applied sciences

MDPI

Article

An Improved NSGA-II-Based Method for Cutting Trajectory Planning of Boom-Type Roadheader

Chao Zhang [1], Xuhui Zhang [1,2,*], Wenjuan Yang [1,2], Jicheng Wan [1], Guangming Zhang [3], Yuyang Du [1,2], Sihao Tian [1] and Zeyao Wang [1]

[1] School of Mechanical Engineering, Xi'an University of Science and Technology, Xi'an 710054, China; 20105016004@stu.xust.edu.cn (C.Z.); yangwj@xust.edu.cn (W.Y.); 20105016013@stu.xust.edu.cn (J.W.); dyy@xust.edu.cn (Y.D.); 23205224164@stu.xust.edu.cn (S.T.); 23205016031@stu.xust.edu.cn (Z.W.)
[2] Shaanxi Key Laboratory of Intelligent Detection and Control for Mining Electromechanical Equipment, Xi'an 710054, China
[3] Faculty of Engineering and Technology, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK; g.zhang@ljmu.ac.uk
* Correspondence: zhangxh@xust.edu.cn

Featured Application: This method enhances autonomous cutting for boom-type roadheaders by improving trajectory planning, ensuring precise roadway section formation in confined environments. It has potential applications in coal mining and tunnel construction, optimizing operational efficiency and reducing manual intervention.

Abstract: This paper proposes a cutting trajectory planning method for boom-type roadheaders using an improved Nondominated Sorting Genetic Algorithm II (NSGA-II) with an elitist strategy. Existing methods often overlook constraints related to cutterhead dimensions and target sections, affecting section formation quality. We develop a kinematic model for coordinate transformations and design a simplified cutterhead and constraint model to generate feasible cutting points. Bi-objective functions—minimizing cutting trajectory length and turning angle—are formulated as a bi-objective traveling salesman problem (BO-TSP) with adjacency constraints. NSGA-II is adapted with enhancements in adjacency constraint handling, population initialization, and genetic operations. Simulations and experiments demonstrate significant improvements in convergence speed and computation time. Virtual cutting experiments confirm trajectory feasibility under varying postures, achieving high formation quality. A comparison of planned and tracked trajectories shows a maximum deviation of 23.879 mm, supporting autonomous cutting control. This method advances cutting trajectory planning for roadway section formation and autonomous roadheader control.

Keywords: cutting trajectory planning; improved NSGA-II; section forming; virtual debugging; boom-type roadheader

Academic Editor: Rui Araújo

Received: 7 January 2025
Revised: 15 February 2025
Accepted: 16 February 2025
Published: 17 February 2025

Citation: Zhang, C.; Zhang, X.; Yang, W.; Wan, J.; Zhang, G.; Du, Y.; Tian, S.; Wang, Z. An Improved NSGA-II-Based Method for Cutting Trajectory Planning of Boom-Type Roadheader. *Appl. Sci.* **2025**, *15*, 2126. https://doi.org/10.3390/app15042126

1. Introduction

In coal mining, cutting trajectory planning for boom-type roadheaders is a key factor influencing the formation quality of roadways and mining efficiency [1]. With advancements in intelligent mining, traditional trajectory planning methods struggle to address the complexities and uncertainties of mining environments. In particular, the dynamic positional changes of cutterheads significantly affect roadway formation quality, an issue that remains inadequately addressed [2,3]. Achieving optimized roadway cutting quality

while ensuing trajectory smoothness and mining efficiency has become a major challenge in coal mine intelligence research.

Unlike manual operations, trajectory planning ensures precise cutting task execution in complex environments, meeting the required roadway formation standards. Existing methods consider various factors such as kinematics, target geometry, and operational conditions. Tian et al. [4] introduced a simplified kinematic model using homogeneous coordinate transformation and robotics theory, achieving effective trajectory planning through simulation [5]. Wu et al. [6] enhanced planning accuracy by incorporating dynamic modeling and path correction. Xu et al. [7] solved the kinematics of the cutterhead using the D-H parameter method and inverse transformation techniques to plan the cutting trajectory. However, these approaches often neglect cutterhead dimensions, leading to suboptimal formation precision [8]. To address this, research has explored cutterhead optimization to improve formation quality. Full-scale simulations of cutterheads at various inclination angles and genetic algorithm optimization of spiral angles demonstrated significant performance improvements in energy consumption, cutting resistance, and vibration control [9,10]. Despite these advances, conventional methods remain limited in adapting to complex mining conditions.

Qu et al. [11] proposed a path correction and scheduling strategy for an underground mining roadheader, integrating BP neural networks and state estimation. By designing a tracking model based on posture deviations and applying SVD-unscented Kalman filtering to estimate the actual posture deviations, they achieved effective path tracking and highly adaptive control of the roadheader. Wang et al. [12] introduced a trajectory planning method for roadheader cutting under uncertainty. By analyzing the cutting mechanism, identifying influencing factors, and using a particle swarm optimization algorithm, they developed an optimal trajectory planning approach that avoids contamination zones and ensures precise roadway formation, laying the foundation for robotic automatic tunneling. Additionally, by analyzing the relationship between coal-rock hardness and the speed of the cutting arm, integrating multi-sensor parameter evaluation, and modeling the environment in a grid format, they proposed a cutting trajectory planning and boundary control method. Experimental results demonstrated that this approach accurately controls the cutterhead to avoid obstacles and completely cut the cross-section [13]. Adaptive algorithms have been proposed to mitigate challenges such as coal-rock variability and obstacles. Wang et al. [5] used ant colony optimization (ACO) with sensor data for obstacle avoidance, enhancing safety and efficiency. Dong et al. [14] integrated machine vision to detect fractures and dynamically adjust cutting paths, demonstrating adaptability to changing environments. However, these approaches often rely on real-time detection methods, which can be ineffective under complex working conditions. Multi-objective optimization has emerged as a critical focus for trajectory planning, addressing conflicting goals such as reducing cutting energy consumption and shortening path length. Mao et al. [15] combined particle swarm optimization (PSO) and genetic algorithms to overcome local optima, improving trajectory quality under varying mining conditions. The NSGA-II algorithm has become a standard for solving multi-objective optimization problems in trajectory planning. Its applications include addressing uncertainties in key parameters, such as cutterhead speed, trajectory path, and environmental factors, which can fluctuate during mining operations. Additionally, it facilitates the design of time-optimal trajectories, ensuring that the cutting process is both efficient and precise while minimizing time consumption in complex mining environments [16–18]. These studies highlight NSGA-II's versatility and effectiveness across diverse scenarios [19–21].

In a related domain, Machmudah et al. [22] optimized the flight trajectories of fixed-wing UAVs using a bank-turn mechanism, demonstrating how optimization techniques can

handle dynamic constraints in trajectory planning. While UAV trajectory planning differs from mining applications, their approach offers valuable insights into the optimization of constrained trajectories. These methodologies can inform the development of optimized cutting trajectory planning for roadheaders, particularly in addressing dynamic positional changes of the cutterhead and ensuring smooth, efficient cuts in complex environments.

To enhance trajectory planning precision and ensure roadway formation quality, this paper proposes an NSGA-II-based method that integrates cutterhead dimensions, target constraints, and machine posture. A bi-objective function is constructed to optimize the cutting trajectory, formalized as a bi-objective traveling salesman problem with adjacency constraints (BOTSP-AC). The NSGA-II algorithm is further improved for this problem, and its effectiveness is validated through numerical simulations and cutting trajectory experiments. The proposed method offers a robust solution for trajectory planning under complex geological conditions, advancing intelligent coal mining technology. The main contributions of this work including the following:

(1) A cutting path planning method based on an improved NSGA-II is proposed to address the issue of inadequate consideration of cutterhead and target constraints in current underground coal mine trajectory planning, which hinders precise cutting control of the section. This method lays the groundwork for further advancements in autonomous cutting control.

(2) A design method for cutting space based on kinematic constraints is introduced, which involves constructing a simplified cutterhead model and establishing a target section constraint model. By taking into account the mechanical characteristics and movement space of the boom-type roadheader, the cutting space driven by the fuselage pose is designed to identify feasible cutting points within the current section.

(3) A bi-objective optimization method for cutting trajectories is presented, aiming to minimize both the total length and the total angle. This problem is formulated as a bi-objective traveling salesman problem with adjacency constraints, and an improved NSGA-II algorithm incorporating adjacency constraints is employed to optimize and solve it, generating an executable cutting trajectory.

2. Methodology

2.1. Framework Overview

To address the insufficient consideration of cutterhead and target section constraints in current section formation planning methods, which hampers roadway formation quality, this study proposes a cutting trajectory planning method for roadway section formation in a boom-type roadheader based on an improved NSGA-II. The detailed framework is illustrated in Figure 1.

Based on the roadway coordinate system, a kinematic model of the roadheader was established to associate the roadheader, target section, and roadway. A target section model and its constraints were developed, along with a simplified cutterhead model, to calculate the spatial coordinates of the cutterhead envelope, providing a theoretical basis for section formation cutting. On this basis, the cutting space satisfying kinematic constraints was designed by integrating the kinematic model and motion range under posture-driven conditions, and the feasible cutting point set for the current posture was generated.

By analyzing and summarizing the section formation cutting process of the roadheader, the optimization of the cutting space point set was transformed into a multi-objective optimization problem, solved using an improved NSGA-II. First, optimization objective functions for cutting space points were constructed, adjacency constraints were designed, and an adjacency matrix was established. The adjacency matrix was then incorporated into the initialization and genetic operations of NSGA-II for adaptive improvements,

enhancing optimization efficiency and enabling the rapid acquisition of Pareto-optimal solutions. The resulting high-quality cutting trajectories provide a data foundation for further development of section formation cutting control.

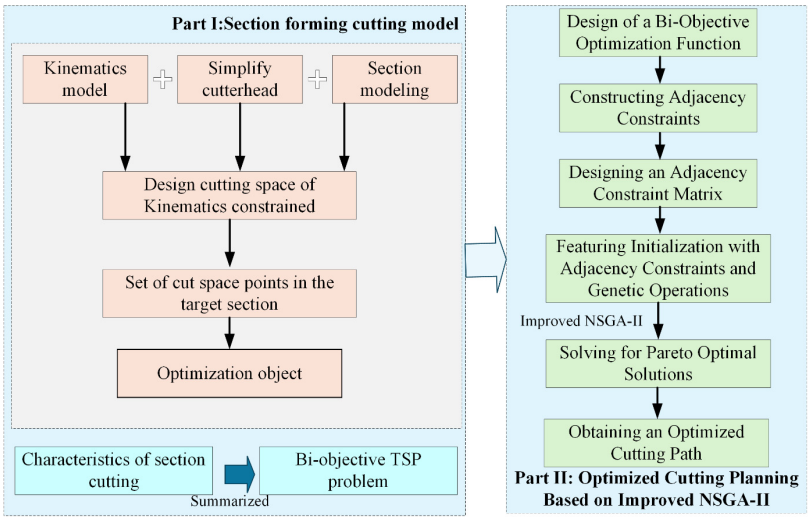

Figure 1. Framework of cutting trajectory planning based on improved NSGA-II for boom-type roadheader.

2.2. Design of Cutting Space Under Kinematic Constraints

2.2.1. Development of Roadheader Kinematic Model

The boom-type roadheader comprises a crawler-mounted mobile body and a longitudinal-axis cutting arm. Upon reaching the preset position, the roadheader engages the cutting arm to initiate excavation. To facilitate the planning and control of the section forming trajectory, a global coordinate system within the excavation tunnel for the roadheader is established, along with a corresponding kinematic model.

