
Electronic supplementary materials

For <https://doi.org/10.1631/jzus.A2400235>

Rail profile optimization through balancing of wear and fatigue

Binjie XU¹, Zhiyong SHI², Yun YANG¹, Jianxi WANG³, Kaiyun WANG^{1✉}

¹State Key Laboratory of Rail Transit Vehicle System, Southwest Jiaotong University, Chengdu 610031, China

²Department of Industrial Engineering, University of Florence, Florence, Italy

³School of Civil Engineering, Shijiazhuang Tiedao University, Shijiazhuang 050001, China

✉ Kaiyun WANG, kywang@swjtu.edu.cn

Section S1 Archard wear theory

In the Archard wear theory, the material wear is directly proportional to the normal contact pressure and relative sliding distance, and inversely proportional to the hardness of the contact material:

$$V_w = k_w \frac{F_n \times S}{H} \quad (S1)$$

where, V_w is the volume of material wear within the contact patch, F_n is the normal contact stress, H is the material hardness, S is the relative sliding distance, and k_w is the wear coefficient.

The relative sliding distance and the sliding velocity of the contact patch are calculated as follows:

$$d = v_0 \left| v_s \right| \frac{\Delta x}{v_0} \quad (S2)$$

$$v_s = v_0 \left| \begin{array}{c} \xi_1 - x_2 \xi_3 - \frac{\partial u_1}{\partial x_1} \\ \xi_2 - x_1 \xi_3 - \frac{\partial u_2}{\partial x_1} \end{array} \right|. \quad (S3)$$

where, ξ_1 , ξ_2 , and ξ_3 are the longitudinal, transverse, and spin creep rates, and u_1 and u_2 are the longitudinal and transverse elastic displacements, respectively.

Section S2 Wheel-rail creep rate

When the wheel rolls, the front part of the wheel produces compressive deformation, the back part produces tensile deformation, while the rail stretches in front and is compressed at the back. As a result, the distance traveled by the wheel during rolling is smaller than that of pure rolling, and this phenomenon is known as wheel-rail creep. The calculation of the longitudinal, lateral, and spin creep rates is as follows:

$$\begin{aligned}\xi_1 &= \frac{(V_{w1} - V_{r1})}{(V_{w1} + V_{r1}) / 2} \\ \xi_2 &= \frac{(V_{w2} - V_{r2})}{(V_{w2} + V_{r2}) / 2} \\ \xi_3 &= \frac{(\Omega_{w3} - \Omega_{r3})}{(V_{w1} + V_{r1}) / 2}\end{aligned}\quad (S4)$$

where V_{w1} , V_{r1} are wheel and rail longitudinal velocity; V_{w2} , V_{r2} are wheel and rail lateral velocity; Ω_{w3} , Ω_{r3} are wheel and rail angular velocity.

Section S3 The equivalent stress

The equivalent stress σ_{eq} for each position j of node i is:

$$\sigma_{eq} = \frac{1}{\sqrt{2}} [(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]^{0.5} + (1 - 2\nu)\sigma_h \quad (S5)$$

$$\sigma_h = \frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3) \quad (S6)$$

Then, the characteristic of the variable stress cycle σ_m and the mean value of the cyclic stress σ_a is:

$$\begin{aligned}\sigma_m &= (\sigma_{eq}^{\max} + \sigma_{eq}^{\min}) / 2 \\ \sigma_a &= (\sigma_{eq}^{\max} - \sigma_{eq}^{\min}) / 2\end{aligned}\quad (S7)$$

The Serensen-Kinasoshvili approximation theory of ultimate stress is used in calculating rail fatigue damage by the rolling contact fatigue damage criterion adopted in this paper. Then, the average value σ_m and amplitude σ_a of the cyclic stress are:

$$\begin{aligned}\sigma_m &= \sigma_a = \sigma_0 / 2 \\ \sigma_{RCF} &= \psi_\sigma \sigma_m + \sigma_a\end{aligned}\quad (S8)$$

where σ_m is the average value of cyclic stress, σ_a is the amplitude of cyclic stress, and ψ_σ is the sensitivity coefficient of material asymmetry.

