ME 498: Multi-Objective Sizing Optimization of the Mercedes-Benz OM606 Connecting Rod via 1D Beam Modeling

Advait Chordia

***Abstract*— This study presents a multi-objective sizing optimization of the Mercedes-Benz OM606 connecting rod to minimize reciprocating mass while ensuring structural integrity under peak gas and inertial loads. A 1D non-uniform beam element model was developed and solved using Sequential Least Squares Programming (SLSQP), allowing for rapid iteration of the shank's cross-sectional geometry. The optimization achieved a final shank mass of 88.34 g, a significant reduction in parasitic inertia. Contrary to initial assumptions of buckling dominance, the post-optimization analysis reveals that the design is constrained primarily by Static Yield under compressive gas loading (Active Constraint, SF=1.3) and Fatigue Life under tensile inertial loading (SF=1.8). Global buckling stability was found to be non-critical with a Safety Factor of 7.3, indicating that the optimized I-beam topology provides exceptional stiffness-to-weight performance. A comparative study with an H-beam topology further validated the I-beam's superiority, as the H-beam incurred a 17.5% mass penalty to achieve comparable structural performance within the engine's packaging constraints.**

I. Introduction

The Mercedes-Benz OM606 is a 3.0-liter inline-six diesel engine renowned for its robust construction. However, its stock connecting rods are over-engineered and heavy, presenting a significant source of parasitic loss in high-performance applications. This project focuses on the shape optimization of the naturally aspirated OM606.910 connecting rod to minimize reciprocating mass while maintaining structural integrity.

II. Methodology

A. System Idealization and Discretization

A 1D non-uniform beam formulation was selected over 3D topology optimization to prioritize computational efficiency and direct manufacturability. While 3D methods identify novel load paths, they are computationally expensive and often prone to artificial modes in buckling-dominated problems [5]. The 1D approach enables rapid evaluation of the Area Moment of Inertia (I), the governing parameter for buckling stability, while producing parametric dimensions (web height, flange width) that are immediately compatible with forging constraints.

B. Load Cases and Boundary Conditions

Three distinct critical loading scenarios were identified based on the engine's operating cycle to ensure the rod withstands all mechanical stresses.

The first case is the Peak Gas Load (Compression). This occurs during the power stroke when combustion pressure peaks. Based on a peak cylinder pressure of 130 bar and a bore of 87 mm, the gas force is calculated as:

$$F_{gas} = P_{combustion} \cdot A_{piston} \approx 77.3\ kN$$

This compressive load drives the yield (crushing) and buckling (stability) constraints. A Factor of Safety (SF) of 1.3 was selected for yield, while a higher factor of 2.5 was chosen for buckling. This higher margin is consistent with recommendations for connecting rods in Shigley's Mechanical Engineering Design [6], reflecting the catastrophic nature of stability failure and its sensitivity to geometric imperfections and dynamic shock loads. To satisfy this safety factor, the rod must resist a critical buckling load governed by the Euler formula:

$$P_{cr} = \frac{\pi^2 EI}{L^2}$$

In this optimization problem, the Young's Modulus (E) is fixed by the material selection, and the effective length (L) is rigidly constrained by the engine stroke and block geometry. Consequently, the only variable available to increase P_{cr} and achieve the required 2.5 safety factor is the Area Moment of Inertia (I). This dictates that the optimization strategy must focus on manipulating the cross-sectional geometry to maximize I rather than simply increasing the cross-sectional area, which would unnecessarily increase mass.

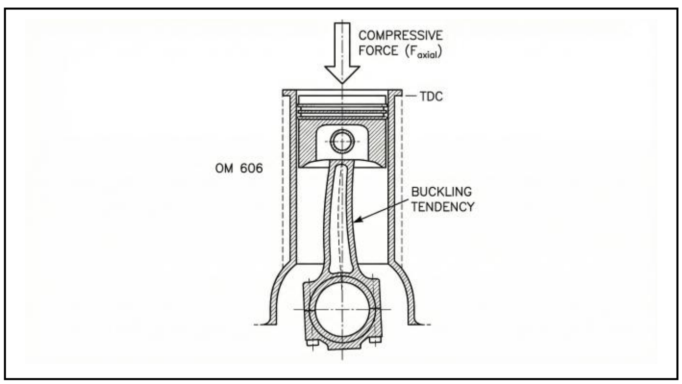

Fig. 1. Conrod loading position for buckling

The second case is Top Dead Center Inertia (Tension). This load occurs at the end of the exhaust stroke (TDC) when the piston changes direction, forcing the crankshaft to pull the piston down against its own inertia. The inertial force is defined by:

$$F_{inertia} = m_{recip} \cdot r\omega^2(1 + \frac{r}{L})$$

In this expression, m_{recip} denotes the reciprocating mass (approximately 1.5kg) which includes the piston assembly and the small end of the rod. The terms r and L represent the crank radius and connecting rod length, respectively. The angular velocity ω was set to correspond to 5500 RPM. This specific speed represents the engine's redline limit, which was selected as the design point because inertial forces scale with the square of the engine speed (ω^2). Designing for red line RPMs ensures the rod is robust against fatigue failure under the most extreme tensile loading conditions encountered during operation.