In order to accurately describe the relative pose of the roadheader and its working components within the tunnel, we have defined several coordinate systems: the tunnel coordinate system $O_hX_hY_hZ_h$, the section coordinate system $O_dX_dY_dZ_d$, the roadheader body coordinate system $O_oX_oY_oZ_o$, and the cutting arm coordinate system $O_iX_iY_iZ_i(i = 1, 2, 3, 4)$. The relationship between these coordinate systems is shown in Figure 2.

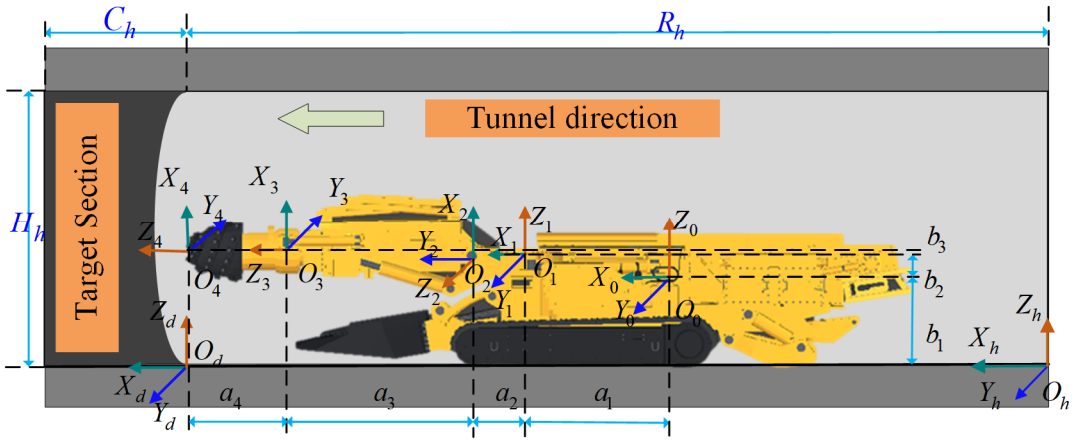

Figure 2. Kinematic model of roadheader.

Where C_h, H_h and R_h, respectively, represent the depth of the target section, the height of the roadway and the length of the roadway that has been cut. The $a_1 \sim a_4$ and $b_1 \sim b_3$ represent the DH parameter of the boom-type roadheader. θ_1, θ_2 and d represent the

lifting angle of the cutting arm, the turning angle and the elongation of the cutting head, respectively.

We derived the transformation matrices among the various coordinate systems of the roadheader to describe the movement and positioning more precisely. The transformation matrix between the cutting head coordinate system $O_iX_iY_iZ_i(i = 1,2,3,4)$ and the roadheader body coordinate system $O_oX_oY_oZ_o$ is derived as follows:

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & b_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2 \\ 0 & 0 & 1 & 0 \\ -\sin\theta_2 & -\cos\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & b_3 \\ 0 & 0 & 1 & a_3+d \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} {}^{3}_{4}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

We derived the homogeneous transformation matrix for the cutterhead relative to the roadheader body coordinate system as:

$$ {}^{0}_{4}T = {}^{0}_{1}T{}^{1}_{2}T{}^{2}_{3}T{}^{3}_{4}T \quad (2) $$

We set the pose ${}^{h}_{0}T$ of the roadheader robot in the tunnel coordinate system, and define the corresponding transformation matrix:

$$ {}^{h}_{0}T = \begin{bmatrix} {}^{h}_{0}R & {}^{h}_{0}p \\ 0 & 1 \end{bmatrix} \quad (3) $$

where ${}^{h}_{0}p = \begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T$, ${}^{h}_{0}R = R_zR_yR_x$.

We calculated the transformation matrix ${}^{h}_{4}T = {}^{h}_{0}T{}^{0}_{4}T$ of the cutting head coordinate system relative to the tunnel coordinate system, and utilized this matrix to determine the position of the cutting head in the tunnel coordinate system. Subsequently, we further solved the forward kinematics problem, namely, solving for the coordinates of the cutting head center point and the envelope surface in the cross-section coordinate system based on the known joint variables.

$$ {}^{d}_{4}P = {}^{d}_{h}T{}^{h}_{4}T = \left({}^{h}_{d}T\right)^{-1}{}^{h}_{4}T{}^{4}P \quad (4) $$

To solve the inverse kinematics, we employed analytical methods, numerical iterative methods, or intelligent algorithms. These methods calculate the joint variables required for the robot to move to a desired pose based on a given target pose. Before solving inverse kinematics, it is essential to define the target pose of the end effector. The boom-type roadheader operates within specific cross-sectional spaces, such as rectangular, trapezoidal, and semicircular arches, which differ significantly from point-to-point movements in tasks like welding and spraying. Therefore, the precise control of swing and telescoping of the cutting arm is required, particularly in narrow tunnel spaces. However, the roadheader is an underactuated mobile robotic arm, consisting of both a mobile mechanism and an actuating mechanism, making it challenging to directly design the target pose based on the target workspace.

To simplify the cutting head model, we calculated the motion space under the current pose of the roadheader body. This space is optimized by constructing a cross-sectional constrained space model to determine the feasible cutting space, which can be used to guide the planning and control of the cutting trajectory.

2.2.2. Simplified Cutterhead Modeling

Figure 3 depicts the actual cutting section of the roadheader robot. Based on the general principles of roadheader design, this paper simplifies the complex curved surface

of the cutting section to a geometric surface formed by rotating a curve, which is composed of a parabola and line segments, around a rotational axis.

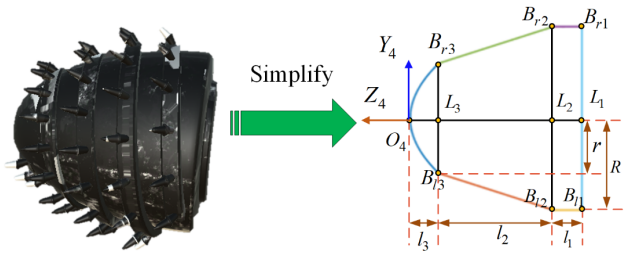

Figure 3. Projection model of cutterhead.

A piecewise curve Γ, composed of a parabola and line segments, can be expressed as:

$$\Gamma: \begin{cases} l_3 y^2 = r^2 z & -r < y \le 0, -l_3 < z \le 0 \\ l_2 y = (R-r)(l_2 + l_3) + (R-r)z & -R < y \le -r, -(l_2+l_3) < z \le -l_3 \\ y = -R & -(l_1+l_2+l_3) \le z \le -(l_2+l_3) \end{cases} \quad (5)$$

where R denotes the large radius of the cutting section, r the small radius, and $l_1 + l_2 + l_3$ the total length of the cutting head. By rotating this piecewise curve around the rotational axis O_4L_1 (designated as the Z_4-axis) of the cutting head coordinate system for one full circle, an outer envelope surface of the cutting head is formed, denoted as surface Π.

$$\Pi: \begin{cases} l_3(x^2+y^2) = r^2 z, -r < y \le r \\ l_2^2 (x^2+y^2) = ((R-r)(l_2 + l_3) + (R-r)z)^2, -R < y \le -r \text{ and } r < y \le R \\ x^2 + y^2 = R^2, -R < y \le R \end{cases} \quad (6)$$

2.2.3. Mathematical Modeling of Target Section

The primary focus of research is on rectangular roadway tunneling, as it constitutes the main type in coal mine tunneling. As illustrated in Figure 4, the rectangular roadway cross-section model is composed of a three-dimensional space enclosed by multiple planes $\pi_1 \sim \pi_6$. By precisely controlling the movement trajectory of the cutting head within this space, the target roadway can be accurately excavated.

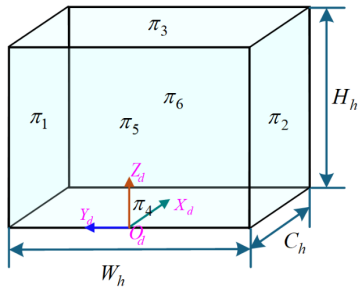

Figure 4. Schematic diagram of cross-sectional model.

The roadway width W_h, height H_h, and specified cutting depth C_h were set. The tunneling process is regarded as forming a rectangular prismatic space through multiple cycles of cutting. Let R_h represent the distance from the current cross-section to the starting point of the roadway. Ideally, R_h equals the product of the number of cuts n and the cutting depth C_h, where n denotes the cutting sequence starting from the origin. Hence, the coordinates of any point within the cross-section space can be denoted as $p_d = (x_d, y_d, z_d)$, where $0 < x_d < C_h$, $-0.5W_h < y_d < 0.5W_h$.

Based on the kinematic model of the boom-type roadheader, the roadway coordinate system differs from the cross-section coordinate system only in terms of an offset R_hx along the x-axis, with the other coordinate axes being consistent. Accordingly, the transformation matrix ${}^{h}_{d}T$ is expressed as:

$$ {}^{h}_{d}T = \begin{bmatrix} 1 & 0 & 0 & R_h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7} $$

In the roadway coordinate system, the coordinates of any point p_d in the cross-section coordinate system are $P_d = {}^{h}_{d}T \cdot p_d$. For non-full-section excavation, a minimum of two cuts are required to complete the excavation of a single cross-section. In this paper, the cross-section is divided into left and right parts based on the roadway centerline, and Left-Section and Right-Section models are constructed accordingly as $P_{d\_left} = (x_d, y_{left}, z_d)$, $P_{d\_right} = (x_d, y_{right}, z_d)$. Specifically, $P_{d\_left} = (x_d, y_{left}, z_d)$ represents the left cross-section model, while $P_{d\_right} = (x_d, y_{right}, z_d)$ represents the right cross-section model.

2.2.4. Generation of Trajectory Space Under Section Constraints

In the absence of cutting section constraints, the cutting arm moves within the range of the joint space, forming the motion space at the current pose. This motion space intersects with the target cutting section, resulting in the desired cutting space.

Let the motion space of the cutting head vertex under the current roadheader pose be denoted as $P_i(1, 2, 3 \cdots n)$, with the corresponding joint space $\theta_i(1, 2, 3 \cdots n)$. The cross-section model is given in Equation (8). Solving the motion space involves finding the mapping relationship between $P_i(1, 2, 3 \cdots n)$ and $\theta_i(1, 2, 3 \cdots n)$. For any point $(x\_left_j$, $y\_left_j, z\_left_j)$ within the coordinate set $P_{solution}$, it must satisfy $(j = 1, 2, \cdots, j < n)$.

$$ \begin{cases} x\_left_j \le R_h + C_h \\ y\_left_j \ge -0.5 * W_h, and, y\_left_j \le 0 \\ z\_left_j \ge 0, and, y\_left_j \le 0.5 * H_h \end{cases} \tag{8} $$

Similarly, the constraint conditions for the right-side cross-section can be determined based on the current pose of the roadheader and the shape and position of the right-side cross-section.

$$ \begin{cases} x\_right_j \le R_h + C_h \\ y\_right_j \ge 0, and, y\_right_j \le 0.5 * W_h \\ z\_right_j \ge 0, and, z\_right_j \le H_h \end{cases} \tag{9} $$

Based on the spatial range of each joint of the cutting arm, the forward kinematics model is utilized to set joint space increments $(\Delta\theta_1, \Delta\theta_2, \Delta d)$. Starting from the initial state, spatial point sets are screened according to cross-section constraint conditions to obtain feasible cutting space under the current roadheader pose. During this process, the size constraint of the cutting head must be considered, with calculations performed using the spatial coordinates of the cutting head's outer envelope surface.

