al.* (2023a): lazy dominance checks reduce per-expansion work, but in this domain they do not reduce the number of expansions, and the resulting benefit is more modest than the order-of-magnitude gains reported on road-network benchmarks. Two factors plausibly explain the smaller margin. The DAO maps are smaller than typical road networks (a 1024×1024 map has on the order of 10^6 cells against 10^7–10^8 road-network vertices), and the Ideal Point heuristic already keeps OPEN small, so the lazy checks operate on a smaller surface. The practical implication is that, since coverage is identical (Claim B), LazyLTMOA* is a strict drop-in improvement over NAMOA* whenever exact search is feasible at all, but it does not extend the range of solvable scenarios.

4.4 Claim D: The Non-Linear Effort Model Produces Qualitatively Distinct Paths

Statement. The proposed non-linear effort function (2) does not merely re-scale the linear baseline; it produces geometrically distinct optimal paths, most visibly in mountainous terrain.

Evidence. For three scenarios spanning the difficulty spectrum, Dijkstra-WS was run twice on the same scenario, grid, heightmap, and heuristic, with the weight vector $(0.5, 0, 0.5)$ (distance plus effort, danger neutralized), varying only the effort function between the linear baseline $c_{\text{linear}}(s) = |s|$ and the proposed $c_{\text{nonlin}}(s) = s^2 + 2\max(0, s)^3$. The scenarios are one mountainous map (`rmtst03`), one mixed map (`arena`), and one mostly flat city map (`den200d`); their metrics are summarized in Table 3. The effect is largest on `rmtst03` (Figure 6): the non-linear path runs 38% longer in distance (49.9 versus 36.2) but cuts the mean absolute slope by 41.7% (0.476 versus 0.816) and slightly lowers the peak altitude, routing along contour lines instead of climbing straight across the mountainside. On the mixed `arena` map the non-linear path is 12% longer yet reduces both mean slope (by 26%) and effort cost (12.49 versus 13.19), a strictly better effort solution at modest distance cost. On the flat `den200d` map the two paths take the same geometric shape, and the non-linear version only makes finer local adjustments that lower effort by 33% (10.47 versus 15.70) at zero distance cost, the expected convergence on terrain with little slope to exploit.

Figure 6 – **On mountainous terrain the non-linear model trades 38% more distance for a 41.7% lower mean slope, routing around the steep face that the linear path climbs directly, a qualitatively different navigational choice.** Scenario `rmtst03` bucket 9. Top: optimal paths under the linear (blue) and non-linear (orange) effort models over the grayscale heightmap (darker = higher). Bottom: elevation profiles along each path. Same seed, grid, and heuristic; only the effort function varies.



Source: Prepared by the author (2026).

Table 3 – **The non-linear model reduces mean slope where it matters most (mountainous `rmtst03`) and is geometrically equivalent to the linear baseline on the flat `den200d`.** Per-path metrics across three scenarios; boldface marks the better value on each row.

Map	Bucket	Distance		Mean $\lvert slope \rvert$	
		linear	non-linear	linear	non-linear
`rmtst03`	9	**36.2**	49.9	0.816	**0.476**
`arena`	8	**34.3**	38.5	0.426	**0.316**
`den200d`	9	38.3	38.3	**0.449**	0.457

Source: Prepared by the author (2026).

Interpretation. The comparison validates the modeling claim. The linear baseline takes the geometrically short route across mountainsides and accumulates high local slope, whereas the non-linear model trades distance for terrain difficulty the way a human navigator would: following contours, avoiding the steepest faces, and accepting a longer route when the slope reduction is substantial. The benefit concentrates where it matters, on the mountainous map, and vanishes on flat terrain, as expected. The effect is achieved at negligible cost: the comparison runs in a few milliseconds per scenario, so the non-linear function is fully tractable for real-time use in production engines.

The two methodological axes of this work, the scalarization choice and the cost-model choice, are favoured by complementary kinds of terrain. Weighted-sum search is adequate precisely where the frontier is close to convex, which the data associate with flat or uniform maps: there the unsupported ratio is low (the ten maps below a 0.5 median in Claim A), few optimal trade-offs are hidden, and a single scalar query lands near the frontier. The non-linear effort model, by contrast, changes the route only where there is slope to exploit, so its effect grows with terrain ruggedness, the same mountainous maps where the unsupported ratio is highest and exact search is most needed. The two contributions therefore reinforce each other on rugged terrain and both become unnecessary on flat terrain, which is itself an actionable guideline: the steeper and more varied the map, the more both a non-linear cost and an exact solver are worth their cost.

5 CONCLUSION AND FUTURE WORK

This work investigated multi-objective pathfinding in digital games with altimetric variation, combining a biomechanically grounded non-linear effort cost model with state-of-the-art exact search (NAMOA* and LazyLTMOA*) on the standardized *Dragon Age: Origins* benchmark. In the language of AIIDE's New Grounds theme, it carries an established AIIDE topic, pathfinding, onto the new ground of biomechanically realistic terrain cost.