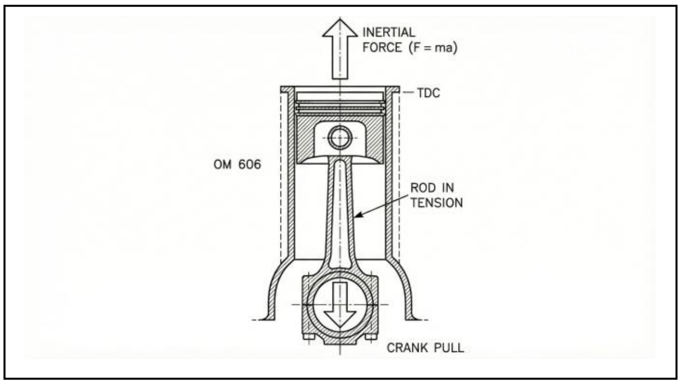

Fig. 2. Conrod loading position for inertial tension

The third case is Transverse Inertia (Whipping). This dynamic load occurs when the crank angle is approximately 90°. The mass of the rod resists the lateral acceleration imposed by the crankshaft, causing it to bow outward. This was modeled as a linearly distributed lateral load $q(x)$ along the beam element:

$$q(x) = \rho A(x) \cdot a_{transverse}(x)$$

Here the transverse acceleration $a_{transverse}$ varies linearly from zero at the pin to $\omega^2 r$ at the crank.

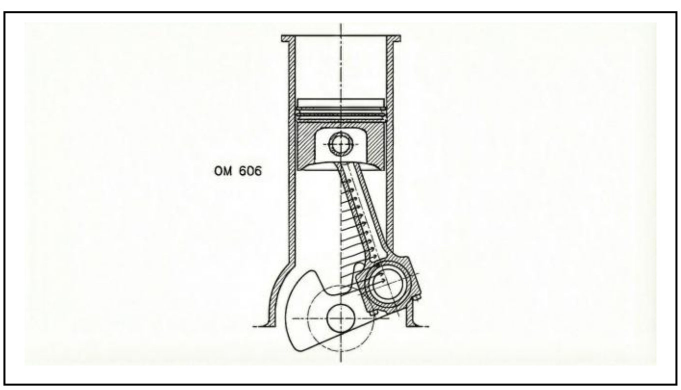

Fig. 3. Conrod loading position for bending at max crank angle

C. Design Parametrization

The shank geometry was parameterized using 44 design variables to allow for local shape variation while maintaining manufacturability. Forty discrete variables were used to define the height (h_i) and width (w_i) of the I-beam cross-section independently at each of the 20 finite elements. To ensure the rod remains forgeable, the web thickness (t_w) and flange thickness (t_f) were not allowed to vary element-by-element. Instead, they were treated as four continuum variables defined by a linear interpolation function controlled by start and end thicknesses. This acts as an implicit geometric constraint, preventing undercut geometries that would be impossible to remove from a forging die.

D. Finite Element Formulation

The structural response was computed using the Direct Stiffness Method. The elastic stiffness matrix (K) was assembled using 2D Bernoulli-Euler beam elements with variable moments of inertia (I) and areas (A). To capture the stress-stiffening or softening effects of the compressive load, a Geometric Stiffness Matrix (K_g) was derived. The critical buckling load P_{cr} was found by solving the generalized eigenvalue problem:

$$(K - \lambda K_g)\phi = 0$$

In this formulation, the smallest positive eigenvalue λ represents the buckling load factor. Material properties were derived from AISI 4340 Steel (Quenched & Tempered), with a Young's Modulus of 205 GPa and a Yield Strength of 1100 MPa.