Figure 5 illustrates the design process of the cutting trajectory space under target cross-section constraints. Given the pose of the roadheader body, the inherent motion space is calculated based on the motion range of the cutting arm joints, and then cutting trajectory points satisfying the constraints are screened in combination with target cross-section constraints. Adding the outer envelope surface of the cutting head forms an effective cutting space. However, the obtained cutting trajectory points are disordered and cannot directly guide the movement of the roadheader, requiring further optimization.

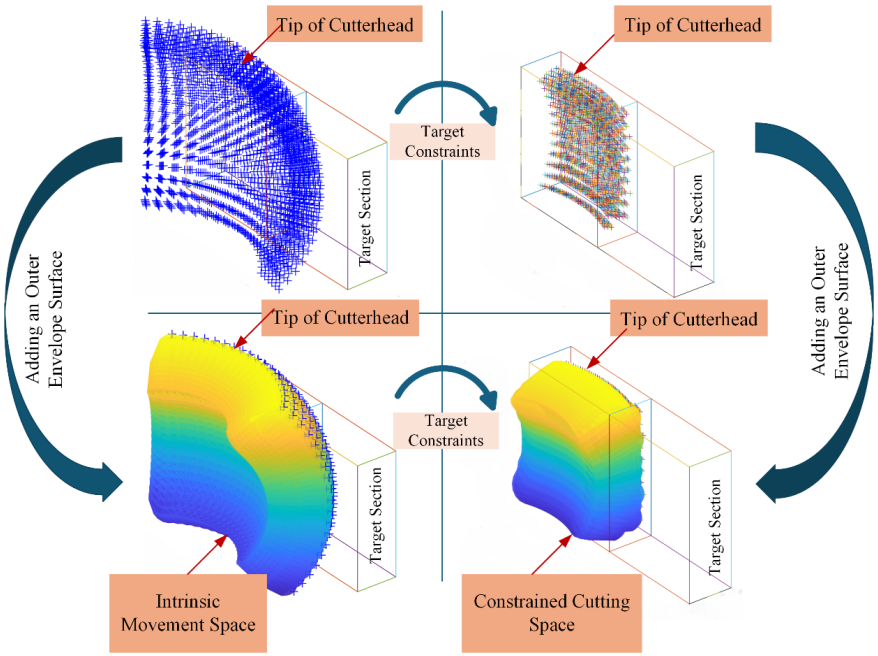

Figure 5. Design of cutting trajectory space under target cross-section constraints.

The cutting space is derived from joint space parameters, encompassing spatial coordinates of the end effector and its envelope surface. The trajectory points represent the spatial positions of the robot's joints, allowing for direct control without the need for frequent recalculations of inverse kinematics during operation, thus improving both control efficiency and accuracy.

2.3. Optimizing Cutting Paths Using Improved NSGA-II

2.3.1. Objective Function Formulation for Path Optimization

The cutting space is derived from joint space parameters, encompassing spatial coordinates of the end effector and its envelope surface. The trajectory points embody spatial joint information, eliminating the need for frequent inverse kinematics solutions during control, thereby enhancing control efficiency and accuracy.

$$\min F(x) = \min(F_1(x), F_2(x)) \tag{10}$$

where the key parameters include the shortest path $F_1(x) = L(\pi)$ and the minimum turning angle $F_2(x) = T(\pi)$.

The trajectory point set $\pi = (p_{\pi(1)}, p_{\pi(2)}, \cdots, p_{\pi(m)})$ is defined as the collection of all path points during roadheader operation. The total length of the cutting trajectory, $L(\pi)$, is the sum of distances between consecutive path points.

$$L(\pi) = \sum_{k=1}^{n-1} d(p_{\pi(k)}, p_{\pi(k+1)}) + d(p_{\pi(n)}, p_{\pi(1)}) \tag{11}$$

The total turning angle, $T(\pi)$, is the sum of angle differences between consecutive path points, calculated using a method based on spatial vectors of joint variables. For any consecutive trajectory points corresponding to joint variables $\theta_{\pi(k-1)}$, $\theta_{\pi(k)}$, $\theta_{\pi(k+1)}$, $2 \leq k \leq n-1$, the direction vector v_k from $\theta_{\pi(k)}$ to $\theta_{\pi(k+1)}$ is defined, and based on these vectors, the total turning angle $T(\pi)$ of the entire path is calculated.

$$T(\pi) = \sum_{k=2}^{n-1} T_{k \to k+1} + T_{n \to 1} \tag{12}$$

where $T_{k\to k+1} = \begin{cases} \left|\cos^{-1}\left(\frac{v_k \cdot v_{k+1}}{|v_k| \cdot |v_{k+1}|}\right)\right|, & \frac{v_k \cdot v_{k+1}}{|v_k| \cdot |v_{k+1}|} < 1 \\ 0, & \frac{v_k \cdot v_{k+1}}{|v_k| \cdot |v_{k+1}|} = 1 \end{cases}$.

Optimization of the cutting trajectory must meet the section forming cutting characteristics: the path should cover all trajectory points, ensuring each is traversed accurately once for completeness; trajectory points should be adjacent to ensure cutting continuity, as the target section is composed of solid coal walls and cannot jump; the starting point should be located on the outermost cutting plane of the section.

2.3.2. Optimization Procedure for Cutting Paths

Trajectory point optimization is essentially the solution to the bi-objective traveling salesman problem (TSP). The bi-objective TSP is a combinatorial optimization challenge aiming to balance conflicting objectives such as minimizing travel distance and maximizing customer satisfaction. Solution methods include the weighted sum method, ε-constraint method, and multi-objective evolutionary algorithms [23]. NSGA-II, an improved genetic algorithm, excels in multi-objective optimization by efficiently managing multiple objectives through rapid non-dominated sorting, crowding distance assessment, and elitist selection, enhancing solution set convergence and diversity. Adaptive improvements to NSGA-II are made to address the cutting trajectory optimization problem.

In the NSGA-II algorithm, setting the sharing radius relies on experience, which is a limitation. To address this, the concept of crowding distance is introduced. As illustrated in Figure 6, points marked within solid circles belong to the same non-dominated front. Crowding distance reflects the density of individuals in space, defined as the perimeter of the quadrilateral formed by adjacent solutions $i-1$ and $i+1$. A larger crowding distance indicates a sparser area around the solution, facilitating the maintenance of population diversity.

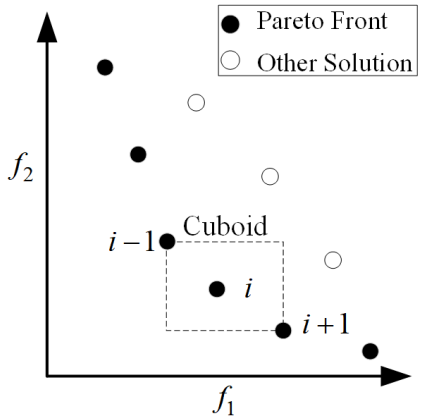

Figure 6. Crowding distance calculation.

The NSGA-II algorithm incorporates an elite strategy to retain superior individuals and eliminate inferior ones, expanding the selection scope for the next generation by merging parental and offspring individuals into a new population, as illustrated in Figure 7.

The direct application of the NSGA-II algorithm falls short in addressing the cutting trajectory problem. Hence, an improved NSGA-II algorithm flowchart, specifically designed for the optimization of cutting trajectories, is presented in Figure 8.

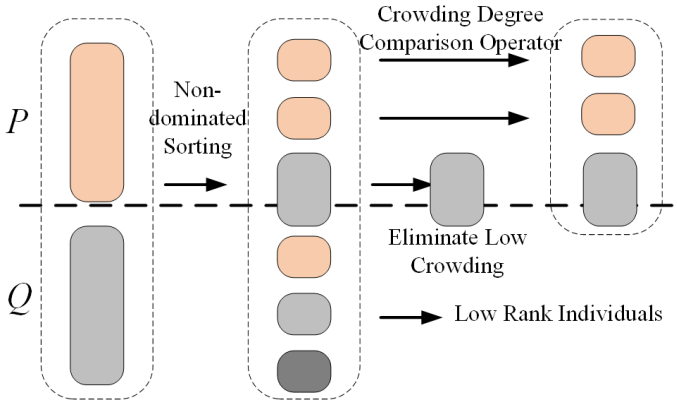

Figure 7. Elite strategy execution steps.

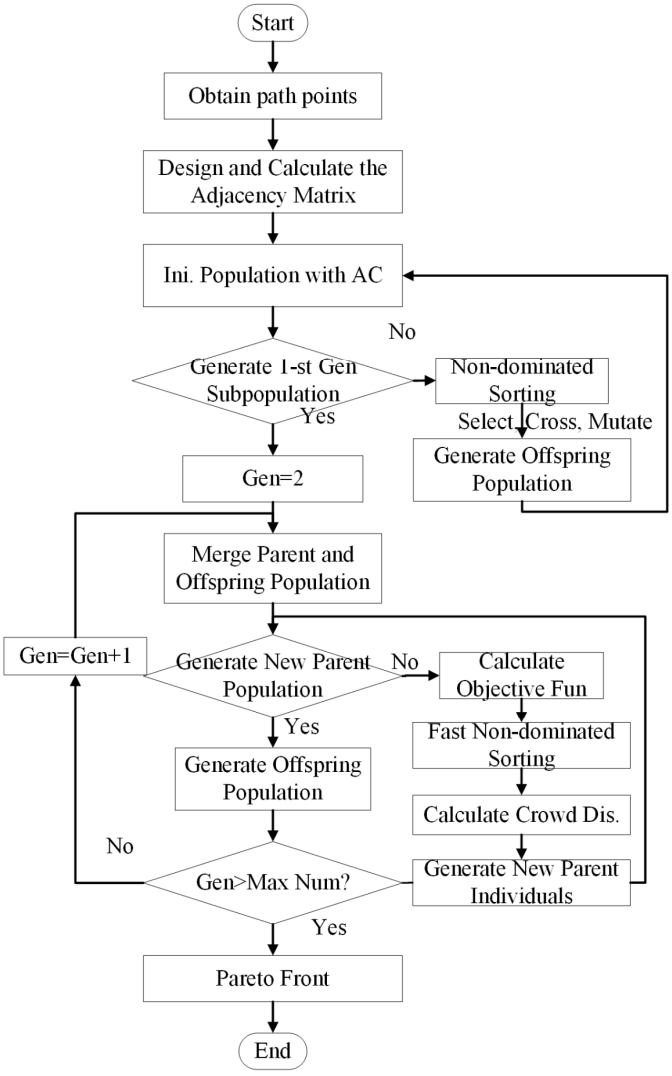

Figure 8. Flowchart of improved NSGA-II algorithm with adjacency constraints.

The specific procedure is as follows:

(1) Obtain Path Points and Design Adjacency Matrix. The process begins by obtaining the feasible cutting space for the roadheader under its current pose. The path points are then numbered, and an adjacency constraint model is established in the joint space. The adjacency matrix is computed based on these constraints, which governs the connections between path points.

(2) Initialize Population. The population is initialized, with the first generation (Gen = 1) being created. However, since the standard NSGA-II algorithm initializes the population randomly, it does not fully respect the adjacency constraints of the roadheader. Therefore, adaptive improvements are introduced to ensure the population initialization is feasible with respect to the constraints.

(3) Generate First Generation of Offspring Population. If the first generation of offspring has already been generated, the algorithm proceeds by setting Gen = 2. Otherwise, non-dominated sorting, selection, crossover, and mutation are performed on the initial population to generate the first generation of offspring. The crossover and mutation processes are modified to adapt to the kinematic constraints of the roadheader.

