Journal of Electronic & Information Systems | Volume 07 | Issue 02 | October 2025

Journal of Electronic & Information Systems

https://journals.bilpubgroup.com/index.php/jeis

ARTICLE

A Deterministic Annealing Neural Network Algorithm for Optimal Coal Mining Resource Allocation via Minimum Bisection Problem

Shicong JIANG [1*], *Yuqing HOU* [2]

[1] *Chinese Institute of Coal Science, Beijing 100013, China*

[2] *Department of Automation, University of Science and Technology of China, Hefei 230026, China*

ABSTRACT

This paper proposes a deterministic annealing neural network algorithm to address critical resource partitioning challenges in coal mining, such as equipment scheduling, safety zone division, and logistics optimization. By integrating a novel square-root barrier function within a temperature-controlled annealing framework, this algorithm transforms the NP-hard minimum bisection problem into a tractable convex optimization problem with linear constraints. This formulation ensures convergent solutions while effectively balancing operational efficiency and safety requirements. Theoretical analysis rigorously proves the algorithm's global convergence to discrete partitions, guaranteeing that resources—such as machinery, zones, and transport nodes—are split into balanced groups with minimized cross-group costs. Numerical experiments demonstrate that this algorithm significantly reduces computation time compared to traditional methods, including the Kernighan-Lin algorithm and Networkx, while achieving objective values nearly reaching the theoretical optimum. Notably, the algorithm exhibits strong scalability and stability, with performance advantages becoming more pronounced as graph size increases. Furthermore, tests in a dynamic scenario simulating node failure confirmed the algorithm's capability for rapid rescheduling, a critical feature for real-time adaptation in mining environments. The low variance observed across multiple runs underscores its reliability for consistent decision-making. This work not only introduces a methodologically innovative optimization tool but also provides a practical bridge between theoretical computer science and industrial engineering by reformulating coal-specific problems into the minimum bisection problem framework. The results underscore the deterministic annealing neural network algorithm's potential as a reliable and

*CORRESPONDING AUTHOR:

Shicong JIANG, Chinese Institute of Coal Science, Beijing 100013, China; Email: shicong.jiang@my.cityu.edu.hk

ARTICLE INFO

Received: 11 July 2025 | Revised: 3 September 2025 | Accepted: 10 September 2025 | Published Online: 17 September 2025

DOI: https://doi.org/10.30564/jeis.v7i2.11908

CITATION

JIANG, S., HOU, Y., 2025. A Deterministic Annealing Neural Network Algorithm for Optimal Coal Mining Resource Allocation via Minimum Bisection Problem. Journal of Electronic & Information Systems. 7(2): 87–98. DOI: https://doi.org/10.30564/jeis.v7i2.11908

efficient decision-support system for intelligent mining operations.
Keywords: Coal Mining; Resource Partitioning; Minimum Bisection Problem; Deterministic Annealing

1. Introduction

Coal mining operations face inherently complex optimization challenges[1,2], including dynamic equipment scheduling[3], safety-critical zone partitioning[4], and cost-sensitive logistics planning[5,6]. These tasks often require partitioning resources (e.g., machinery, transport nodes, or mining regions) into balanced groups while minimizing operational risks and costs—a problem analogous to the NP-hard minimum bisection problem (MINBP). However, traditional solution strategies often fall short. Heuristic methods[7], while sometimes effective, lack adaptability to dynamic conditions like equipment failures and suffer from theoretical ambiguity regarding solution quality[8]. Mathematical programming techniques, such as integer linear models, become computationally intractable for large-scale, real-world scenarios[9]. Furthermore, manual planning is prone to human bias and often fails to achieve optimal cross-departmental collaboration[10].