E. Optimization Statement

The problem was formulated as a constrained non-linear programming (NLP) problem. The objective function $f(x)$ is the total mass of the shank, defined as the summation of the mass of individual elements:

$$Minimize: f(x) = \Sigma \rho A_i L_i$$

Subject to the following constraints:

1. Buckling Stability: The ratio of the critical buckling load to the applied gas load, scaled by the safety factor, must be greater than unity.
$$\frac{P_{cr}}{F_{gas} \cdot 2.5} - 1 \geq 0$$
2. Fatigue Life: The ratio of the material fatigue strength to the tensile stress, scaled by the safety factor, must be greater than unity.
$$\frac{\sigma_{fatigue}}{\sigma_{tensile} \cdot 1.3} - 1 \geq 0$$
3. Static Yield: The ratio of yield strength to compressive stress must be greater than unity.
$$\frac{\sigma_{yield}}{\sigma_{tensile} \cdot 1.3} - 1 \geq 0$$
4. Monotonicity: To ensure the rod tapers smoothly from the small end to the big end, a geometric constraint enforces that the height of a subsequent element must be greater than or equal to the previous element.
$$h_{i+1} - h_i \geq 0$$

F. Numerical Implementation

The optimization was solved using the Sequential Least Squares Programming (SLSQP) algorithm from the scipy.optimize library. This gradient-based method is well-suited for constrained non-linear problems. Gradients were approximated using the Finite Difference method with a step size of $\epsilon = 10^{-4}$ to capture the sensitivity of the mass to small geometric changes. To verify the robustness of the design, a parametric sweep was performed, optimizing the rod 20 times across a range of Buckling Safety Factors (1.5 to 4.0) to generate a Pareto Front trade-off curve.

TABLE I. MATERIAL PROPERTIES OF AISI 4340 STEEL [6]

Property	Symbol	Value	Unit
Density	ρ	7850	kg/m^3
Young's Modulus	E	205	GPa
Yield Strength	σ_y	1100	MPa
Ultimate Tensile Strength	σ_{ut}	1200	MPa
Fatigue Strength	σ_f	550	MPa
Poisson's Ratio	v	0.29	-

TABLE II. GEOMETRIC CONSTRAINTS (MERCEDES-BENZ OM606)

Parameter	Value	Justification
Center-to-Center Length	149 mm	Fixed engine stroke geometry
Big End Diameter	51.6 mm	Matches Crankshaft Journal
Small End Diameter	26.0 mm	Matches Piston Pin
Min. Shank Width	12.00 mm	Stability & Forging limit
Min. Web Thickness	2.5 mm	Casting/Forging limit
Max. Shank Height	45.0 mm	Block clearance limit

III. RESULTS & DISCUSSION

A. Optimization Convergence and Morphological Evolution

The sizing optimization procedure successfully converged to a stable minimum mass solution using the Sequential Least Squares Programming (SLSQP) algorithm. For the primary design case utilizing an I-beam cross-section, the solver reduced the shank mass from an initial feasible guess to a final optimized mass of 88.34 g. The convergence history, illustrated in Fig. 4, demonstrates a rapid initial reduction in mass over the first 2 iterations. As seen in the plot, the objective function drops steeply from the initial guess as the algorithm sheds excess material from the non-critical mid-span regions. By iteration 10, the curve flattens completely, indicating that the solver has hit the "hard floor" of the active constraints.

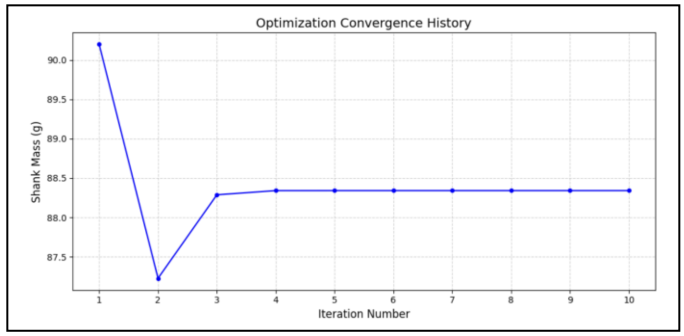

Fig. 4. Convergence plot

The final geometry, shown in Fig. 5, exhibits a distinct profile driven by the need to balance compressive area against bending stiffness. The cross-sectional height (red line) is maximized at the piston and crank ends (22 mm and 38 mm respectively) to accommodate the fixed interface constraints. The visual evolution of the cross-section is detailed in Fig. 6, showing a transition from a compact, thick-walled section at the pin end to a taller, thinner-walled section at the crank end.