(4) Merge Parent and Offspring Populations. The parent and offspring populations are merged into a new population. If a new parent population has not been generated, the algorithm calculates the objective functions of the individuals in the new population. Then, it performs rapid non-dominated sorting, crowding distance calculation, and applies an elite strategy to generate the new parent population.

(5) Generate New Offspring Population. Once a new parent population is generated, selection, crossover, and mutation operations are performed on the parent population to generate the offspring. This process is again constrained by the adjacency model to maintain feasibility for the roadheader's movement.

(6) Check Evolution Generation Limit. The algorithm checks whether the generation count (Gen) has reached the maximum allowed number. If not, it increments the generation count and loops back to Step 3 to repeat the process. If the maximum number of generations is reached, the algorithm terminates.

2.3.3. Adjacent Constraints Design for Cutting Space Points

In robotics, joint space is defined as the space with joint parameters as coordinate axes. The state of each joint varies independently, without directly considering the position of the end effector in Cartesian space. This paper establishes adjacency constraints using joint parameters as the basis, focusing primarily on the relative positional relationships and kinematic constraints among joints. Although joint space does not directly utilize the concept of geometric neighborhoods, the notion of "neighborhood" can be extended in this context.

Let the coordinates of the set of spatial points be $M = \{m_1, m_2, \ldots, m_n\}$, where each spatial point $m_1 = (x_i, y_i, z_i)$ corresponds to a joint space point $Q = \{q_1, q_2, \ldots, q_n\}$. Specifically, spatial point $m_1 = (x_i, y_i, z_i)$ corresponds one-to-one with joint space point $q_1 = (\theta_{i1}, \theta_{i2}, d_i)$, where q_1 represents the joint parameters of the cutting part. This means that for each spatial point m_i, there is a unique joint space point q_i, and the positioning of the spatial point is achieved through the joint space parameters q_i.

The adjacency matrix **A** is used to represent the adjacency relationship between spatial points. The adjacency matrix **A** is defined as:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \tag{13}$$

where m_i and m_j represent adjacent points $a_{ij} = 1$, and the motion corresponding to joint points q_i and q_j must satisfy the continuity constraint. m_i and m_j indicate $a_{ij} = 0$ that there is no adjacency relationship between them.

When considering a single joint, its neighborhood can be defined as the two states resulting from adding or subtracting a small increment to the current joint angle. Since the three joints of the cantilever roadheader are independent, they can be controlled independently. By adding or subtracting a small increment $\Delta\theta$ to the current joint space coordinate $O(\theta_1, \theta_2, d)$ along the rotational joint's direction of motion, two states $A(\theta_1 + \Delta\theta, \theta_2, d)$ and $B(\theta_1 - \Delta\theta, \theta_2, d)$ are generated. Similarly, by adding or subtracting $\Delta\theta$ along the lifting joint's direction of motion, two states $C(\theta_1, \theta_2 + \Delta\theta, d)$ and $D(\theta_1, \theta_2 - \Delta\theta, d)$ are produced. By adding or subtracting a small increment Δd along the telescoping joint's direction of motion, two states $E(\theta_1, \theta_2, d + \Delta d)$ and $F(\theta_1, \theta_2, d - \Delta d)$ are generated. These states are combined to form the adjacency constraint model in joint space, as illustrated in Figure 9.

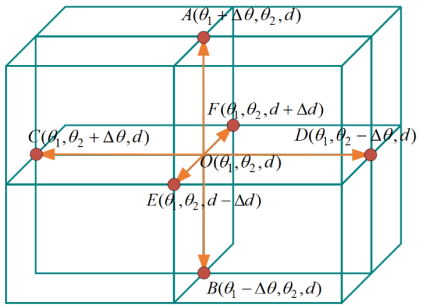

Figure 9. Model of adjacency constraints in joint space.

At this point, the changes of points q_i and q_j in the θ_1-direction and θ_2-direction satisfy

$$\left|\theta_{i1} - \theta_{j1}\right| \leq \Delta\theta, \left|\theta_{i2} - \theta_{j2}\right| \leq \Delta\theta \tag{14}$$

where $\Delta\theta$ represent the maximum change in the θ_1-direction and θ_2-direction, respectively. This condition can be used to determine the adjacency conditions for the joint rotation directions of θ_1 and θ_2. The change in points q_i and q_j in the direction of the cutterhead's telescoping motion satisfies:

$$\left|d_i - d_j\right| \leq \Delta d \tag{15}$$

where Δd represents the maximum allowable change in the cutterhead's telescoping direction. Therefore, the adjacency relationship between points q_i and q_j holds if and only if all the above conditions are satisfied, that is:

$$a_{ij} = \begin{cases} 1, & \left|\theta_{i1} - \theta_{j1}\right| \leq \Delta\theta, \left|\theta_{i2} - \theta_{j2}\right| \leq \Delta\theta, \left|d_i - d_j\right| \leq \Delta d \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

By calculating the adjacency values of all spatial points, the adjacency matrix **A** can be obtained.

Assuming bidirectional and unordered trajectory points with uniform cost, the creation of adjacency lists and adjacency matrices hinges on the criterion for adjacency determination. This paper employs the joint space adjacency model depicted in Figure 10 for trajectory points as the basis for adjacency judgment. For instance, trajectory points "1" and "4" are adjacent to trajectory point "0", and thus are classified into the adjacency list of point "0". Similarly, other trajectory points are classified accordingly. A 16 × 16 matrix is constructed, where, based on the established adjacency lists, the corresponding positions in the matrix are set to 1 if adjacent, and to 0 otherwise.

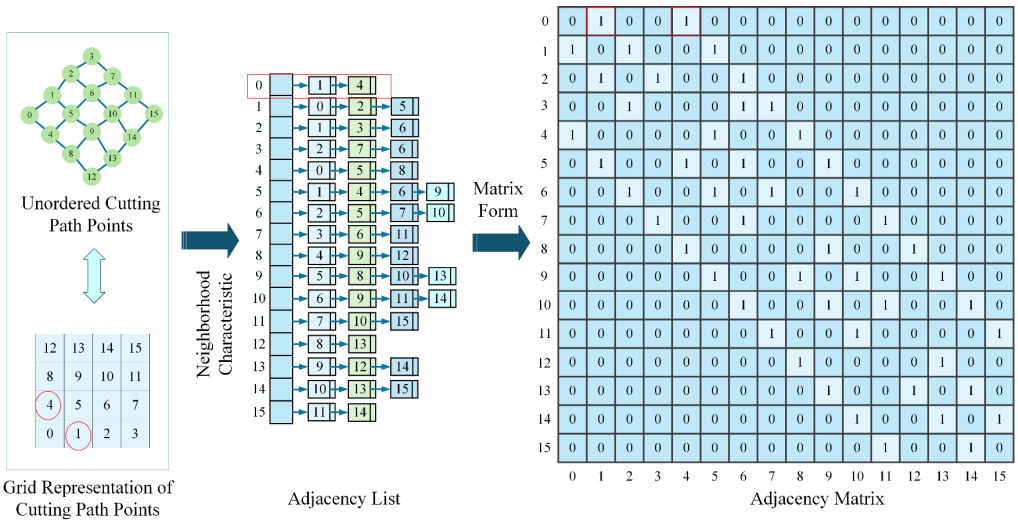

Figure 10. Representation of path points on grid and generation of adjacency list and matrix.

Adjacency lists and adjacency matrices are crucial components in the optimization of cutting trajectories. In the context of high-dimensional path optimization, the rational design of adjacency lists and matrices can significantly enhance efficiency. Furthermore, they provide a foundation for subsequent operations such as initialization and mutation using the NSGA-II algorithm.

2.3.4. Application of Improved NSGA-II to Path Optimization

In this paper, we present the application of depth-first search (DFS) with adjacency constraints for initialization, crossover, and mutation operations in pathfinding. DFS is initiated from the starting path point and sequentially searches for adjacent, unvisited nodes until all potential paths are identified. However, its efficiency diminishes in the context of higher-dimensional path sequences. To mitigate this, we introduce pruning operations based on the unique constraint that each cutting trajectory point can only be traversed once. The pruning procedure entails updating the adjacency list by removing visited nodes, identifying connected components among unvisited nodes, and assessing the size of each component. If a component's size is smaller than the remaining nodes, it suggests fragmentation of the path, preventing the formation of a closed loop due to adjacency constraints. Consequently, the current node is bypassed for backtracking purposes. Additionally, this study introduces a DFS algorithm integrated with pruning optimization to refine the initialization and mutation processes, thereby ensuring path adjacency and maintaining high search efficiency.

(1) Population Initialization with Adjacency Constraints

The NSGA-II algorithm initializes its population using a random generation method, which introduces uncertainty. This randomness results in spatially uneven point sets, failing to guarantee adjacency constraints during each initialization. Given a population size m, NSGA- II randomly generates m path points, exhibiting randomness and uncertainty. While the path length is ensured, incomplete traversal and redundant traversal issues may arise, as illustrated in Figure 11. To address this, a depth-first search (DFS) algorithm with pruning optimization is employed for population initialization. A schematic diagram of one path after optimization is presented, demonstrating that the initialized path satisfies the predefined adjacency constraints without redundant traversal.

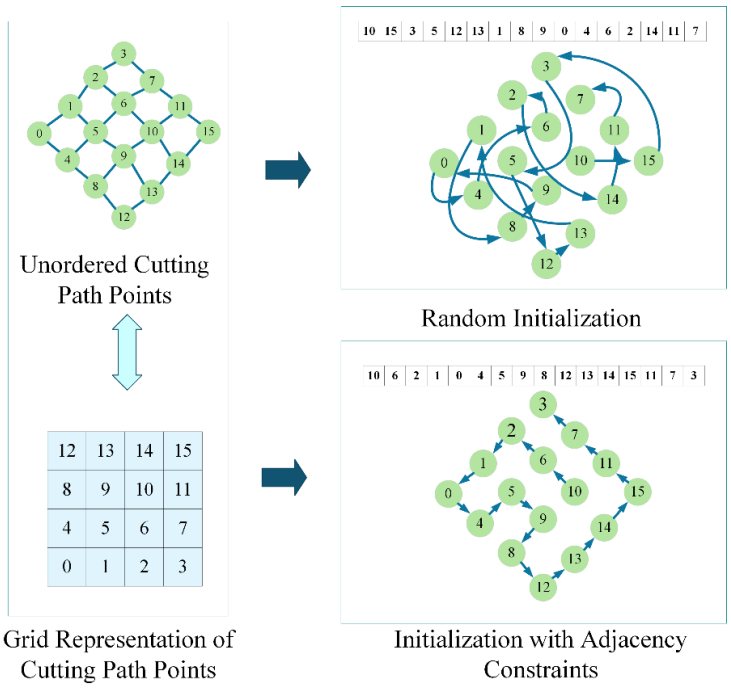

Figure 11. Comparison of initialization with adjacent constraints and random initialization.

(2) Mutation with Adjacency Constraints

In the mutation operation of NSGA-II, individuals from the current population are selected for mutation, with one or more genes on their chromosomes randomly chosen as mutation targets. The polynomial mutation operator is applied to these genes, resulting in mutated gene values that replace the corresponding values in the original individuals, thereby generating new mutated individuals. However, when performing mutation with adjacency constraints, it is necessary to ensure that the subsequent path points after the mutation point also satisfy adjacency.

Given the limitations of NSGA-II in crossover and mutation handling, further adjacency processing is required for the crossover gene segments or mutated gene sequences after these operations. Therefore, we combine these two processing steps. Although this approach sacrifices some population diversity, it significantly improves the efficiency of generating offspring gene sequences due to the adjacency constraints.