The empirical results, drawn from 3,120 scenarios across 156 maps, support four conclusions. Weighted-sum scalarization, still the dominant production approach, structurally hides the majority of Pareto-optimal solutions: the unsupported ratio reaches 99.88% in extreme scenarios and exceeds 90% in median on 107 of 143 maps (Claim A). Exact frontier computation is tractable only for short scenarios, with completion rates collapsing beyond an optimal length of about 25 cells under an interactive budget (Claim B). LazyLTMOA* gives a consistent but modest speedup over NAMOA* (88.4% of non-trivial scenarios faster, median 1.08×) that comes entirely from cheaper expansions rather than fewer, since the two algorithms explore the graph identically and return identical frontiers (Claim C). And the proposed non-linear effort function produces geometrically distinct paths from the linear baseline, cutting the mean absolute slope along the chosen route by 41.7% in mountainous terrain (Claim D). Taken together, these findings reframe algorithm selection in multi-objective game pathfinding as a *map-conditional* and *path-length-conditional* decision, with the non-linear cost model offering a cheap, algorithm-independent improvement in either regime.

Limitations and Future Work

The boundaries of the present study map directly onto the next steps. The terrain is procedurally generated by Perlin noise: this isolates the effect of the cost model and guarantees reproducibility, but procedural noise does not reproduce the choke-points, vistas, and traversal narratives of hand-authored terrain, nor the correlation that real games impose between danger and elevation (here generated independently). Replicating Claim D on heightmaps extracted from commercial titles such as *Skyrim* or *Breath of the Wild*, with their gameplay-driven structural correlations, is the most direct way to strengthen its external validity. Likewise, the 20 scenarios per map were drawn by stratified sampling on optimal length for uniform difficulty coverage; gameplay-informed sampling, reflecting the source-goal pairs that actually arise in play, would refine the coverage analysis. The effort parameters $\alpha = 1$, $\beta = 2$ were chosen to emphasize the cubic asymmetry rather than calibrated against human-trial data; since the frontier is invariant under positive affine transformations of any single component, only their ratio affects the comparison, but a calibrated ratio would sharpen the absolute interpretation of effort, and a systematic sensitivity analysis of the $\alpha{:}\beta$ ratio would quantify how the chosen paths shift as the balance between the quadratic and cubic terms changes. Two further methodological extensions follow from the committee discussion. First, the five weight configurations used here locate the weighted-sum solutions relative to the frontier but do not sample the simplex densely; a fine grid sweep over the weight space (for example, thousands of uniformly distributed weight vectors) would confirm that the unsupported solutions remain unreachable for every weight rather than only these five, strengthening Claim A. Second, the terrain itself could be stress-tested with deliberately extreme features such as cliffs and plateaus, probing the effort model at the boundary of Minetti's measured slope range.

The scalability ceiling of Claim B is the richest research frontier. It motivates approximate and bounded-suboptimal multi-objective search, such as ε-Pareto approximation (Papadimitriou; Yannakakis, 2000) or anytime variants of NAMOA*/LazyLTMOA* that return progressively refined frontiers under time pressure, and it motivates hierarchical decomposition that runs exact multi-objective search at coarse granularity (between waypoints on a region graph) while using fast single-objective search within regions, recovering large-scale tractability while preserving the multi-objective view where it matters. A complementary open problem, distinct from computing the frontier, is *selecting* a single path from it: a frontier of several thousand solutions is not directly actionable, and design-time use needs a principled way to reduce it. Promising directions include ε-dominance pruning to a bounded, evenly spread representative set whose size the designer controls, and using a weighted-sum criterion inside the exact search as a ranking rule that steers it toward a designer-specified region of the frontier without losing exactness; comparing these selection strategies on game-design criteria is a natural next step. Two further extensions stand out: additional objectives relevant to games, such as visibility for stealth, cover for projectile avoidance, and accumulated fatigue on long traversals; and any-angle multi-objective search, combining the path-smoothing ideas of Theta* (Nash *et al.*, 2007) and ANYA (Harabor; Grastien, 2013) with the Pareto machinery of NAMOA*/LazyLTMOA* to remove the grid-induced jaggedness and length inflation and yield paths fit for direct use as NPC trajectories.

REFERENCES

Das, I.; Dennis, J. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. **Structural Optimization**, v. 14, n. 1, p. 63–69, 1997. Cited 2 times on page 2 and 3.

Dijkstra, E. A note on two problems in connexion with graphs. **Numerische Mathematik**, Springer, v. 1, n. 1, p. 269–271, 1959. Cited 2 times on page 2 and 3.

Harabor, D.; Grastien, A. An optimal any-angle pathfinding algorithm. In: **Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS)**. [S.l.]: AAAI Press, 2013. p. 308–311. Cited on page 17.

Hart, P.; Nilsson, N.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE Transactions on Systems Science and Cybernetics**, v. 4, n. 2, p. 100–107, 1968. Cited 2 times on page 2 and 3.