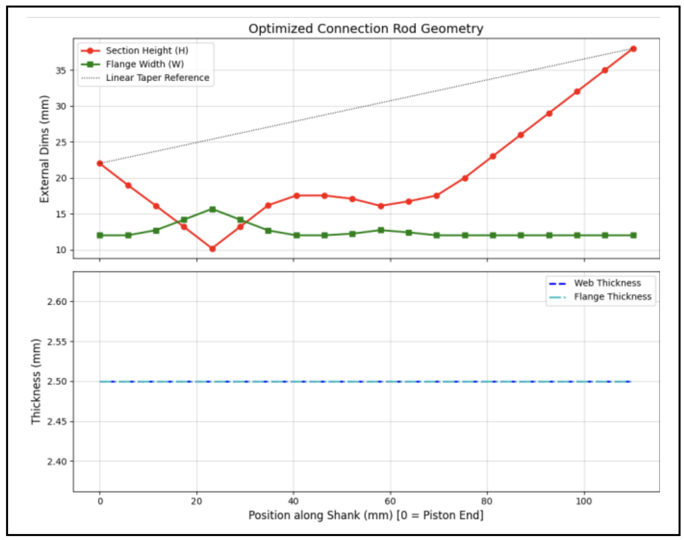

Fig. 5. Optimized connecting Rod Geom cross-sectional geometry relation for I beam configuration

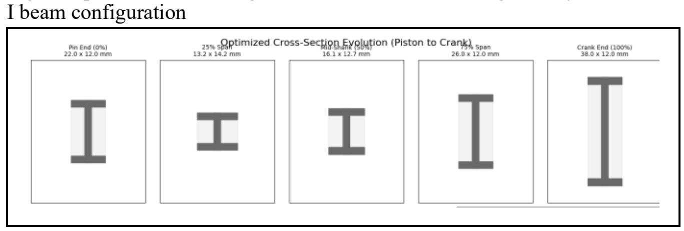

Fig. 6. Represented cross sectional areas along the shank length

B. Constraint Activity and Factor of Safety Analysis

A critical aspect of the optimization result is the identification of active constraints. The post-optimization analysis, summarized in Table III, reveals a fundamental shift in the understanding of the rod's failure modes.

The analysis indicates that the design is Yield-Critical. The calculated Factor of Safety for Static Yield under peak gas load converged exactly to the target of 1.3. This alignment identifies Yield as the active constraint that prevented further mass removal. Physically, this means the cross-sectional area (A)

could not be reduced further without exceeding the material's yield strength (σ_y), regardless of the shape's stiffness.

Furthermore, Fatigue Life was the next most critical factor (SF of 1.8). Unexpectedly, the Buckling Safety Factor reached 7.3, nearly triple the requirement of 2.5. This result is significant: it proves that the optimized I-beam geometry provides an enormous surplus of bending stiffness (I_{xx}) relative to what is required. The organic shape and I-beam flanges are so efficient at resisting buckling that the rod will fail by crushing (Yield) or snapping (Fatigue) long before it ever buckles. This insight suggests that future optimizations could potentially relax the geometric width constraints or explore lower-modulus materials without risking instability.

TABLE III. FACTOR OF SAFETY ANALYSIS FOR THE OPTIMIZED DESIGN

Constraint Mode	Required FOS	Actual FOS	Status
Buckling Stability	2.5	7.3	Inactive
Fatigue Life	1.3	1.8	Inactive
Static Yield	**1.3**	**1.3**	**Active**
Whipping	1.3	52.2	Inactive

C. *Comparative Topology Study: I-Beam vs. H-Beam*

To satisfy the requirement for exploring multiple solution spaces, a comparative optimization was performed using an H-Beam topology (refer to Appendix A). This alternative design maintained identical material properties and envelope constraints. The H-Beam optimization converged to a final mass of 103.8 g, representing a 17.5% mass penalty compared to the I-Beam.

This performance gap underscores the superior efficiency of the I-Beam for this loading scenario. Even though the design is Yield-limited (dependent on Area), the I-Beam distribution allows for a geometry that simultaneously satisfies the stiffness requirements with less material. The H-Beam, with its vertical side walls, is geometrically less efficient at maximizing the Area Moment of Inertia (I_{xx}) for the given width constraint.

IV. CONCLUSION

This project successfully demonstrated the efficacy of 1D beam sizing optimization for reducing the inertial mass of high-performance internal combustion engine components. The following conclusions are drawn from the study:

1. Yield Dominance: Contrary to the initial hypothesis of buckling criticality, the primary failure mode for the connecting rod is Static Yield under compressive gas loads. The active constraint in the final design was the Yield Safety Factor (converged to 1.3), identifying the cross-sectional area as the limiting geometric parameter.
2. Buckling Surplus: The optimized I-beam geometry provides exceptional stability, achieving a Buckling Safety Factor of 7.3. This surplus stiffness confirms that the I-beam topology is highly efficient for resisting instability in the plane of oscillation, far exceeding the requirement of 2.5.
3. Fatigue Sensitivity: Fatigue life was the secondary critical factor (SF = 1.8), driven by the inertial tensile loads at 5500 RPM. This indicates that while the rod is safe, the dynamic loads at redline are a more significant threat than buckling instability.
4. Topology Selection: The comparative study conclusively proved that the I-beam topology is superior to the H-beam for this application. The H-beam design incurred a 17.5% mass penalty (103.8 g vs. 88.34 g) to satisfy the same structural constraints within the fixed packaging envelope.
5. Methodological Efficiency: The low-fidelity 1D beam formulation enabled rapid convergence (15 iterations) and clearly identified the active constraints, validating its utility as a powerful tool for early-stage component sizing compared to computationally expensive 3D topology optimization methods.