3. Simulations and Experiments

3.1. Simulation Verification

Path points spaced at intervals of 10 were selected, with their joint space coordinates provided in Figure 12. The adjacency matrix for this path set was computed based on the joint space adjacency constraint model. Simulation settings: population size of 128, 500 iterations. By comparing path optimization outcomes under different mutation and crossover probabilities, and considering convergence, runtime, minimum distance, and turning cost, a crossover probability of 0.7 and a mutation probability of 0.6 were selected as the simulation parameters.

3.1.1. Presentation of Optimization Results

Figure 13 presents the Pareto optimal solution and iterative convergence of trajectory point optimization using the improved NSGA-II path optimization algorithm.

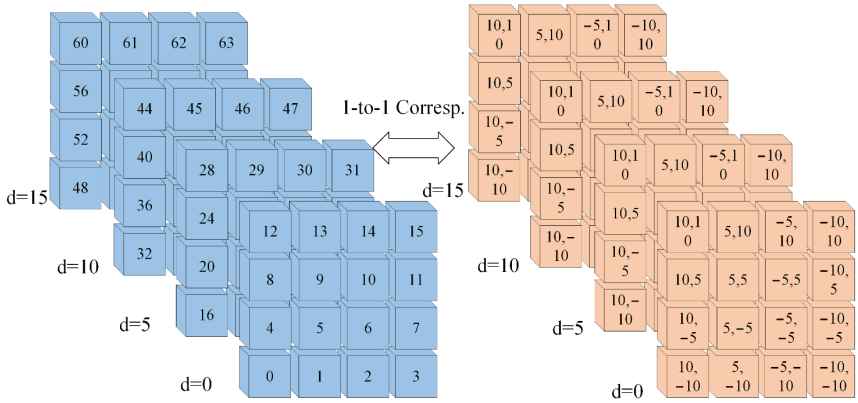

Figure 12. Path sets and their corresponding joint space coordinates.

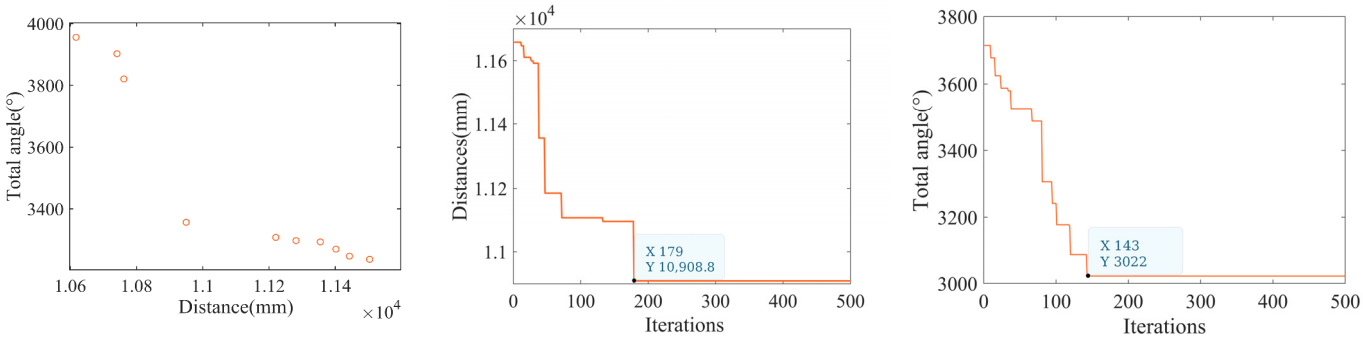

Figure 13. Improved NSGA-II trajectory point optimization results. (**a**) Pareto frontier solution; (**b**) Convergence of the total distance; (**c**) Convergence of total Angle.

Figure 13a presents the Pareto optimal front. For the objective functions of total distance and total turning angle, each has an optimal solution, but optimizing one may impact the other. Using the improved NSGA-II path optimization method, a set of Pareto optimal solutions were obtained, forming the Pareto front. Figure 13b shows that at 178 iterations, the total distance reached a minimum of 10,594.57; Figure 13c indicates that at 145 iterations, the total turning angle reached a minimum of 3023.74. This study verifies the convergence of the proposed method, and subsequent ablation experiments will further validate the effectiveness of the improvements.

3.1.2. Ablation Comparison

To validate the effectiveness of algorithm improvements, ablation comparison experiments were conducted, comparing initial population and crossover processes. $\sim (F + P_C + P_V)$: no improved initialization or crossover with adjacency constraints; F: only NSGA-II with adjacency-constrained initialization; $P_C + P_V$: only crossover with adjacency constraints; $F + P_C + P_V$: both improved initialization and crossover with adjacency constraints. Experimental results are shown in Figure 14, including convergence generations, optimal objective function value, and running speed. The improved method yields the best objective function value.

Through ablation experiments and statistical significance analysis, the proposed method demonstrates significant improvements over other modifications in terms of convergence speed, total path length, steering angle, and runtime ($p < 0.001$). Specifically, the proposed method reduces the number of iterations by 60.9% ($p < 0.001$). In terms of runtime, the proposed method reduces the time by 91.6% compared to other methods ($p < 0.001$).

The total path length and runtime of the proposed method are significantly better than those of the other improvements.

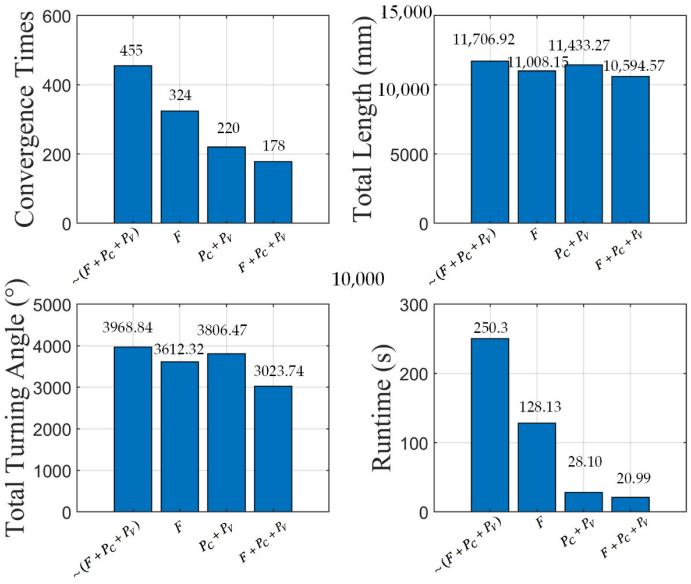

Figure 14. Ablation experiment results.

This study validates the effectiveness and superiority of the improved NSGA-II path optimization method in cross-sectional forming cutting trajectory optimization from both convergence and performance improvement perspectives. However, further verification is required for its specific application in boom-type roadheader path optimization. Given the complexity and difficulty of actual cutting path testing, a virtual debugging system is constructed in a virtual environment to verify the correctness of the cutting trajectory by comparing planning results with virtual motion results.

3.1.3. Comparative Simulation Validation

Under identical simulation conditions, a comparative analysis of the modified NSGA-II algorithm with PSO and ACO algorithms was conducted, focusing on convergence frequency, total path length, total turning angle, and runtime. The results of this comparison are presented in Figure 15.

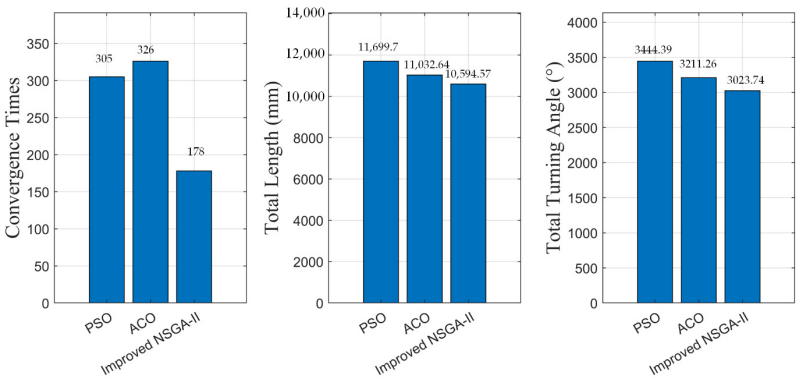

Figure 15. Comparative analysis of optimization results: improved NSGA-II versus PSO and ACO algorithms.

Under identical simulation conditions, the performance of the improved NSGA-II algorithm was compared with the PSO and ACO algorithms in terms of convergence times, total path length, total turning angle, and runtime. The results are presented in Figure 15.

To validate the comparative results, we conducted statistical significance testing using ANOVA ($p < 0.001$), Tukey HSD, and Kruskal–Wallis tests where appropriate. The analysis showed that the improved NSGA-II algorithm significantly outperforms the PSO and ACO algorithms across all evaluated metrics. The improved NSGA-II algorithm reduced convergence times by 127 iterations compared to PSO, and 148 iterations compared to ACO (Kruskal–Wallis test, $p < 0.001$). The improved NSGA-II algorithm achieved a 1105.16 mm reduction in total path length compared to PSO, and a 438.07 mm reduction compared to ACO (ANOVA, $p < 0.001$). The improved NSGA-II algorithm decreased the total turning angle by 420.65° compared to PSO, and 187.52° compared to ACO (ANOVA, $p < 0.001$). The runtime of the improved NSGA-II algorithm was reduced by 49.64% compared to PSO, and by 61.54% compared to ACO (Tukey HSD, $p < 0.001$).

Therefore, in terms of convergence times, running time, and trajectory results, the improved NSGA-II algorithm demonstrates superior performance, providing a basis for further application in boom-type roadheader cutting trajectory planning.

3.2. Virtual Simulation of Sectional Cutting Verification

To elucidate the impact of the body pose on the motion space of the roadheader, this paper establishes the size parameters for the boom-type roadheader, as detailed in Table 1. The motion ranges of the cutting arm include the rotation angle $\theta_2 \in [-45(^\circ), 45(^\circ)]$, lifting angle $\theta_1 \in [-135(^\circ), -45(^\circ)]$, and telescoping length $d \in [0, 500\ (\mathrm{mm})]$. Additionally, tunnel parameters $5 \times 3\ (\mathrm{m})$, target cross-section parameters: $R_h = 20\ (\mathrm{m})$, $C_h = 1\ (\mathrm{m})$, and cutting head size parameters ($R = 380\ (\mathrm{mm})$, $r = 266\ (\mathrm{mm})$, $l_1 = 84\ (\mathrm{mm})$, $l_2 = 343\ (\mathrm{mm})$, $l_3 = 122\ (\mathrm{mm})$) are defined.

Table 1. Dimensional parameters of roadheader.

Para.	a_1	a_2	a_3	a_4	b_1	b_2	b_3
Roadheader (mm)	1544	501.5	2011.5	962	983	290	13

To improve the quality of section forming, this paper adopts a segmented cutting method and constructs a feasible cutting trajectory space based on different roadheader body poses. The discrete path points are optimized using an improved NSGA-II algorithm to obtain feasible cutting trajectories. Three scenarios are compared and analyzed as follows:

Scenario 1: The roadheader body pose for left-side section cutting is set to $P = (15{,}563.5\ \mathrm{mm}, -1250\ \mathrm{mm}, 983\ \mathrm{mm}, 0^\circ, 0^\circ, 0^\circ)$, and for right-side section cutting, it is set to $P = (15{,}563.5\ \mathrm{mm}, 1250\ \mathrm{mm}, 983\ \mathrm{mm}, 0^\circ, 0^\circ, 0^\circ)$.