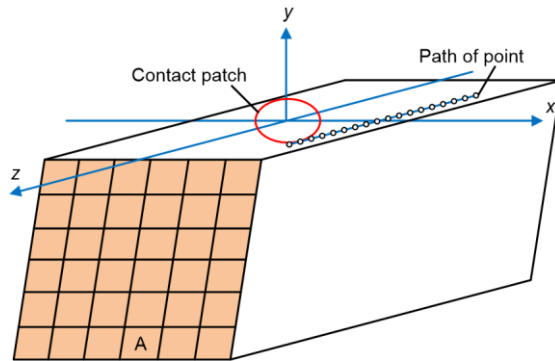


Fig. S1 Rail-wheel contact model

Section S4 The wagon structures

Friction pairs such as the center plate friction and side bearing friction are considered between the bolster and car body; vertical and lateral suspensions are directly equivalent between the car body and side frame, with vertical, lateral, and longitudinal stiffnesses of the bolster spring and friction characteristics of the wedge between the side frame and bolster being accounted for. Moreover, the lateral and longitudinal clearance of the axle box suspension is considered, and a spring-damping device is used within the range of axle box clearance; also, a spring-damping

device in parallel with a linear spring stop is implemented beyond the contact of the axle box with the side frame. The lateral support rods between the side frames are simplified to display torsional stiffness.

Section S5 Model validation

Fig. S2 compares the differences between simulation results and measurement results. From Fig. S2a and S2b, the trend of the rail wear test is consistent with the simulation results. The maximum error between the simulated and calculated results was 7.9%, most errors were less than 6%, and the average error was 4.2%. Observing Fig. S2c and S2d, the simulated trends of coupler force and coupler yaw angle closely resemble the measured data. The maximum coupler yaw angles were measured at 10.9° and simulated to be 11.2° , with the maximum measured coupler force around 594 kN, compared to 611 kN in simulation. The maximum coupler force and rotation angle differences in the simulation results were less than 3%. The simulation results indicate that these dynamic and wear models reasonably reflect real-world conditions.

