57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

Comparative Optimization of Electric Vehicle Routes Using Graph Algorithms and Mathematical Programming

Enzo Reis de Oliveira
Universidade Federal de São Paulo (UNIFESP)

Luiz Leduíno de Salles Neto
Universidade Federal de São Paulo (UNIFESP)

Abstract

We present an integrated approach for optimizing routes for electric vehicles (EVs) using three methods: A* search, Dijkstra's algorithm, and a mixed-integer programming (MIP) model. A unified cost framework combines travel time, charging time, and distance under nine weighting scenarios. On a corridor between São Paulo and Florianópolis with stations from OpenChargeMap and a 400 km nominal autonomy, Dijkstra and MIP consistently attain scenario-optimal routes, while A* is competitive on single-metric cases and slightly inferior (3–5 min) in mixed objectives due to charging choices. Dijkstra solves each case in about 0.2 s versus ∼8 s for MIP on commodity hardware, highlighting a strong accuracy–speed trade-off for real-time navigation.

Keywords: electric vehicle routing; A*; Dijkstra; mixed-integer programming; multi-criteria costs.

1 Introduction

The rapid adoption of EVs has renewed interest in routing that balances driving and charging, consistent with recent growth reported by the International Energy Agency [2]. Practical planning is challenging because: (i) limited battery range; (ii) charging times that vary with station power; and (iii) uneven infrastructure. Classical works incorporate energy and recharging into routing [3, 5]; multi-criteria formulations consider distance, time and energy [7]. Planners such as ABRP and PlugShare operationalize these ideas [1, 11]. We study the single-vehicle O–D case and compare exact and algorithmic solvers under a common cost model.

Contributions. (1) A directed graph with charging stations as nodes and feasible legs as arcs; (2) nine cost scenarios mixing distance (D), driving time (T_{drive}) and charging time (T_{chg}); (3) a head-to-head of A*, Dijkstra and MIP across all scenarios; (4) analysis of quality vs. runtime for deployment.

https://proceedings.science/p/212539?lang=pt-br

Acknowledgments

We thank UNIFESP colleagues for feedback and OpenChargeMap for data. Enzo acknowledges guidance from Luiz Leduíno de Salles Neto.

2 Model and Cost Scenarios

Let $G = (N, A)$ be a directed graph with origin s and destination t. We add arc (i, j) if geodesic distance $d_{ij} \le R$, storing per-arc $(D_{ij}, T_{ij}^{\text{drive}}, T_{ij}^{\text{chg}})$. The cost is

$$c_{ij} = \alpha T_{ij}^{\text{drive}} + \beta T_{ij}^{\text{chg}} + \gamma D_{ij}, \qquad \alpha, \beta, \gamma \ge 0,\ \alpha + \beta + \gamma = 1. \tag{1}$$

We evaluate nine scenarios: (1) T_{drive}, (2) T_{chg}, (3) D, (4) $0.5T_{\text{drive}}$+$0.5T_{\text{chg}}$, (5) $0.5T_{\text{drive}}$+$0.5D$, (6) $0.5T_{\text{chg}}$+$0.5D$, (7) $0.4T_{\text{drive}}$+$0.4T_{\text{chg}}$+$0.2D$, (8) $0.4T_{\text{drive}}$+$0.4D$+$0.2T_{\text{chg}}$, (9) $0.4T_{\text{chg}}$+$0.4D$+$0.2T_{\text{drive}}$. More realistic battery/infrastructure layers can be integrated (e.g., SOC-dependent charging and siting [10, 12]), but are beyond the present scope.

3 Methods

A*. Best-first search with $f = g + h$, using straight-line distance scaled by average speed (for T_{drive}) or by a power-derived rate (for T_{chg}); h is admissible/consistent [6]. **Dijkstra.** A* with $h \equiv 0$; label-setting with a min-queue, optimal for nonnegative arcs [4]. **Exact MIP.** Binary $x_{ij} \in \{0, 1\}$, flow conservation and objective min $\sum c_{ij}x_{ij}$ return the exact shortest s–t path; solved with CBC. See Papadimitriou and Steiglitz [9] for background.

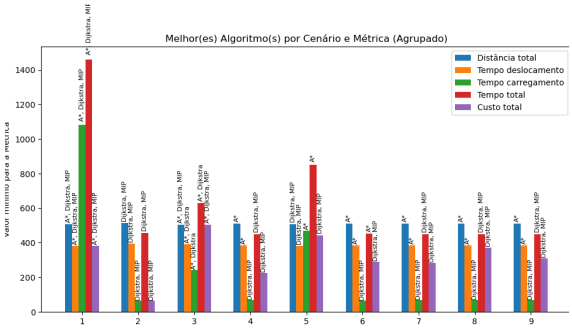
4 Experimental Setup

Region. São Paulo (−23.5505, −46.6333) to Florianópolis (−27.5954, −48.5480); stations from OpenChargeMap [8]; autonomy $R = 400$ km. *Implementation.* Python/NetworkX, Geopy, and PuLP+CBC. We log D, T_{drive}, T_{chg}, total time and objective for each scenario/method.