REFERENCES

[1] Mercedes-Benz AG, "The New Diesel Generation: OM606," *Mercedes-Benz Service Introduction Manual*, Stuttgart, Germany, 1993.

[2] M. S. Shaari, M. M. Rahman, and M. M. Noor, "Design of connecting rod of internal combustion engine: A topology optimization approach," *National Conference in Mechanical Engineering Research and Postgraduate Studies*, pp. 155-166, 2010.

[3] G. P. Blair, *Design and Simulation of Four-Stroke Engines*. Warrendale, PA: SAE International, 1999.

[4] J. E. Shigley and J. J. Uicker, *Theory of Machines and Mechanisms*. New York: McGraw-Hill, 1995.

[5] F. Ferrari and O. Sigmund, "Revisiting topology optimization with buckling constraints," *Structural and Multidisciplinary Optimization*, vol. 59, no. 5, pp. 1401–1415, 2019.

[6] R. G. Budynas and J. K. Nisbett, *Shigley's Mechanical Engineering Design*, 10th ed. New York, NY: McGraw-Hill Education, 2015.

V. APPENDIX

A. *Appendix A: H-Beam Topology Results*

To validate the selection of the I-Beam topology, a comparative optimization was performed using an H-Beam cross-section under identical loading and boundary conditions. The H-Beam configuration is characterized by two vertical side walls (flanges) and a central horizontal bridge (web).

Figure A1 illustrates the optimized geometric profile of the H-Beam shank. Similar to the I-Beam, it exhibits a tapered "fish-belly" profile to minimize mass near the inflection points. However, due to the geometric inefficiency of vertical walls in resisting in-plane buckling (restricted by the 35mm width constraint), the solver was forced to maintain a larger cross-sectional area to meet the active Yield constraint.

Figure A2 visualizes the cross-sectional evolution of the H-Beam from the pin end to the crank end. The final optimized mass for this topology was 103.8 g, which is 17.5% heavier than the optimal I-Beam design (88.34 g), confirming the I-Beam as the superior choice for this specific application.

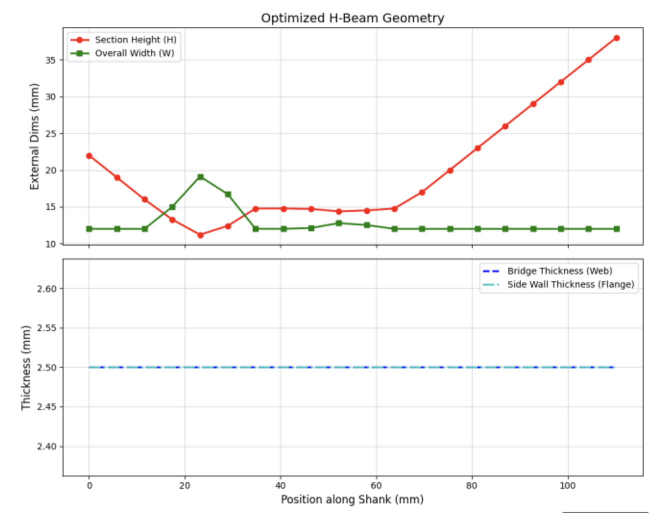

Fig. A1. Optimized connecting Rod Geom cross-sectional geometry relation for H beam configuration

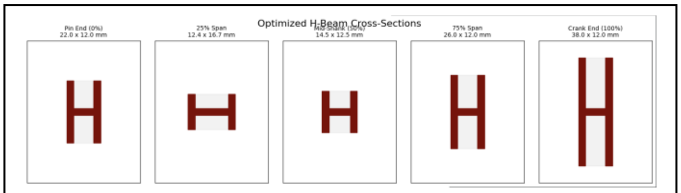

Fig. A2. Represented cross sectional areas along the shank length for H beam configuration