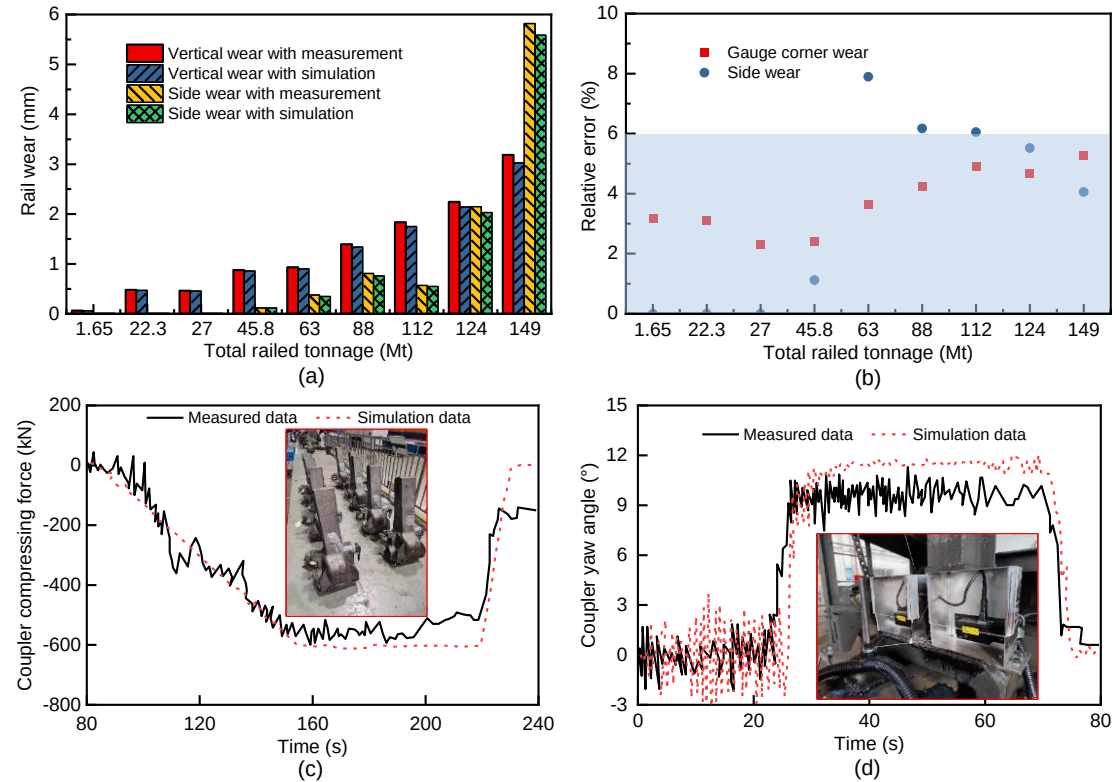


Fig. S2 Model validation results: (a) Rail wear; (b) Relative error; (c) Coupler compressing force; (d) Coupler yaw angle

Section S6 NURBS curve fitting method

NURBS is a curve of degree k defined by $n+1$ control points of the form:

$$p(u) = \frac{\sum_{i=0}^n w_i d_i N_{i,k}(u)}{w_i N_{i,k}(u)} \quad (i = 0, 1, \dots, n) \quad (S9)$$

where d_i ($i=0, 1, \dots, n$) is the control point; w_i is the corresponding weight factor; $N_{i,k}(u)$ is a B-spline basis function of degree k , determined by the node vector $U=[u_0, u_1, \dots, u_{n+k+1}]$. The B-spline basis function of degree k is defined by the Cox-DeBoor recursion formula as follows:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, u \in [u_i, u_{i+1}] \\ 0, \text{else} \end{cases} \\ N_{i,0}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \end{cases} \quad (\text{S10})$$

The node vector $U=[u_0, u_1, \dots, u_{n+k+1}]$ is determined according to the normalized cumulative chord length parameterization method:

$$\begin{cases} u_0 = 0 \\ u_i = u_{i-1} + |\Delta p_{i-1}|, i = 1, 2, \dots, n \end{cases} \quad (\text{S11})$$

where $|\Delta p_{i-1}|$ is the forward difference vector, and $\Delta p_i = p_{i+1} - p_i$ is the chord length vector.

The NURBS description of the rail profile curve is established by using finite discrete points (type value points) on the rail profile. Given n -type value points p_i and their corresponding $n+2$ weight factors h ($i=0, \dots, n+1$), we need to perform inverse design for the control points. The relationship between the weight factor of control point w_i and the weight factor of type value point h_i is as follows:

$$h_{i+1} = h(u_{i+3}) = \sum \omega_i N_{i,3}(u_{i+3}). \quad (\text{S12})$$

According to Eq. (S12), n equations can be listed. Two tangent vector boundary conditions are added according to the curve characteristics of the rail profile:

$$\frac{3(\omega_1 - \omega_0)}{u_4 - u_3} = h_1 = h_0, \quad (\text{S13})$$

$$\frac{3(\omega_{n+1} - \omega_n)}{u_{n+2} - u_{n+1}} = h_n = h_{n+1}. \quad (\text{S14})$$

Section S7 BP neural network

The calculation method of the BP neural network is the Gauss-Newton method:

$$\omega(k+1) = \omega(k) - [J^T(\omega_k)J(\omega_k)]^{-1} J(\omega_k)e(\omega_k) \quad (\text{S15})$$

where $\omega(k)$ represents the vector composed of the weights and thresholds of the K th iteration.

The structure of the Levenberg-Marquardt algorithm is as follows:

$$\omega(k+1) = \omega(k) - [J^T(\omega_k)J(\omega_k) + \mu_k I]^{-1} J(\omega_k)e(\omega_k) \quad (\text{S16})$$