Scenario 2: The roadheader body pose for left-side section cutting is set to $P = (15{,}563.5\ \mathrm{mm}, -1250\ \mathrm{mm}, 983\ \mathrm{mm}, 0^\circ, 0^\circ, 0^\circ)$, and for right-side section cutting, it is set to $P = (15{,}503.5\ \mathrm{mm}, 1230\ \mathrm{mm}, 963\ \mathrm{mm}, 2^\circ, -2^\circ, 10^\circ)$.

Scenario 3: The roadheader body pose for left-side section cutting is set to $P = (15{,}503.5\ \mathrm{mm}, -1230\ \mathrm{mm}, 963\ \mathrm{mm}, 2^\circ, -2^\circ, 10^\circ)$, and for right-side section cutting, it is set to $P = (15{,}563.5\ \mathrm{mm}, 1230\ \mathrm{mm}, 1003\ \mathrm{mm}, 2^\circ, -2^\circ, -10^\circ)$.

Based on the above poses, an improved NSGA-II is used to plan the cutting trajectories, and the trajectory tracking effect and forming quality are verified on a virtual simulation platform. As shown in Figure 16, a virtual simulation platform for section forming and cutting is built, achieving coordinate mapping and scale unification between physical and virtual spaces. A cutting head with virtual collision detection function is used to simulate the cutting process.

Figure 16. Virtual simulation platform for forming.

3.2.1. Simulation of Non-Full-Section Cutting Paths Under Different Body Poses

In this paper, an improved NSGA-II algorithm is adopted for trajectory planning under three different roadheader body poses. Figure 17 presents the cutting trajectory planning results for three scenarios, including: (a) a three-dimensional comparison of the cutting head envelope space; (b) a comparison of the planned cutting trajectories; (c) a comparison of the joint variables for the left-side section cutting trajectories; and (d) a comparison of the joint variables for the right-side section cutting trajectories.

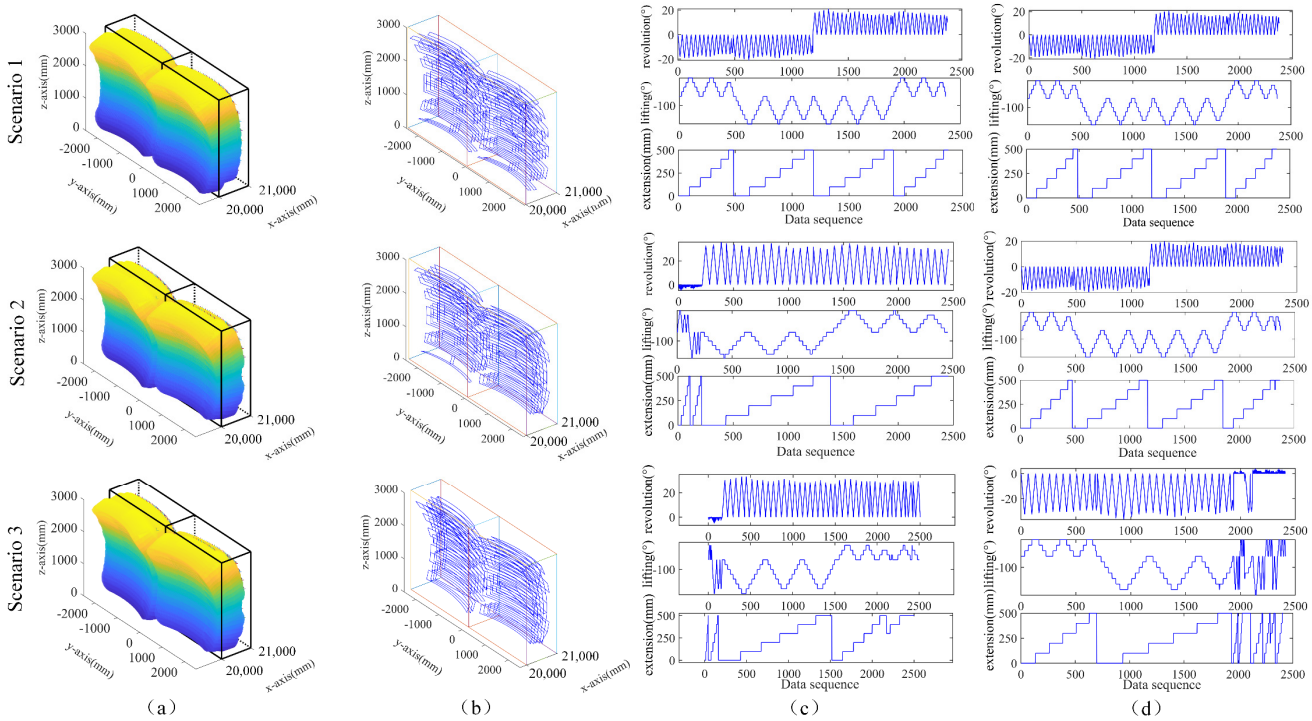

Figure 17. Comparison of cutting path planning results for three different poses. (**a**) Operational space. (**b**) Cutting path. (**c**) Joint variables for left section cutting path. (**d**) Joint variables for right section cutting path.

Figure 17a shows that the cutting spaces of the planned trajectories under different body poses differ from the target section. Segmented cutting trajectory planning can achieve full-section cutting, but changes in the body pose lead to changes in the cutting space. The left-side section cutting poses are the same for Scenario 1 and Scenario 2, but their right-side section cutting spaces differ. Similarly, the right-side section cutting poses are the

same for Scenario 2 and Scenario 3, with the same cutting space, but their left-side section spaces differ. Therefore, changes in the body pose affect the formation of the cutting space.

Figure 17b compares the cutting trajectories formed by the cutting head end coordinates under the three poses, consistent with Figure 17a, indicating that the cutting head end coordinates do not fully contact the target section, demonstrating the need to consider the size constraint of the cutting head for cutting planning. Figure 17c,d present the joint variables for the left-side and right-side section cutting trajectory planning, respectively, avoiding frequent calculation of inverse kinematics solutions during automatic cutting control.

In summary, the trajectory planning method proposed in this paper adapts to different roadheader body poses and ensures good cutting quality, but further verification and analysis are needed to determine the specific cutting quality.

3.2.2. Comparative Analysis of Virtual Trajectory Tracking

Based on the cutting trajectory planning results under three poses, we conducted virtual simulations in Unity3D. A unified virtual coordinate system was established, and virtual roadheader models and virtual sections were created. The virtual models were then driven for cutting operations using the obtained joint space motion parameters. By comparing the section forming quality and cutting trajectories, the validity of the cutting trajectories constructed using the proposed method was verified. Figure 16 presents the trajectory tracking and virtual forming effects.

Figure 18 shows that the virtual cutting trajectories are basically consistent with the planned trajectories. Under the three body poses, the virtual cutting results in good section forming effects, ensuring the quality of the section cutting and precise boundary control.

In the three poses, deviations exist between the virtual cutting tracking trajectories and the planned trajectories. Therefore, an in-depth analysis of the trajectory tracking errors is required to verify the forming cutting performance of the planned trajectories. To quantify the cutting effect, a forming quality evaluation index, namely the cutting volume ratio, $P_{FormQuality}$, is proposed.

$$P_{FormQuality} = \frac{V_{Cutdone}}{V_{all}}$$

Herein, $V_{Cutdone}$ denotes the cut volume, and V_{all} denotes the total section volume. Figure 19 presents a comparison of the cutting quality of section forming in virtual simulations. Under the three poses, the forming quality is good; however, as the pose deviation increases, the forming quality gradually decreases.

However, it is difficult to directly verify the effectiveness of trajectory planning solely through the forming process and quality. Therefore, it is necessary to compare the recorded virtual cutting trajectories with the planning results. Figure 20 presents comparisons of positions in various directions under three conditions, including data comparisons of cutting trajectories in the X-direction, Y-direction, and Z-direction.

Figure 20 shows that the coordinate data in each direction are basically consistent. The trajectory tracking results for scenario 1 are: maximum deviation in the X-direction is -13 mm, Y-direction is -4 mm, and Z-direction is -6 mm; for scenario 2: X-direction is 6 mm, Y-direction is 4 mm, and Z-direction is -5 mm; for scenario 3: X-direction is -9 mm, Y-direction is -3 mm, and Z-direction is -5 mm. In summary, regardless of the boom loader's pose, deviations exist between the virtual cutting trajectory and the planned trajectory, but their trends are consistent. From the perspective of forming quality and motion trend, the improved NSGA-II cutting trajectory planning method proposed in this paper is effective and can guide the automatic control process of the cantilever boom loader.

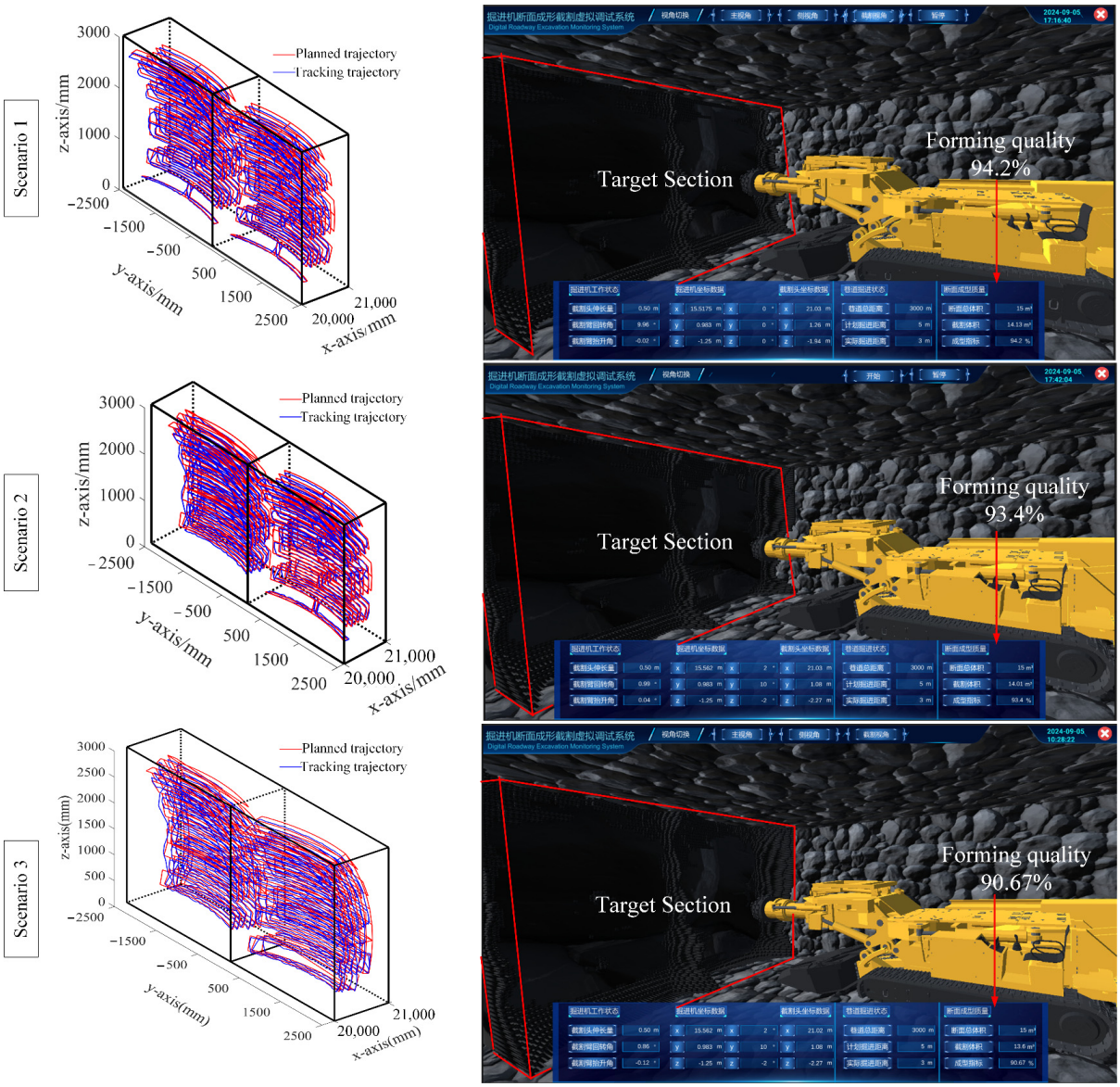

Figure 18. Evaluation of trajectory tracking and virtual shaping quality.