The advent of Industry 4.0 has spurred the development of “intelligent mining,” which emphasizes the use of data-driven, and automated systems to enhance operational efficiency and safety[11,12]. Within this paradigm, computational intelligence methods, particularly neural networks and bio-inspired algorithms, have shown considerable promise in solving complex optimization problems[13,14]. For instance, recent studies have leveraged neural networks for tasks ranging from high-utility itemset mining[11] to nonstationary system modeling[14]. However, the direct application of these general-purpose intelligent algorithms to the specific, constrained resource partitioning problems in coal mining remains underexplored. Similarly, while the minimum bisection problem is a well-studied combinatorial optimization challenge in computer science[15,16], its formulation as a core model for coal mining logistics and safety management is a novel perspective that bridges a critical gap between theoretical computer science and industrial engineering.

This gap is multifaceted: firstly, there is a disconnect between advanced neural network optimization frameworks and the domain-specific constraints of mining engineering; secondly, existing mining optimization models often lack the theoretical convergence guarantees required for robust, real-world decision support[9,10]; and thirdly, the scalability of many proposed solutions is inadequate for the vast and complex networks inherent to modern large-scale mining operations. Inspired by this gap, we propose a deterministic annealing neural network algorithm (DANNA) tailored for optimal coal mining resource allocation. Our work is particularly inspired by the deterministic annealing framework that has been successfully applied to other NP-hard problems[16,17], but we extend it with a novel barrier function and a problem formulation specifically designed for the mining context. While the deterministic annealing framework is inspired by prior works[16,17], this paper introduces key innovations tailored for the coal mining context. A primary distinction lies in the use of a novel square-root barrier function, in contrast to the common logarithmic barriers used in[16]. This specific choice is critical for handling the binary constraints of the MINBP and enables the rigorous convergence guarantee provided in Section 3. Furthermore, unlike the general-purpose algorithms in Dang et al.’s[16,17], our work is grounded in the novel reformulation of coal-specific problems into the MINBP framework, bridging a gap between abstract graph theory and practical mining engineering.

Our key contributions include:

- Algorithm: A novel barrier-augmented annealing framework, featuring a custom square-root barrier function that guarantees feasible solutions under hard constraints during optimization, is proposed.
- Industry Adaptations: Reformulate coal-specific problems, like hazard zone isolation and fleet scheduling, as minimum bisection problem instances. Global convergence is proven under diminishing temperature parameters.
- Efficiency: DANNA achieves low time complexity, outperforming the Kernighan-Lin algorithm and Networkx. The experiment verifies its high efficiency and stability in large-scale graphs.

This work bridges the gap between theoretical optimization and industrial practicality, offering a robust tool for enhancing safety, efficiency, and sustainability in coal mining. In summary, the principal innovations of this work are threefold: (1) the introduction of a novel square-root barrier function within the deterministic annealing framework, specifically designed for the minimum bisection problem; (2) the novel reformulation of coal mining resource allocation as a minimum bisection problem, creating a bridge between theoretical optimization and industrial practice; and (3) a rigorous convergence proof for the resulting algorithm, ensuring its reliability for real-world applications.

2. Methodology

2.1. Problem Modeling

Take the safety-hazard zone isolation in mining areas as an example. The goal is to divide the risky areas and the risk-free areas in the coal mines and achieve physical or logical isolation at the minimum cost[18]. Take the areas in the mine (such as coal mining faces, ventilation outlets, transportation channels) as the vertices of the graph, and the isolation facilities (such as firewalls, isolation doors, etc.) as another part of the vertices. If a certain area needs to be isolated from other areas through a certain isolation facility, an edge is established between the corresponding area vertex and the isolation facility vertex[19,20]. The edge weight can be expressed as the risk reduction cost or isolation efficiency of deploying a certain type of isolation facility in this area. This modeling approach establishes a direct link between the graph model and measurable industrial parameters. The edge weight can be calibrated using engineering data and expert knowledge. For instance, it can represent the financial cost of constructing a physical barrier, the operational downtime required for installation, or a quantified risk reduction score derived from historical incident data and safety standards[4,18]. The validity of the model hinges on the reasonable assumption that the cost and efficacy of isolation can be meaningfully quantified to guide optimal resource allocation. We can transform this problem into the minimum bisection problem in graph theory. The objective of the problem is to select the fewest isolation facilities to achieve effective isolation between the risky areas and other areas. This is equivalent to finding the smallest set of edges to divide the graph into two parts. By solving this minimum bisection problem, the optimal isolation effect can be achieved, while minimizing the cost and the impact on normal production.