We randomly initialize the weights and thresholds for the LM-BP neural network. If the initial value is incorrectly selected, each iteration will amplify the system error. Therefore, we employ the PSO (particle swarm optimization) algorithm to optimize the weights and thresholds of the neural network. PSO is a swarm intelligence optimization algorithm where in each iteration, particles self-update by tracking their individual best (P_{best}) and the global best (G_{best}) performances. The updating equations for velocity and position are as follows:

$$\begin{aligned}
V_{id}^{t+1} &= \omega V_{id}^t + c_1 r_1 (P_{\text{best}} - X_{id}^t) + c_2 r_2 (G_{\text{best}} - X_{id}^t), \\
X_{id}^{t+1} &= X_{id}^t + V_{id}^{t+1}, \\
\omega(t) &= \omega_{\max} - \frac{t(\omega_{\max} - \omega_{\min})}{t_{\max}}.
\end{aligned} \tag{S17}$$

Here, V_{id}^t , X_{id}^t , and P_{best} are the iterative velocity, position, and individual optimal solutions for particle i at time t , respectively; G_{best} is the optimal position of the population at time t ; c_1 and c_2 are the learning factor coefficients; r_1 and r_2 are random numbers from 0 to 1, and ω is the weight. Finally, $t_{\max}$ indicates the maximum number of iterations. The PSO-LM-BP neural network is shown in Fig. S3.

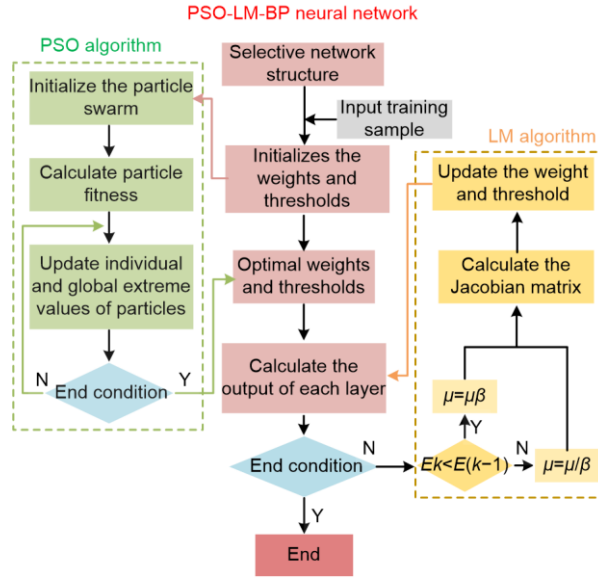


Fig. S3 PSO-LM-BP neural network architecture

Section S8 Levenberg-Marquardt algorithm

The most critical step in the LM algorithm is the calculation of the Jacobian matrix, which is computed using a modified version of the BP algorithm:

$$J(\omega) = \begin{bmatrix} \frac{\partial e_1(\omega)}{\partial \omega_{11}^1} & \frac{\partial e_1(\omega)}{\partial \omega_{21}^1} & \dots & \frac{\partial e_1(\omega)}{\partial \omega_{LL}^2} & \frac{\partial e_1(\omega)}{\partial \theta_1^1} & \dots & \frac{\partial e_1(\omega)}{\partial \theta_L^2} \\ \frac{\partial e_2(\omega)}{\partial \omega_{11}^1} & \frac{\partial e_2(\omega)}{\partial \omega_{21}^1} & \dots & \frac{\partial e_2(\omega)}{\partial \omega_{LL}^2} & \frac{\partial e_2(\omega)}{\partial \theta_1^1} & \dots & \frac{\partial e_2(\omega)}{\partial \theta_L^2} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial e_L(\omega)}{\partial \omega_{11}^1} & \frac{\partial e_L(\omega)}{\partial \omega_{21}^1} & \dots & \frac{\partial e_L(\omega)}{\partial \omega_{LL}^2} & \frac{\partial e_L(\omega)}{\partial \theta_1^1} & \dots & \frac{\partial e_L(\omega)}{\partial \theta_L^2} \end{bmatrix} \tag{S18}$$

If the scaling factor $\mu=0$, the LM algorithm is equivalent to the Gauss-Newton method; if the scaling factor takes a tremendous value, the LM algorithm approaches gradient descent.

Section S9 Microvariation, Adaptive mutation operator and Chaotic disturbance

Microvariation

A small probability mutation is introduced by incorporating chaotic disturbances during

individual replication. This approach partly mitigates the phenomenon of “premature convergence” caused by the selection process. Also, it reduces the probability of crossover between identical individuals. The concept of chaotic disturbance is described further in the electronic supplementary materials.