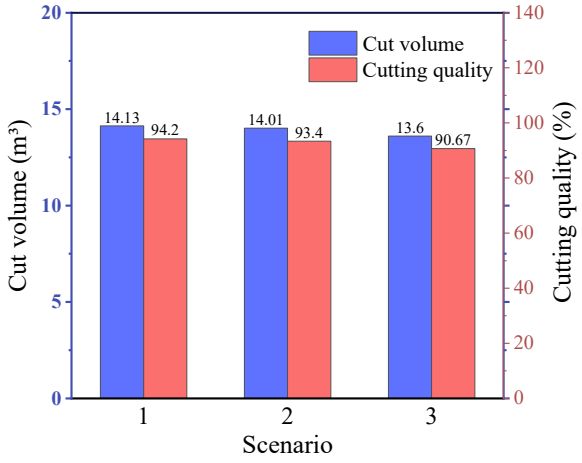

Figure 19. Assessment of sectional shaping quality.

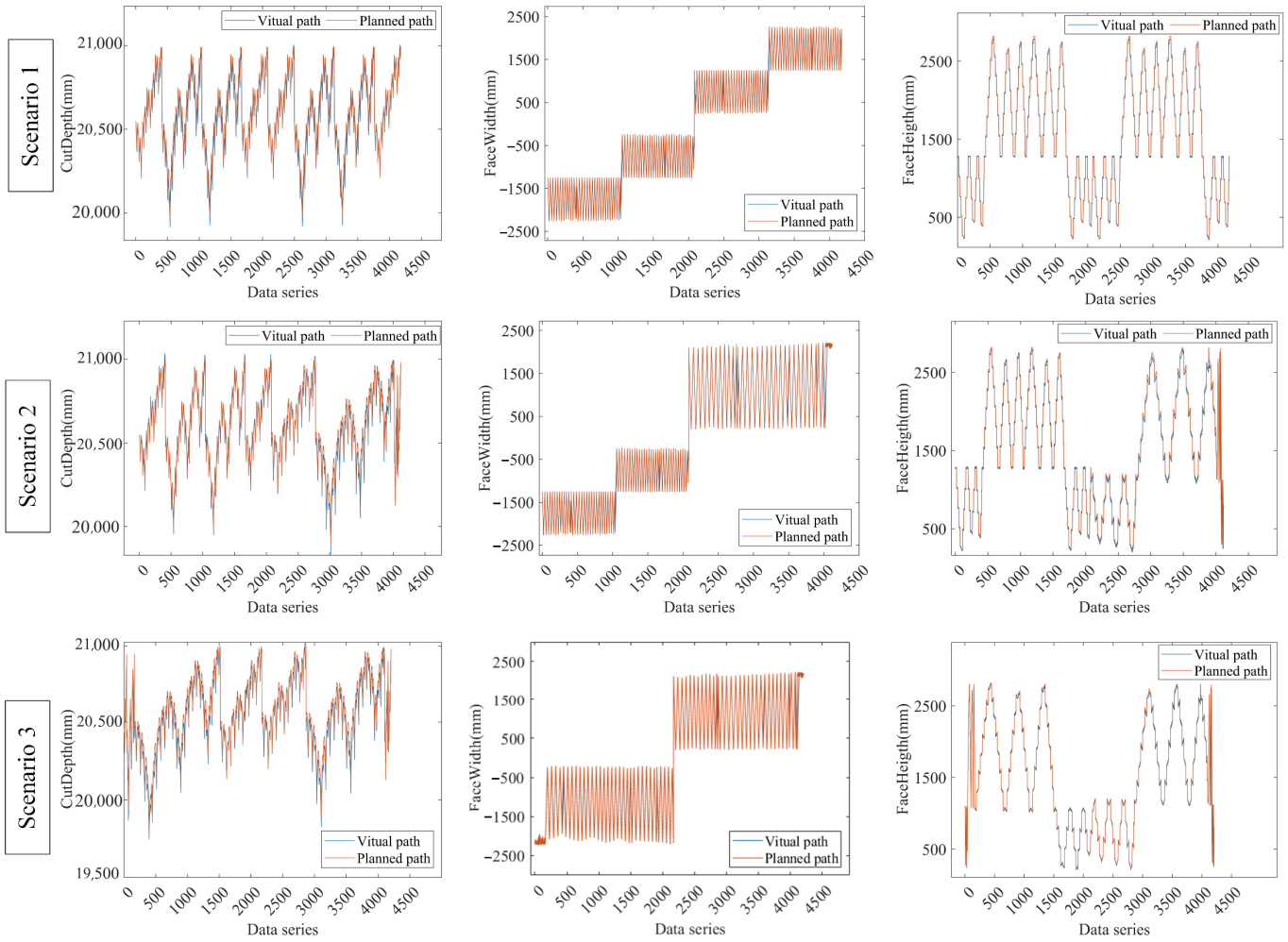

Figure 20. Comparison of planned and virtual simulation trajectories.

4. Experimental Validation

4.1. Experimental Platform Description

Figure 21 presents the experimental platform for validating cutting trajectory planning. The platform adopts the cutting trajectory planning method proposed in this paper, controls the roadheader to follow the planned path, and verifies the effectiveness of the method by comparing the differences between the actual and planned paths. The platform consists of a body vision positioning system [24], a cutting head positioning system, a control unit, and a target section (2 m × 1.9 m). The body vision positioning system includes explosion-proof industrial cameras, laser pointers, and an inertial navigation system. The cutting head positioning system includes displacement sensors for lifting, rotating, and telescoping cylinders, and calculates the attitude angle of the cutting arm using a mathematical model of the cutting arm joint mechanism.

4.2. Analysis of Experimental Results

Given the pose of the roadheader body, the cutting arm is controlled to move according to the planned trajectory. The left image in Figure 22 shows the cutting head moving autonomously to the boundary, while the right image compares the cutting trajectory with the planned trajectory. The results show that the trajectory tracking is consistent with the planning, achieving sectional forming cutting with good boundary control.

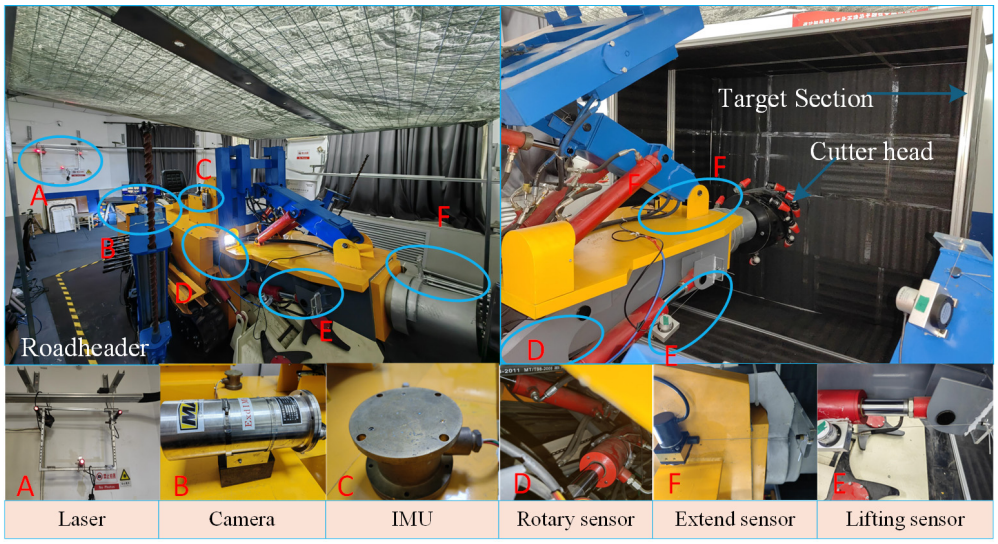

Figure 21. Verification platform for sectional shaping and cutting path planning.

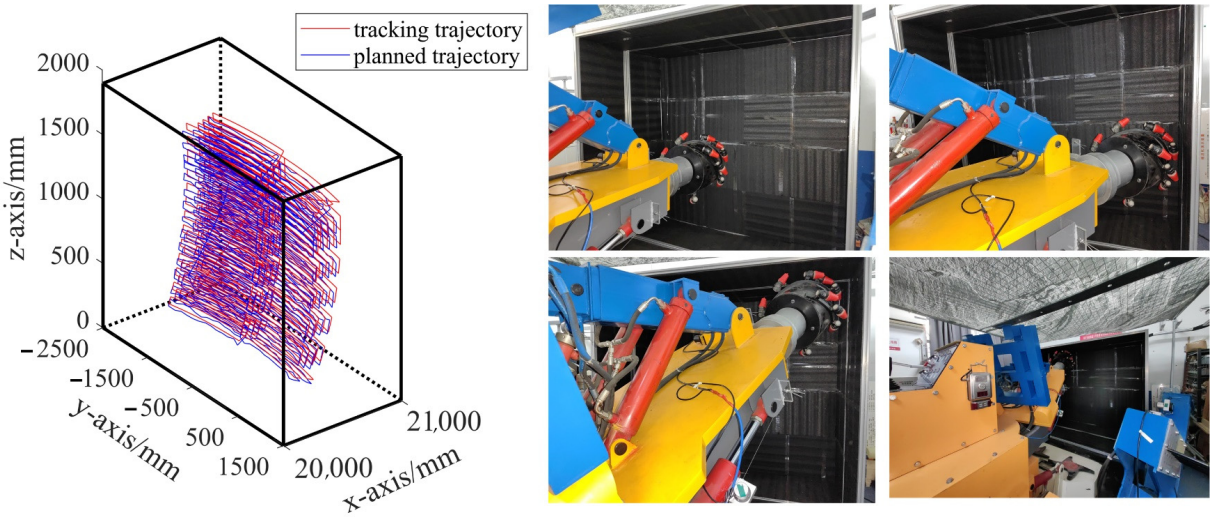

Figure 22. Comparison of cutting paths.

Figure 23 presents a comparative analysis of the cutting trajectory tracking in the x, y, and z directions. (a) shows the trajectory comparison and error in the x-direction, (b) in the y-direction, and (c) in the z-direction.

The experimental results show that the average tracking error in the X-direction is −7.17 mm, with a maximum error of 9.32 mm. In the Y-direction, the average error is −1.75 mm, with a maximum error of 23.88 mm, which is notably higher than in other directions. This significant deviation suggests that the *Y*-axis trajectory tracking may be more susceptible to certain challenges such as sensor noise or limitations in the algorithm's handling of vertical movements. In the Z-direction, the average error is −8.16 mm, with a maximum of 15.93 mm. While these deviations are within an acceptable range for most applications, further refinement of the planning algorithm and sensor fusion techniques is recommended to minimize these errors, particularly in constrained or complex environments. Overall, these results validate the cutting trajectory tracking algorithm and provide a strong foundation for further advancements in autonomous cutting technology. However, the higher deviations observed in the Y-direction warrant attention and could be addressed in future work to ensure more consistent performance across all axes.