Hernández, C.; Yeoh, W.; Baier, J.; Felner, A.; Salzman, O.; Asai, H.; Zhang, H.; Koenig, S. Multi-objective search via lazy and efficient dominance checks. In: **Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.: s.n.], 2023. p. 5638–5646. Cited 5 times on page 2, 3, 4, 7, and 14.

Hernández, C.; Yeoh, W.; Baier, J.; Felner, A.; Salzman, O.; Asai, H.; Zhang, H.; Koenig, S. Simple and efficient bi-objective search algorithms via fast dominance checks. **Artificial Intelligence**, v. 314, p. 103807, 2023. Cited on page 3.

Mandow, L.; Pérez-de-la-Cruz, J. A new approach to multiobjective A* search. In: **Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.: s.n.], 2005. p. 218–223. Cited 3 times on page 2, 3, and 7.

Mandow, L.; Pérez-de-la-Cruz, J. Multiobjective A* search with consistent heuristics. **Journal of the ACM**, v. 57, n. 5, p. 1–25, 2010. Cited 3 times on page 2, 3, and 7.

Minetti, A.; Moia, C.; Roi, G.; Susta, D.; Ferretti, G. Energy cost of walking and running at extreme uphill and downhill slopes. **Journal of Applied Physiology**, v. 93, n. 3, p. 1039–1046, 2002. Cited 2 times on page 2 and 5.

Nash, A.; Daniel, K.; Koenig, S.; Felner, A. Theta*: Any-angle path planning on grids. In: **Proceedings of the 22nd AAAI Conference on Artificial Intelligence**. [S.l.]: AAAI Press, 2007. p. 1177–1183. Cited on page 17.

Papadimitriou, C.; Yannakakis, M. On the approximability of trade-offs and optimal access of web sources. In: **Proceedings of the 41st Annual Symposium on Foundations of Computer Science (FOCS)**. [S.l.]: IEEE, 2000. p. 86–92. Cited on page 17.

Pulido, F.; Mandow, L.; Pérez-de-la-Cruz, J. Dimensionality reduction in multiobjective shortest path search. **Computers & Operations Research**, v. 64, p. 60–70, 2015. Cited on page 3.

Rees, W. Least-cost paths in mountainous terrain. **Computers & Geosciences**, v. 30, n. 3, p. 203–209, 2004. Cited on page 5.

Ren, Z.; Rathinam, S.; Choset, H. Enhanced multi-objective A* using balanced binary search trees. In: **Proceedings of the 15th International Symposium on Combinatorial Search (SoCS)**. [S.l.]: AAAI Press, 2022. p. 162–170. Cited on page 3.

Stewart, B.; White, C. Multiobjective A*. **Journal of the ACM**, v. 38, n. 4, p. 775–814, 1991. Cited on page 3.

Sturtevant, N. Benchmarks for grid-based pathfinding. **IEEE Transactions on Computational Intelligence and AI in Games**, v. 4, n. 2, p. 144–148, 2012. Cited on page 5.

Tobler, W. **Three presentations on geographical analysis and modeling**. [S.l.], 1993. Cited on page 5.

FOLHA DE APROVAÇÃO DE TCC Nº 19 / 2026 - DECOM (11.56.03)

Nº do Protocolo: 23062.032850/2026-95

Belo Horizonte-MG, 24 de junho de 2026.

PAULO RENATO SOUZA MAGALHÃES

BEYOND THE SHORTEST PATH: A non-linear altimetric effort model for multi-objective pathfinding in digital games

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Computação do Centro Federal de Educação Tecnológica de Minas Gerais, do Campus Nova Gameleira, como parte do requisito para obtenção do grau de Bacharel em Engenharia de Computação.

Aprovado em 18 de junho de 2026.

Banca Examinadora:

Prof. Doutor Mateus Felipe Tymburibá Ferreira - Orientador

Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Doutor André Rodrigues da Cruz

Centro Federal de Educação Tecnológica de Minas Gerais

Prof.ª Doutora Elizabeth Fialho Wanner

Centro Federal de Educação Tecnológica de Minas Gerais

Belo Horizonte

2026

(Assinado digitalmente em 24/06/2026 23:37)
ANDRE RODRIGUES DA CRUZ
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 2683746

(Assinado digitalmente em 26/06/2026 06:40)
ELIZABETH FIALHO WANNER
PROFESSOR DO MAGISTERIO SUPERIOR
DECOM (11.56.03)
Matrícula: 1539394

(Assinado digitalmente em 24/06/2026 19:00)
MATEUS FELIPE TYMBURIBA FERREIRA
PROFESSOR ENS BASICO TECN TECNOLOGICO
DECOM (11.56.03)
Matrícula: 2277448

Visualize o documento original em **https://sig.cefetmg.br/public/documentos/index.jsp** informando seu número: **19**, ano: **2026**, tipo: **FOLHA DE APROVAÇÃO DE TCC**, data de emissão: **24/06/2026** e o código de verificação: **c5fd17d112**