B. Appendix B: Optimization Code

```
import numpy as np
from scipy.optimize import minimize
from scipy.linalg import eigh
import matplotlib.pyplot as plt
import matplotlib.patches as patches
from matplotlib.ticker import MaxNLocator

# 1. SETUP
NUM_ELEMENTS = 20
L_center_to_center = 0.149
R_big = 0.0516 / 2
R_small = 0.026 / 2
L_shank = L_center_to_center - R_big - R_small
L_elem = L_shank / NUM_ELEMENTS

# Mass Definitions
Mass_Recip_Total = 1.500
Mass_Small_End = 0.250
Mass_Piston_Assy = Mass_Recip_Total - Mass_Small_End

F_comp = 77300.0
RPM_Max = 5500.0

# Material (AISI 4340)
E = 205e9
Rho = 7850.0
Sigma_Y = 1100e6
Sigma_Fatigue = 550e6

# Safety Factors
SF_buckling = 2.5
SF_yield = 1.3
SF_fatigue = 1.3
```

```
# Geometric Limits
W_min, W_max = 0.012, 0.035
H_min, H_max = 0.010, 0.045
T_min, T_max = 0.0025, 0.008

# Fixed Constraints for Ends
H_small_end = 0.022
H_big_end = 0.038

# Max Slope (Smoothness Constraint)
MAX_SLOPE_PER_ELEM = 0.003

def unpack_variables(x):
    h = x[0:NUM_ELEMENTS]
    w = x[NUM_ELEMENTS:2*NUM_ELEMENTS]

    t_scale = np.linspace(0, 1, NUM_ELEMENTS)
    t_web = x[-4] + (x[-3] - x[-4]) * t_scale
    t_flange = x[-2] + (x[-1] - x[-2]) * t_scale

    return h, w, t_web, t_flange

def get_section_properties(x):
    h, w, t_web, t_flange = unpack_variables(x)
    A = np.zeros(NUM_ELEMENTS)
    I = np.zeros(NUM_ELEMENTS)

    for i in range(NUM_ELEMENTS):
        h_web = h[i] - 2 * t_flange[i]
        if h_web < 0: h_web = 0

        A[i] = (2 * w[i] * t_flange[i]) + (h_web * t_web[i])

        b_void = w[i] - t_web[i]
        if b_void < 0: b_void = 0
        I[i] = (w[i] * h[i]**3 - b_void * h_web**3) / 12

    return A, I

def get_shank_mass(x):
    A, _ = get_section_properties(x)
    return np.sum(A) * L_elem * Rho

def solve_buckling(I_vals):
    dof = (NUM_ELEMENTS + 1) * 2
    K = np.zeros((dof, dof))
    Kg = np.zeros((dof, dof))

    k_fac = E / L_elem**3
    g_fac = 1.0 / (30 * L_elem)

    Ke_base = np.array([[12, 6*L_elem, -12, 6*L_elem],
                        [6*L_elem, 4*L_elem**2, -6*L_elem, 2*L_elem**2],
                        [-12, -6*L_elem, 12, -6*L_elem],
                        [6*L_elem, 2*L_elem**2, -6*L_elem, 4*L_elem**2]])

    Kge_base = np.array([[36, 3*L_elem, -36, 3*L_elem],
                         [3*L_elem, 4*L_elem**2, -3*L_elem, -L_elem**2],
                         [-36, -3*L_elem, 36, -3*L_elem],
                         [3*L_elem, -L_elem**2, -3*L_elem, 4*L_elem**2]])

    for i in range(NUM_ELEMENTS):
        Ke = (k_fac * I_vals[i]) * Ke_base
        Kge = g_fac * Kge_base

        idx = [2*i, 2*i+1, 2*i+2, 2*i+3]
        for r in range(4):
            for c in range(4):
                K[idx[r], idx[c]] += Ke[r,c]
                Kg[idx[r], idx[c]] += Kge[r,c]

    fixed_dofs = [0, 2*NUM_ELEMENTS]
    free = [i for i in range(dof) if i not in fixed_dofs]

    K_reduced = K[np.ix_(free, free)]
    Kg_reduced = Kg[np.ix_(free, free)]

    vals = eigh(K_reduced, Kg_reduced, eigvals_only=True)
    pos = vals[vals > 0.1]

    return np.min(pos) if len(pos) > 0 else 0.1

def solve_whipping(x):
    h, _, _, _ = unpack_variables(x)
    A, I = get_section_properties(x)
    omega = RPM_Max * 0.1047

    x_c = np.linspace(L_elem/2, L_shank - L_elem/2, NUM_ELEMENTS)
    accel = (omega**2 * 0.042) * (x_c / L_shank)
    q = Rho * A * accel

    total_force = np.sum(q * L_elem)
    moment_about_pin = np.sum(q * L_elem * x_c)
```

```
    R_big = moment_about_pin / L_shank
    R_small = total_force - R_big

    V = R_small
    M = 0
    max_sigma = 0

    for i in range(NUM_ELEMENTS):