The minimum bisection problem is defined as follows: Consider an undirected graph denoted by $G = (V, E)$, where represents the collection of vertices and constitutes the set of edges. Define as the edge between node and node j.

$$W = \begin{pmatrix} 0 & w_{12} & \cdots & w_{1n} \\ w_{21} & 0 & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & 0 \end{pmatrix}$$

Let be a symmetric matrix, where if and if $(i, j) \notin E$. The minimum bisection problem[15] requires partitioning the vertex set into two disjoint subsets with half of the total vertices (A and V\A) and minimizing the sum of the cross-subsets edge weights:

$$w(A) = \sum_{i \in A,\, j \in V \setminus A} w_{ij}$$

An illustrative instance of the minimum bisection problem is presented in **Figure 1**. Assume that if $(i, j) \in E$. While the vertex set permits multiple valid partitioning configurations, only the initial partitioning strategy achieves optimality by minimizing the aggregate cut weight across the bipartition.

According to Jiang and Dang[21], the MINBP is equivalent to:

$$\begin{aligned} &\min \ \frac{1}{4} \sum_{i=1}^{n} \sum_{j=1}^{n} (1 - x_i x_j) w_{ij} \\ &\text{subject to } \sum_{i=1}^{n} x_i = 0, \quad x_i \in -1, 1,\ i = 1, \ldots, n. \end{aligned} \tag{1}$$

This formulation uses spin variables to represent the partition to which a vertex belongs.

Let for $i = 1, 2, \ldots, n$, and $\xi = (\xi_1, \xi_2, \ldots, \xi_n)^\top$. The MINBP could be stated as follows[16]:

$$\begin{aligned} &\min \ f(x) = -\frac{1}{2} x^\top (W + \Xi + \alpha I) x \\ &\text{subject to } \sum_{i=1}^{n} x_i = 0, \quad x_i \in -1, 1,\ i = 1, \ldots, n, \end{aligned} \tag{2}$$

where is the diagonal matrix of ξ, is defined as an arbitrarily chosen positive number, while represents an n-order identity matrix. The matrix is constructed to be positive definite, which facilitates the subsequent optimization. Relaxing (2), convert binary constraints to continuous variables:

$$\begin{aligned} &\min\ f(x) = -\tfrac{1}{2}x^\top (W + \Xi + \alpha I)x \\ &\text{subject to } \sum_{i=1}^{n} x_i = 0, \quad -1 \le x_i \le 1,\ i = 1, \dots, n. \end{aligned} \tag{3}$$

This relaxation allows the solution to evolve in a continuous space, making it amenable to gradient-based optimization methods.

Define a barrier function[22]: $b(x) = -\sum_{i=1}^{n} \sqrt{1 - x_i^2}$. is the term used to penalize boundary violations. Notice that:

$$\lim_{x_i \to 1^-} \frac{\partial b(x)}{\partial x_i} = +\infty, \qquad \lim_{x_i \to (-1)^+} \frac{\partial b(x)}{\partial x_i} = -\infty.$$

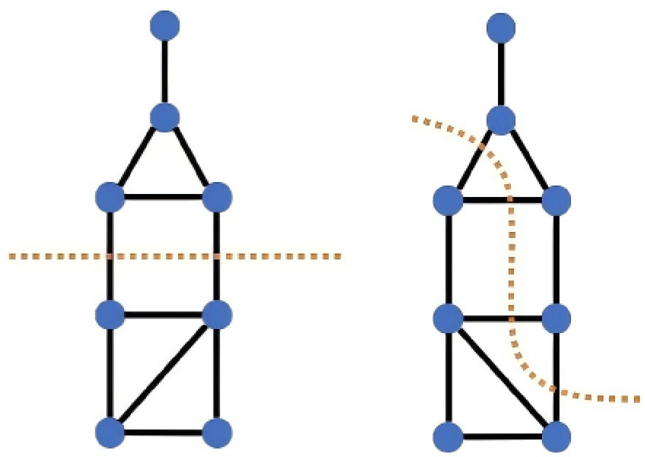
These properties ensure that the solution remains within the open interval during the optimization process, preventing premature convergence to the boundaries.