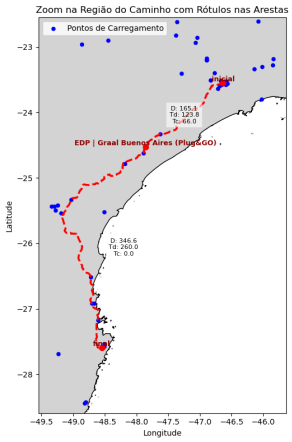
5 Results

Across the nine scenarios, Dijkstra and MIP always recover the scenario-optimal objective. A* matches the optimum for Scen. 1 and 3 but trails by ~3–5 min when objectives mix components due to less aggressive detours to higher-power chargers. Runtime-wise, Dijkstra (≈0.2 s/instance on an M3 laptop) is ~40× faster than MIP (≈8 s) while achieving the same objective; A* depends on heuristic quality.

Visual assessment. Figure 1b shows the preference for high-power chargers even with mild detours, lowering T_{chg} with little impact on D. Figure 1a confirms that Dijkstra and MIP tie on the optimal objective across scenarios; A* is competitive when a single component dominates or the heuristic is informative.



(a) Best algorithm(s) per scenario and metric.



(b) Route zoom with edge labels (D, T_d, T_c).

6 Discussion

Sensitivity. Sweeping (α, β, γ) yields piecewise-constant solutions: routes change only at few breakpoints where two paths tie, as the shortest-path value is the lower envelope of linear functions. Lowering R to 300 km increases hop count and T_{chg} by roughly 15–25%; raising R to 500 km enables skipping clusters, cutting total time by 6–9% in mixed scenarios. Random removal of 10–20% of stations worsens total time by 4–7% but preserves connectivity along BR-116/BR-101, indicating robustness to gaps in [8].

Scalability and deployment. Dijkstra/A* run in $O(|A| \log |N|)$ and memory $O(|N|+|A|)$ [4, 6]; the ILP has $|A|$ binaries and $|N|$ flow constraints, with nontrivial presolve overheads [9]. Practical tips: enforce a reserve margin ($d_{ij} \leq (1-\rho)R$, $\rho \simeq 10\%$), throttle degrees at dense hubs, and (optionally) z-score $(T_{\text{drive}}, T_{\text{chg}}, D)$ to reduce scaling artifacts. For online navigation, use Dijkstra with the chosen (α, β, γ); for business rules (mandatory stops, time windows, dwell caps) or certification, encode constraints and validate with MIP. Enrich data with SOC-aware charging curves [10] and periodic refresh from [8].

Runtime profile and engineering tips. On corridor-sized graphs (order of 10^3 nodes / 10^4 arcs), Dijkstra's priority-queue operations dominate time but remain sub-second on commodity laptops; A* can further reduce node expansions when h tracks the dominant component. Warm-starting MIP with a Dijkstra route provides an incumbent that often shortens branch-and-bound. Time limits plus acceptance of the best incumbent yield valid routes when exact optimality is unnecessary. Caching arc attributes and precomputing feasible-neighborhood lists (per node) reduce Python overhead. For repeated queries with varying (α, β, γ), reuse the same graph and swap the cost accessor to avoid rebuilds.

Reproducibility and data management. Make runs deterministic by fixing the OpenChargeMap snapshot date/time and persisting the processed node-link JSON along with a YAML of $(R, \kappa, \eta, \tau, \bar{v})$ and (α, β, γ). Log commit hashes and solver versions (CBC/PuLP) with seeds. The pipeline is modular: updating T_{drive} from geodesic/speed to OSRM travel times or replacing the charge proxy

by SOC curves [10] only changes arc attributes, leaving solvers intact. Periodic metadata refresh (station power/status) from [8] keeps recommendations aligned with reality.

7 Conclusion and Perspectives

We proposed a compact EV-routing framework and compared A*, Dijkstra, and exact MIP under nine cost mixes. On realistic data, Dijkstra and MIP coincide in solution quality; MIP pays a significant runtime premium. Future work: (a) OSRM/Maps travel times, (b) SOC-dependent charging [10], (c) dynamic re-routing by live availability and tariffs, and (d) sensitivity to R and charger density; see also multi-criteria and siting directions in Montoya et al. [7], Zhang and Wang [12].

References

[1] A Better Routeplanner (2025). A better routeplanner: Online ev route planning. `https://abetterrouteplanner.com`.
[2] Agency, I. E. (2021). Global ev outlook 2021. Technical report, IEA.
[3] Campbell, A. M. and Savelsbergh, M. W. P. (2010). The electric vehicle routing problem. *Transportation Science*, 44(2):318–329.
[4] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1: 269–271.
[5] Erdogan, S. and Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114.
[6] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. In *IEEE Transactions on Systems Science and Cybernetics*, volume 4, p. 100–107.
[7] Montoya, A., Morales, J., and Cappanera, P. (2017). Multi-criteria routing in electric vehicle networks. p. 1–6.
[8] OpenChargeMap (2023). Open charge map api documentation. `https://openchargemap.org/site`.
[9] Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall.
[10] Plett, G. L. (2004). Extended kalman filtering for battery management systems of li-ion battery packs: Part i. background. *Journal of Power Sources*, 134(2):252–261.
[11] PlugShare (2025). Plugshare: Ev charging station map and navigation. `https://www.plugshare.com`.
[12] Zhang, R. and Wang, S. (2014). Electric vehicle charging station placement. *IEEE Transactions on Smart Grid*, 5(1):162–170.