        load = q[i] * L_elem
        M += V * L_elem - load * (L_elem/2)
        V -= load

        if I[i] > 1e-12:
            s = abs(M) * (h[i]/2) / I[i]
        else:
            s = 1e9

        if s > max_sigma: max_sigma = s

    return max_sigma

#CONSTRAINTS

def c_buckling(x):
    _, I = get_section_properties(x)
    return (solve_buckling(I) / (SF_buckling * F_comp)) - 1.0

def c_fatigue(x):
    A, _ = get_section_properties(x)
    shank_mass = get_shank_mass(x)

    m_effective = Mass_Piston_Assy + Mass_Small_End + (0.33 * shank_mass)
    accel = 0.042 * (RPM_Max*0.1047)**2 * (1 + 0.042/L_center_to_center)
    F_inertia = m_effective * accel

    sigma = F_inertia / A
    return (Sigma_Fatigue/SF_fatigue) - sigma

def c_yield(x):
    A, _ = get_section_properties(x)
    sigma = F_comp / A
    return (Sigma_Y/SF_yield) - sigma

def c_whipping(x):
    return (Sigma_Y/SF_yield) - solve_whipping(x)

def c_smoothness(x):
    h = x[0:NUM_ELEMENTS]
    diffs = np.diff(h)
    return np.concatenate((MAX_SLOPE_PER_ELEM - diffs, diffs + MAX_SLOPE_PER_ELEM))

def c_geometry(x):
    _, w, t_web, _ = unpack_variables(x)
    return (w - t_web) - 0.002

#EXECUTION & RESULTS
def generate_report_results():
    print("Generating Final Report Figures...")

    # 1. SETUP OPTIMIZATION
    bounds = []
    # Height Bounds
    for i in range(NUM_ELEMENTS):
        if i == 0: bounds.append((H_small_end, H_small_end + 0.0001))
        elif i == NUM_ELEMENTS-1: bounds.append((H_big_end, H_big_end + 0.0001))
        else: bounds.append((H_min, H_max))
    # Width Bounds
    for i in range(NUM_ELEMENTS): bounds.append((W_min, W_max))
    # Thickness Bounds
    for _ in range(4): bounds.append((T_min, T_max))

    # Initial Guess
    x_norm = np.linspace(-1, 1, NUM_ELEMENTS)
    h_guess = 0.025 + 0.005 * x_norm**2
    w_guess = np.full(NUM_ELEMENTS, 0.020)
    t_guess = np.array([0.004, 0.004, 0.004, 0.004])
    x0 = np.concatenate((h_guess, w_guess, t_guess))

    # Constraints
    cons = (
        {'type': 'ineq', 'fun': c_buckling},
        {'type': 'ineq', 'fun': c_fatigue},
        {'type': 'ineq', 'fun': c_yield},
        {'type': 'ineq', 'fun': c_whipping},
        {'type': 'ineq', 'fun': c_smoothness},
        {'type': 'ineq', 'fun': c_geometry}
    )

    history = []
    def callback_function(xk):
        mass = get_shank_mass(xk)
        history.append(mass * 1000)
```

```
    #Optimization runner
    res = minimize(
        fun=get_shank_mass,
        x0=x0,
        method='SLSQP',
        bounds=bounds,
        constraints=cons,
        callback=callback_function, # Hook in the tracker
        options={'ftol': 1e-4, 'maxiter': 200, 'disp': True}
    )

    print(f"Optimization Complete. Final Mass: {get_shank_mass(res.x)*1000:.2f} g")

    # Unpack Results
    h_opt, w_opt, t_web, t_flange = unpack_variables(res.x)
    x_loc = np.linspace(0, L_shank*1000, NUM_ELEMENTS)
    fig1, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8), sharex=True)

    ax1.plot(x_loc, h_opt*1000, 'r-o', label='Section Height (H)', linewidth=2)
    ax1.plot(x_loc, w_opt*1000, 'g-s', label='Flange Width (W)', linewidth=2)
    h_min_line = np.linspace(H_small_end*1000, H_big_end*1000, NUM_ELEMENTS)
    ax1.plot(x_loc, h_min_line, 'k:', label='Linear Taper Reference', linewidth=1)

    ax1.set_ylabel('External Dims (mm)', fontsize=12)
    ax1.set_title('Optimized Connection Rod Geometry', fontsize=14)
    ax1.legend(loc='upper left')
    ax1.grid(True, alpha=0.5)

    ax2.plot(x_loc, t_web*1000, 'b--', label='Web Thickness', linewidth=2)