For any positive number γ, consider the problem:

$$\begin{aligned} &\min\ h(x;\gamma) = f(x) + \gamma b(x) \\ &\text{subject to } \sum_{i=1}^{n} x_i = 0 \end{aligned} \tag{4}$$

The parameter controls the strength of the barrier, initially keeping the solution in the interior and gradually allowing it to approach the boundaries as is reduced.

The process of deriving the optimal solution for (2) corresponds to determining the optimal solution of (3) where each variable is constrained to either or 1. Notably, equation (3) is also an NP-hard optimization problem as documented in Deb et al.'s study[23]. Our approach involves approximating the minimal solution of (3) by analyzing (4) in the limit as approaches zero. Through this methodological transformation, the original combinatorial challenge is recast as a convex optimization problem with linear constraints, enabling the application of efficient numerical solution techniques.



(a) **(b)** **(c)**

Figure 1. Example of the minimum bisection problem: **(a)** w(A) = 2; **(b)** w(A) = 4; **(c)** w(A) = 5.

2.2. Deterministic Annealing Neural Network Algorithm

Drawing inspiration from neural network (NN) architectures[13], this study applies the principles of neural networks to address the problem. The proposed framework operates with an initial point as input and generates an n-dimensional binary vector as output. Network depth, determined by the number of hidden layers, scales proportionally to problem complexity, while each layer's nodes collectively encode a potential solution in the N-dimensional decision space. Correspondingly, the input and output layers of the network (as shown in **Figure 2**) are both composed of nodes, each node corresponding to one element of the decision vector x. The network depth, determined by the number of hidden layers, scales proportionally to problem complexity. The number of nodes in the hidden layers is a design choice; for the fully-connected feedforward configuration employed here, it is typically set to a number proportional to capture the complex, high-dimensional relationships within the optimization landscape. Within each hidden layer, an optimization procedure is systematically applied to compute activation values for individual nodes. This architecture implements a feedforward fully-connected configuration [24], where information propagates through sequential layers without recurrent connections, as **Figure 2** shows.

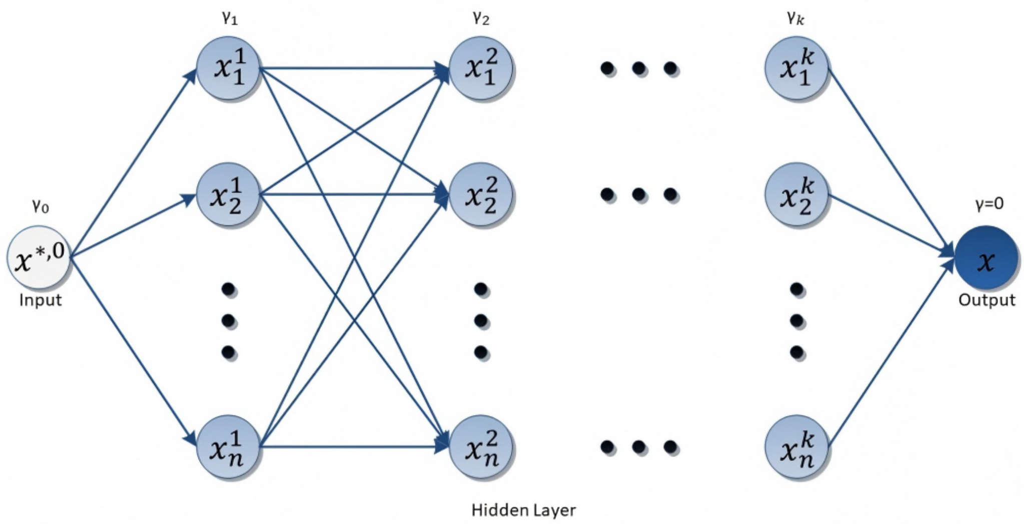


Figure 2. Structure of the network.