Adaptive mutation operator

The adaptive mutation operator implies that the mutation strength should gradually increase with the number of generations, enabling the search algorithm to escape local optima:

$$P_m = \begin{cases} P_{m\max}, & f \geq \bar{f}, \\ P_{m\min} + \eta \cdot \frac{f_{\min} - f}{f_{\max} - f_{\min}} \cdot \frac{1}{1 + e^{-t}}, & f < \bar{f}. \end{cases} \quad (S19)$$

In Eq. (S19), the maximum and minimum values of the mutation operators are $P_{m\max}=0.05$ and $P_{m\min}=0.001$, respectively, with $\eta=P_{m\max}-P_{m\min}$.

Also, the parameter variation mode is:

$$X'_n = (1 - \beta) \cdot X_n + \frac{(-1)^n \cdot \beta \cdot X_n}{|X_n|}. \quad (S20)$$

Chaotic disturbance

The concept of chaotic disturbance is:

$$\begin{cases} X'_n = (1 - \alpha \cdot \beta) \cdot X_n + (-1)^n \cdot \alpha \cdot \beta \cdot X_n / |X_n| \\ X_n = (x_1, x_2, \dots, x_i), n = 1, 2, \dots, size \\ \beta = (\beta_1, \beta_2, \dots, \beta_i, \dots, \beta_L) \end{cases} \quad (S21)$$

where X'_n is the new individual; X_n is the current individual; β is the chaotic disturbance operator; α is the influence factor of artificial degradation, and its value range is (0, 0.01); size is the population size. Firstly, the chaos disturbance factor β is calculated. Then, the disturbance radius $\alpha\beta$ is determined by reducing the disturbance size by artificial degradation factor α . $(-1)^n$ can make the disturbance evenly traverse the points with gene as the center and interval length $2\alpha\beta$.

Individuals with high fitness functions are preserved within the population to avoid falling into local optima. In contrast, individuals with low fitness functions undergo additional chaotic optimization processing. Additionally, incorporating chaotic optimization can bring individuals closer to the optimal solution and reduce the number of evolutionary generations. The iterative formula for chaotic optimization utilizes a tent map:

$$t_{n+1} = \begin{cases} v \cdot t_n, & t_n \leq 0.5, \\ v \cdot (1 - t_n), & t_n > 0.5. \end{cases} \quad (S22)$$

Here, the tent has a chaotic effect when v is in the range of 1 to 2.

Chaotic disturbance can be added as follows:

$$X'_n = (X_{\max} - X_{\min}) \cdot (t - 0.5) + (X_{\max} + X_{\min}) \cdot 0.5. \quad (S23)$$

where $X_{\max}$ and $X_{\min}$ are the upper and lower limits of an individual, respectively.

Thus, we employed a chaotic microvariation adaptive genetic algorithm to address the multi-

objective optimization problem involving control point coordinates and wheel-rail damage. The target value in Fig. S4 represents the rail damage. As the number of evolutionary iterations increases, the target value progressively diminishes. In the optimization algorithm, when the number of evolutionary iterations reaches 50, the objective function achieves a global optimal value. When the number of iterations reaches 100, the target value achieves convergence. This result demonstrates that the proposed optimization algorithm exhibits strong convergence properties for rail profile optimization.

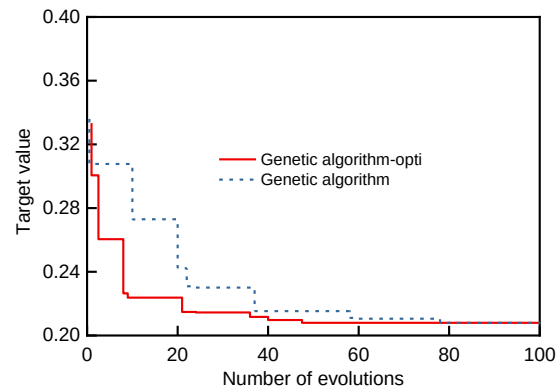


Fig. S4 Genetic Algorithm-Opti evolution curve