Optimal Speed Determination

1 Introduction

Choosing an appropriate driving speed involves balancing several competing factors. Driving too fast increases fuel consumption because aerodynamic drag grows rapidly at high speeds. Driving too slowly causes long travel times and may also waste fuel due to idling and low-efficiency operation. Additional elements such as engine capacity, traffic congestion, and passenger load also influence the ideal speed.

In this project, we build a mathematical model that represents these real-world behaviours and then apply numerical optimisation methods (Steepest Descent, Newton's Method, BFGS and a Trust-Region method) to compute the speed that minimises a combined cost. The cost includes fuel usage, travel time, and a soft behavioural penalty that keeps the chosen speed close to a realistic, human-preferred range.

2 Mathematical Formulation

We optimise a single variable, the vehicle speed

$$v \in [v_{\min}, v_{\max}],$$

where $v_{\min}$ and $v_{\max}$ are lower and upper speed limits. All other quantities (traffic index, engine capacity, number of passengers, distance etc.) are treated as inputs.

2.1 Baseline Fuel Consumption Model

Fuel consumption per kilometre is modelled by a quadratic function:

$$FC_{\text{baseline}}(v) = K + c_{e1}v + c_{e2}v^2. \tag{1}$$

Here:

- K represents constant engine losses such as idling and friction,
- $c_{e1}v$ models rolling resistance and light engine load,
- $c_{e2}v^2$ captures aerodynamic drag, which becomes dominant at higher speeds.

This form produces the familiar U-shaped fuel-efficiency curve.

2.2 Engine Capacity Adjustment

Different engine capacities have different efficiency profiles. To incorporate this, the fuel coefficients are scaled using

$$\text{efficiency scale} = \left(\frac{C_{\text{ref}}}{C}\right)^{\beta_{\text{eff}}}, \tag{2}$$

where C is the engine capacity and C_{ref} is a reference capacity. The effective coefficients become

$$c_{e1}^{\text{eff}} = c_{e1}\left(\frac{C_{\text{ref}}}{C}\right)^{\beta_{\text{eff}}}, \qquad c_{e2}^{\text{eff}} = c_{e2}\left(\frac{C_{\text{ref}}}{C}\right)^{\beta_{\text{eff}}}. \tag{3}$$

This makes smaller engines consume relatively more fuel at the same speed, while larger engines consume slightly less.

2.3 Effect of Weight and Traffic on K

Passenger load and engine size also influence the baseline loss term K:

$$K = c_0 + c_3\tau + c_4(nw_{\text{avg}}) + \frac{c_5}{C}, \tag{4}$$

where τ is the traffic index, n is the number of passengers, and w_{avg} is the average passenger weight. Heavier traffic and heavier load both increase fuel consumption, and smaller engines are more strained under the same load.

2.4 Time and Traffic Delay Cost

Travel time and congestion contribute to an additional cost:

$$\text{TimeCost}(v) = C_t\left(\frac{D}{v} + s\tau D\right), \tag{5}$$

where D is the trip distance, C_t is a time-cost coefficient, s is a constant that scales the traffic delay, and τ is again the traffic index. The first term penalises very low speeds, while the second term increases cost in heavy traffic regardless of speed.

2.5 Preferred Speed Model

Drivers generally have a comfortable travel speed depending on conditions. We capture this using

$$v_{\text{pref}} = V_{\max} - c_1\tau - \frac{c_2}{C} - c_3(nw_{\text{avg}}), \tag{6}$$

which is then clipped to stay inside $[v_{\min}, v_{\max}]$. Here:

- higher traffic (τ) reduces the preferred speed,
- smaller engine capacity C reduces the preferred speed,
- a higher number of passengers n also reduces the preferred speed.

2.6 Regularisation Toward Preferred Speed

To avoid unrealistic extreme speeds (too slow or too fast), we add a soft penalty:

$$\lambda(v - v_{\text{pref}})^2, \tag{7}$$

where $\lambda > 0$ controls how strongly the solution is pulled towards v_{pref}. This term ensures that the optimal speed remains human-like without dominating the main fuel and time terms.

2.7 Final Objective Function

Putting everything together, the total cost function to be minimised is

$$f(v) = B\big(K + c_{e1}v + c_{e2}v^2\big) + C_t\frac{D}{v} + C_t s\tau D + \lambda(v - v_{\text{pref}})^2, \tag{8}$$

where B is a fuel-price scaling factor. This function combines fuel cost, time cost, traffic delay, and the preferred-speed regularisation.

2.8 Convexity

Every term in the function is convex for $v > 0$, meaning the whole function is convex. So there is always one unique global minimum, and gradient-based methods converge very reliably.

3 Implementation and Solvers

The program takes user inputs such as distance D, traffic index τ, engine capacity C, and number of passengers n. Using these values, it constructs the cost function $f(v)$ defined above.

Each optimisation method (Steepest Descent, Newton's Method, BFGS and Trust-Region) follows the same high-level loop:

1. Evaluate the gradient $f'(v)$, and for Newton's method also the second derivative $f''(v)$.
2. Compute a search direction based on the method.
3. Use a line search or a trust-region rule to choose a step length.

4. Update the current speed estimate v.

The iteration stops when the gradient magnitude becomes sufficiently small. The program then outputs the optimal speed, the predicted fuel usage for the trip, and the total cost.

4 Comparison of Methods

Because the objective function is convex, all four methods converge to the same optimal speed for a given set of inputs. The difference is in how quickly and robustly they converge.

Newton's Method converges the fastest, typically in about 3–6 iterations, because it uses second-derivative information. The BFGS method, which approximates the Hessian instead of computing it exactly, achieves similar performance, usually in 2–5 iterations after a good approximation builds up.

Steepest Descent uses only first-order information and therefore takes more iterations (around 10–15) to reach the same accuracy. The Trust-Region method is the most robust when the initial guess is poor or when the model behaves unexpectedly. It may take slightly more iterations than Newton but handles difficult starting points better.

5 Future Improvements

There are several natural extensions to this project:

- Learn the preferred-speed model v_{pref} directly from real driving data instead of using a rule-based expression.
- Fit the fuel coefficients c_{e1} and c_{e2} using real mileage logs to obtain engine-specific or bike-specific models.
- Add road-type information, for example separate models for highways, dense city roads, and ghat sections.
- Use an adaptive regularisation weight $\lambda(\tau)$ that becomes stronger in heavy traffic.
- Extend the framework to multi-objective optimisation, allowing the user to choose between fuel-saving, time-saving, or comfort-focused modes.
- Integrate GPS data and perform real-time optimisation to suggest an optimal speed profile $v^*(x(t))$ along the route.

6 Conclusion

This project shows how fuel consumption, traffic delay, engine characteristics, and human driving preferences can be combined into a single smooth optimisation problem for choosing an ideal driving speed. The final objective function is convex, which guarantees a unique global minimum and allows numerical methods like Newton's method and BFGS to converge quickly and reliably.

The resulting model produces realistic and practical speed recommendations and provides a solid base for further work, such as learning parameters from real data and supporting multi-objective, real-time optimisation in driver assistance systems.