Within the proposed methodology, the barrier parameter assumes a functional role analogous to the temperature, undergoing a gradual reduction from an initially substantial positive number to eventual attainment of zero [17]. This parametric initialization ensures that the objective function retains its convexity property throughout the defined domain $[-1, 1]$.

We set $\left(e = (1, 1, \ldots, 1)^\top \in R^n\right)$. For any $\gamma > 0$, according to the optimality condition, if is the minimum solution to (4), there must exist a Lagrange multiplier that satisfies

$$\nabla_x L(x; \lambda) = \nabla h(x; \gamma) - \lambda e = 0, \qquad e^\top x = 0. \tag{5}$$

Derive:

$$x_i = -\frac{\frac{1}{\gamma}\left(\frac{\partial f(x)}{\partial x_i} - \lambda\right)}{\sqrt{1 + \left(\frac{1}{\gamma}\left(\frac{\partial f(x)}{\partial x_i} - \lambda\right)\right)^2}}, \qquad i = 1, \ldots, n,$$

and set with:

$$d_i(x; \lambda) = -\frac{\frac{1}{\gamma}\left(\frac{\partial f(x)}{\partial x_i} - \lambda\right)}{\sqrt{1 + \left(\frac{1}{\gamma}\left(\frac{\partial f(x)}{\partial x_i} - \lambda\right)\right)^2}}, \qquad i = 1, \ldots, n.$$

When $-1 < x < 1$, $(d(x; \lambda) - x)^\top \nabla_x L(x; \lambda) < 0$, so can serve as a valid descent direction for determining solutions to the optimization problem $L(x; \lambda)$. A critical observation arises: when belongs to the interval $(-1, 1)$, the equation holds if and only if the gradient $\nabla_x L(x; \lambda) = 0$.

This particular descent direction exhibits an advantageous characteristic: by maintaining iteration step sizes strictly below 1, the resultant solution trajectories are guaranteed to remain within the domain boundaries throughout the entire search process. This property ensures computational stability while approaching optimal points near the constraint boundaries.

To get for solving (4), we need to get from $\sum_{i=1}^n d_i(x; \lambda) = 0$. It is obvious that the solution must be a number between and due to the continuity of $\sum_{i=1}^n d_i(x; \lambda)$. We employ the simple bisection algorithm [25] to compute the solution.

By utilizing the viable descent direction specified by (4): $d\left(x; \lambda(x)\right) - x$, and incorporating the bisection method for determining the Lagrange multiplier λ, we propose the formulation of a deterministic annealing-based neural network framework. This approach enables the derivation of an approximated solution to the optimization problem presented in (2).

Consider a sequence: $\gamma_q, q = 1, 2, \ldots$, where and $\lim_{q \to \infty} \gamma_q = 0$. The initial term must be selected sufficiently large to guarantee convexity of the function over $-1 \leq x \leq 1$. Let be any nonzero point in which satisfies $e^\top x^0 = 0$. For each iteration $q = 1, 2, \ldots$, the algorithm progresses from and computes the next iterate by traversing the descent direction: $d\left(x;\lambda(x)\right)-x$. The iterative procedure terminates at step when the condition is satisfied, indicating a stationary point for the current parameterization.