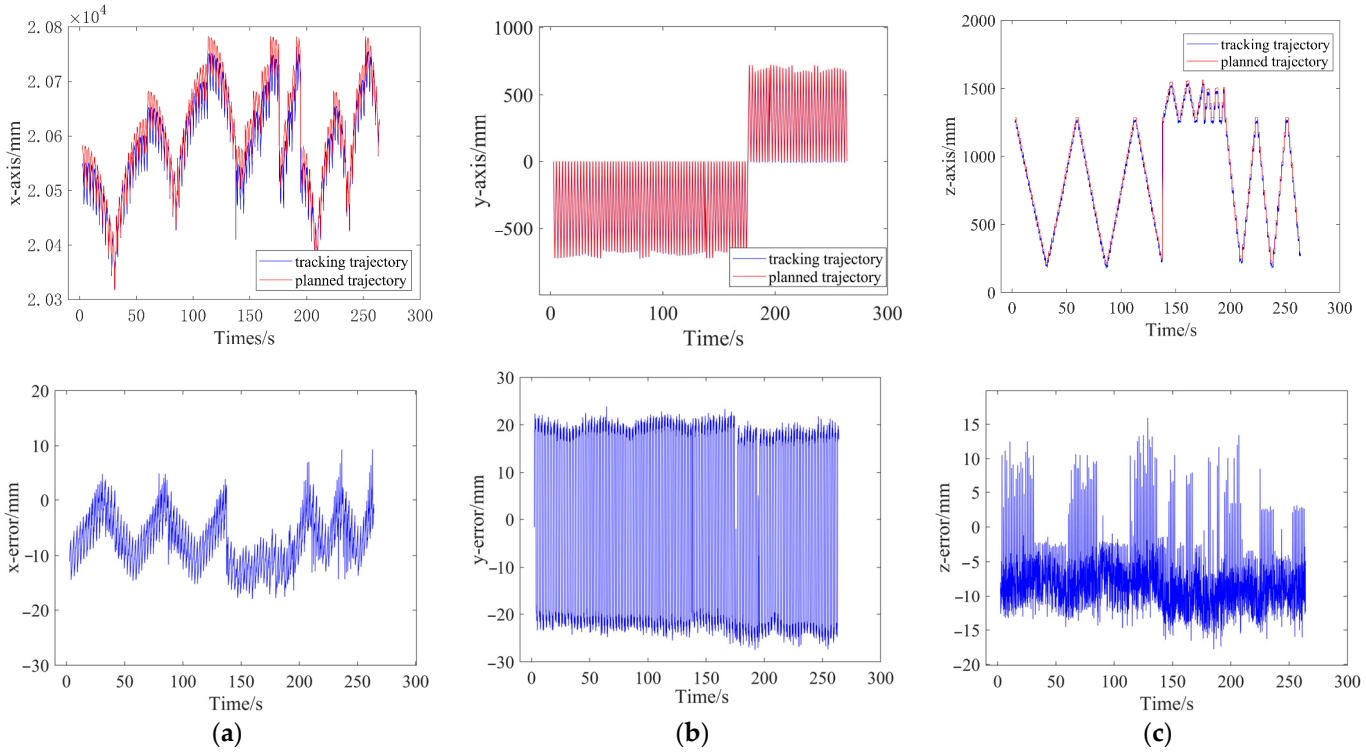

Figure 23. Comparative analysis and error of tracking paths in various directions. (**a**) Trajectory comparison and deviation in *x*-axis. (**b**) Trajectory comparison and deviation in *y*-axis. (**c**) Trajectory comparison and deviation in *z*-axis.

5. Conclusions

(1) A kinematic model of the boom-type roadheader was established to achieve coordinate transformation among the roadheader, target section, and roadway. By simplifying the cutting head model using segmented curves and establishing a section constraint model, a cutting space driven by the body pose was designed. Within this space, cutting trajectory points that meet the requirements for sectional forming cutting were obtained.

(2) A bi-objective optimization function was designed for the total length and total turning angle of the cutting trajectory, and an improved NSGA-II algorithm was used to solve for executable cutting trajectories. A simulation and experimental validation platform was built to verify the superiority of the improved algorithm. Additionally, a virtual simulation platform for trajectory tracking was used to verify the effectiveness of the cutting trajectories under different body poses, ensuring high forming quality.

(3) A trajectory-tracking experimental platform was built. The experimental results show that the cutting trajectory tracking is effective, with a maximum deviation of 23.879 mm, which can effectively support autonomous cutting control. This method not only provides effective cutting trajectories for autonomous cutting but also avoids frequent inverse kinematics solutions during the control process, improving computational efficiency and laying a foundation for autonomous control of sectional forming cutting.

(4) Future work will focus on enhancing the adaptability of the proposed method to more complex roadway conditions, including variable cross-sections and irregular geological environments. Additionally, the trajectory-planning approach will be further optimized to improve computational efficiency and real-time performance for practical applications.

Author Contributions: Conceptualization, C.Z. and X.Z.; Methodology, C.Z. and X.Z.; Software, W.Y. and J.W.; Validation, S.T. and Z.W.; Formal analysis, G.Z.; Investigation, C.Z.; Resources, C.Z.; Data curation, C.Z. and Y.D.; Writing—original draft, C.Z.; Writing—review & editing, X.Z.; Project administration, X.Z.; Funding acquisition, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 52104166), the Natural Science Foundation of Shaanxi Province (Grant No. 2021JLM-03), the China Postdoctoral Science Foundation funded project (No. 2022MD723826) and Key R & D project in Shaanxi (No. 2023-YBGY-063).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Deshmukh, S.; Raina, A.K.; Murthy, V.M.S.R.; Trivedi, R.; Vajre, R. Roadheader—A comprehensive review. *Tunn. Undergr. Space Technol.* **2020**, *95*, 103148. [CrossRef]
2. Zhang, X.; Yang, W.; Xue, X.; Zhang, C.; Wan, J.; Mao, Q.; Lei, M.; Du, Y.; Ma, H.; Zhao, Y.; et al. Challenges and development of the intelligent remote control on roadheaders in coal mine. *J. China Coal Soc.* **2022**, *47*, 579–597.
3. Qu, Y.; Ji, X.; Lv, F. Trajectory Planning for the Cantilevered Road-Header in Path Correction Under the Shaft. In *Advances in Intelligent Systems and Interactive Applications, Proceedings of the 4th International Conference on Intelligent, Interactive Systems and Applications (IISA2019), Bangkok, Thailand, 28–30 June 2019*; Springer: Cham, Switzerland, 2020; pp. 268–276.
4. Tian, J.; Wang, S.; Wu, M. Kinematic models and simulations for trajectory planning in the cutting of spatially-arbitrary cross-sections by a robotic roadheader. *Tunn. Undergr. Space Technol.* **2018**, *78*, 115–123. [CrossRef]
5. Wang, S.; Wu, M. Cutting trajectory planning of sections with complex composition for roadheader. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 1441–1452. [CrossRef]
6. Wu, J.; Xu, Z.; Fang, X.; Shi, G.; Wang, H. Research on path planning and control method for secondary autonomous cutting of cantilever roadheader in a large-section coal roadway. *Sustainability* **2023**, *15*, 560. [CrossRef]
7. Xu, Z.; Liang, M.; Fang, X.; Wu, G.; Chen, N.; Song, Y. Research on autonomous cutting method of cantilever roadheader. *Energies* **2022**, *15*, 6190. [CrossRef]
8. Qin, W.; Cheng, K.; Wang, B. Research on intelligent cutting control technology of transverse moving machine for large cross-section roadheader. *J. Vibro Eng.* **2024**, *26*, 24. [CrossRef]
9. Liu, X.; Du, C.; Liu, M. Research on spiral angle optimization for longitudinal road header's cutterhead. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2020**, *234*, 3346–3359. [CrossRef]
10. Wang, S.; Wu, M. Study on influencing factors and autonomous correction method of roadheader cutting trajectory. In Proceedings of the 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) 2017, Chengdu, China, 15–17 December 2017; pp. 654–658.
11. Qu, Y.; Yang, T.; Li, T.; Zhan, Y.; Fu, S. Path tracking of underground mining boom roadheader combining BP neural network and state estimation. *Appl. Sci.* **2022**, *12*, 5165. [CrossRef]
12. Wang, S.; Li, X.; Wu, M. Research on planning method of tunneling cutting trajectory with uncertain constraints. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC) 2018, Shenyang, China, 9–11 June 2018; pp. 2668–2672.
13. Wang, S.; Tian, J.; Wu, M. Study on cutting trace planning of longitudinal roadheader and boundary control method. *Coal Sci. Technol.* **2016**, *44*, 89–94+118.
14. Dong, Z.; Zhang, X.; Yang, W.; Lei, M.; Zhang, C.; Wan, J. Ant colony optimization-based method for energy-efficient cutting trajectory planning in axial robotic roadheader. *Appl. Soft Comput.* **2024**, *163*, 111965. [CrossRef]
15. Mao, J.; Chen, H.Y.; Xie, M.; Li, J.G.; Wang, J. Study on Trajectory Planning and Control Method in Profile Cutting of Roadheader. *Adv. Mater. Res.* **2010**, *139*, 1723–1727. [CrossRef]
16. Vargas, D.E.; Lemonge, A.C.; Barbosa, H.J. Solving multi-objective structural optimization problems using GDE3 and NSGA-II with reference points. *Eng. Struct.* **2021**, *239*, 112187. [CrossRef]
17. Kang, Y.; Cheng, L.; Zhang, Q.; Liu, X.; Ni, K. Data-driven RLV multi-objective reentry trajectory optimization based on new QABC algorithm. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 453–471. [CrossRef]

18. Sun, J.; Han, X.; Zuo, Y. Trajectory planning in joint space for a pointing mechanism based on a novel hybrid interpolation algorithm and NSGA-II algorithm. *IEEE Access* **2020**, *8*, 228628–228638. [CrossRef]
19. Gao, Y.; Cui, J.; Wang, N. Multi-Objective Drilling Trajectory Optimization Under Parameter Uncertainty. In Proceedings of the 2024 4th International Conference on Computer Science and Blockchain (CCSB) 2024, Shenzhen, China, 6–8 September 2024; pp. 11–14.
20. Chai, R.; Savvaris, A.; Tsourdos, A.; Xia, Y.; Chai, S. Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm. *IEEE Trans. Cybern.* **2018**, *50*, 1630–1643. [CrossRef] [PubMed]
21. Zhang, D.; Zhang, Z.; Li, Y.; Wang, Y.; Zhang, W.; Zhu, Y. An NSGA-II-based multi-objective trajectory planning method for autonomous driving. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD) 2024, Tianjin, China, 8–10 May 2024; pp. 2140–2146.
22. Machmudah, A.; Shanmugavel, M.; Parman, S.; Manan TS, A.; Dutykh, D.; Beddu, S.; Rajabi, A. Flight Trajectories Optimization of Fixed-Wing UAV by Bank-Turn Mechanism. *Drones* **2022**, *6*, 69. [CrossRef]
23. Qamar, N.; Akhtar, N.; Younas, I. Comparative analysis of evolutionary algorithms for multi-objective travelling salesman problem. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 371–379. [CrossRef]
24. Yang, W.; Zhang, X.; Ma, H.; Zhang, G. Laser beams-based localization methods for boom-type roadheader using underground camera non-uniform blur model. *IEEE Access* **2020**, *8*, 190327–190341. [CrossRef]