    ax2.plot(x_loc, t_flange*1000, 'c-.', label='Flange Thickness', linewidth=2)

    ax2.set_ylabel('Thickness (mm)', fontsize=12)
    ax2.set_xlabel('Position along Shank (mm) [0 = Piston End]', fontsize=12)
    ax2.legend(loc='upper right')
    ax2.grid(True, alpha=0.5)
    plt.tight_layout()
    plt.show()

    _, I_opt = get_section_properties(res.x)
    A_opt, _ = get_section_properties(res.x)

    P_cr = solve_buckling(I_opt)
    SF_actual_buckling = P_cr / F_comp

    m_recip = Mass_Piston_Assy + Mass_Small_End + (0.33 * get_shank_mass(res.x))
    F_inertia = m_recip * 0.042 * (RPM_Max*0.1047)**2 * (1 + 0.28)
    SF_actual_fatigue = Sigma_Fatigue / (F_inertia / np.min(A_opt))

    SF_actual_yield = Sigma_Y / (F_comp / np.min(A_opt))

    sigma_whip = solve_whipping(res.x)
    SF_actual_whip = Sigma_Y / sigma_whip

    fig2, ax = plt.subplots(figsize=(10, 5))
    modes = ['Buckling', 'Fatigue', 'Yield', 'Whipping']
    actuals = [SF_actual_buckling, SF_actual_fatigue, SF_actual_yield, SF_actual_whip]
    targets = [SF_buckling, SF_fatigue, SF_yield, SF_yield]

    x_pos = np.arange(len(modes))
    width = 0.35

    rects1 = ax.bar(x_pos - width/2, actuals, width, label='Actual SF', color='royalblue')
    rects2 = ax.bar(x_pos + width/2, targets, width, label='Required SF', color='firebrick')

    ax.set_ylabel('Factor of Safety')
    ax.set_title('Constraint Activity Check (Active vs Inactive)', fontsize=14)
    ax.set_xticks(x_pos)
    ax.set_xticklabels(modes)
    ax.legend()
    ax.grid(axis='y', alpha=0.3)

    def autolabel(rects):
        for rect in rects:
            height = rect.get_height()
            ax.annotate(f'{height:.1f}', xy=(rect.get_x() + rect.get_width() / 2, height),
                        xytext=(0, 3), textcoords="offset points", ha='center', va='bottom')
    autolabel(rects1)
    autolabel(rects2)
    plt.show()
```

```
    fig3, axes = plt.subplots(1, 5, figsize=(15, 4))
    indices = [0, NUM_ELEMENTS//4, NUM_ELEMENTS//2, 3*NUM_ELEMENTS//4, NUM_ELEMENTS-1]
    labels = ['Pin End (0%)', '25% Span', 'Mid-Shank (50%)', '75% Span', 'Crank End (100%)']

    for ax, idx, label in zip(axes, indices, labels):
        H = h_opt[idx] * 1000
        W = w_opt[idx] * 1000
        Tw = t_web[idx] * 1000
        Tf = t_flange[idx] * 1000

        ax.add_patch(patches.Rectangle((-W/2, -H/2), W, H, color='gray', alpha=0.1))
        ax.add_patch(patches.Rectangle((-Tw/2, -H/2), Tw, H, color='dimgray'))
        ax.add_patch(patches.Rectangle((-W/2, H/2 - Tf), W, Tf, color='dimgray'))
        ax.add_patch(patches.Rectangle((-W/2, -H/2), W, Tf, color='dimgray'))

        ax.set_xlim(-20, 20)
        ax.set_ylim(-25, 25)
        ax.set_title(f"{label}\n{H:.1f} x {W:.1f} mm", fontsize=10)
        ax.set_aspect('equal')
        ax.grid(True, alpha=0.3)
        ax.set_xticks([])
        ax.set_yticks([])

    plt.suptitle("Optimized Cross-Section Evolution (Piston to Crank)", fontsize=16)
    plt.tight_layout()
    plt.show()

    fig4, ax4 = plt.subplots(figsize=(10, 5))

    # Plot history
    iterations = np.arange(1, len(history) + 1)
    ax4.plot(iterations, history, 'b-o', markersize=4, label='Objective Value')

    # Formatting
    ax4.set_title("Optimization Convergence History", fontsize=14)
    ax4.set_xlabel("Iteration Number", fontsize=12)
    ax4.set_ylabel("Shank Mass (g)", fontsize=12)
    ax4.grid(True, linestyle='--', alpha=0.6)

    # Force integer X-axis
    ax4.xaxis.set_major_locator(MaxNLocator(integer=True))

    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    generate_report_results()
```