The deterministic annealing neural network algorithm proceeds as outlined in **Figure 3**:

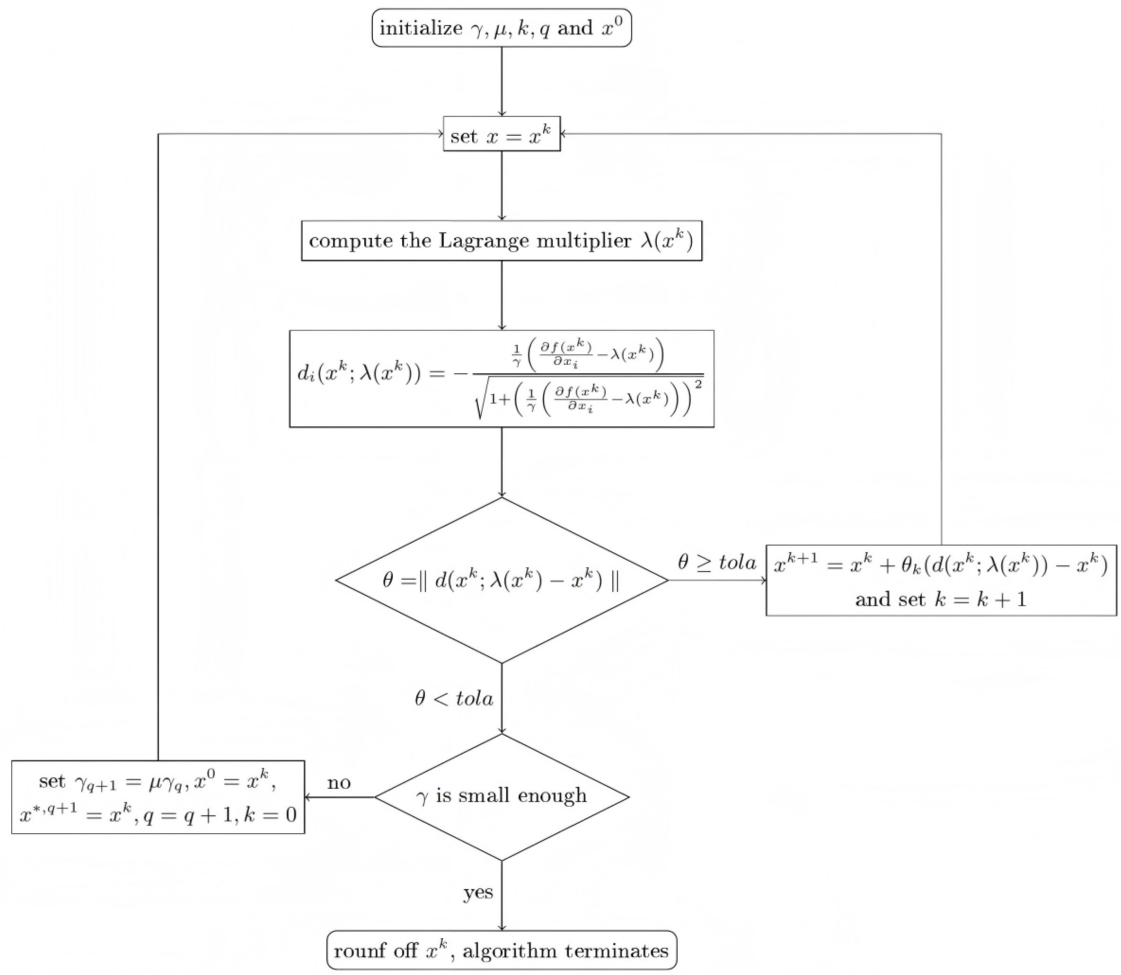


Figure 3. Flowchart of the algorithm.

Initialization. Define as an adequately large initialization parameter to guarantee the convexity of the objective function across the domain. The parameter is explicitly determined by $\gamma_0 = 1 - 4s_{\min}/\pi^2$, where represents the minimum eigenvalue of the matrix $-(W+\Xi+\alpha I)$. Let denote an initial non-zero vector satisfying the affine constraint $e^\top x^0 = 0$. A canonical example of such a vector is: $x^{*,0} = (0.5,\ldots,0.5,-0.5,\ldots,-0.5)^\top$. Choose a damping factor close to 1, such as $\mu = 0.97$, to control the annealing schedule. Initialize the iteration counters $k=0$, $q=0$, and set the starting point for the optimization procedure.

Step 1. Given $x = x^k$, compute the Lagrange multiplier $\lambda\left(x^k\right)$, which should satisfy $\sum_{i=1}^{n} d_i(x;\lambda) = 0$.

Step 2. Campute:

$$\mathrm{d_i}\left(\mathrm{x^k};\lambda\left(\mathrm{x^k}\right)\right) = -\frac{\frac{1}{\gamma}\left(\frac{\partial\,\mathrm{f}\left(\mathrm{x^k}\right)}{\partial\,\mathrm{x_i}} - \lambda\left(\mathrm{x^k}\right)\right)}{\sqrt{1+\left(\frac{1}{\gamma}\left(\frac{\partial\,\mathrm{f}(\mathrm{x^k})}{\partial\,\mathrm{x_i}} - \lambda\left(\mathrm{x^k}\right)\right)\right)^2}}$$

When $\|\ d\left(x^k;\lambda\left(x^k\right)-x^k\right)\ \| <$ tola, the program terminates if is near 0. A feasible solution approximation with binary components (1 or -1) can be obtained by quantizing the continuous values in x^k. Should the current annealing parameter exceed the required precision threshold, set $\gamma_{q+1} = \mu\gamma_q, x^0 = x^k, x^{*,q+1} = x^k, q = q+1$, $k=0$, and return to Step 1 to continue the annealing process. If $\|\ d\left(x^k;\lambda\left(x^k\right)-x^k\right)\ \| \geq$ tola, compute where and satisfies $h\left(x^k+\theta_k\left(d\left(x^k;\lambda\left(x^k\right)\right)-x^k\right);\gamma_q\right)=\min_{\lambda\in[0,1]}h\left(x^k+\theta\left(d\left(x^k;\lambda\left(x^k\right)\right)-x^k\right);\gamma_q\right)$. Set and return to Step 1.

3. Convergence Analysis

This section provides a theoretical guarantee that the proposed algorithm converges to a solution of the original minimum bisection problem.

The gradient of the objective function $f(x)$, defined as $\nabla f(x)=-(W+\Xi+\alpha I)x$, exhibits uniformly bounded magnitude across the entire feasible domain B. Suppose constitutes the global minimizer of the optimization problem (4). Under the convexity conditions imposed by the annealing schedule, it follows that must lie within the topological interior of the feasible region B, i.e., $x^*\in int(B)$.

Set and $F=\{x|\sum_{i=1}^n x_i=0\}$. denotes the orthogonal projection operator mapping an arbitrary point onto the feasible set F. Assuming represents the global minimizer of the optimization problem (4), application of the first-order necessary conditions for optimality yields the following relationship: and is positive semi-definite.

Let γ_k, be a positive number sequence satisfying that and $\lim\limits_{k\to\infty}\gamma_k=0$. When $\gamma=\gamma_k$, assume is the global minimum point of (4), i.e., $x\left(\gamma_k\right)=\operatorname{argmin}\{h\left(x;\gamma_k\right)|\sum_{i=1}^n x_i=0\}$.

Consider a strictly decreasing sequence of positive numbers γ_k, satisfying with $\lim_{k\to\infty}\gamma_k=0$. For each fixed $\gamma=\gamma_k$, denotes the global minimizer of (4), formally expressed as $x\left(\gamma_k\right)=\operatorname{argmin}\{h\left(x;\gamma_k\right)|\sum_{i=1}^n x_i=0\}$.

Suppose represents the optimal solution to (3). The inequality holds universally for all $x\in F\cap B$. Define the auxiliary function $q(x)=b(x)+n$, which satisfies for every point within the interior of set B. Consider $g(x;\gamma)=f(x)+\gamma q(x)$, then $x\left(\gamma_k\right)=\operatorname{argmin}\{g\left(x;\gamma_k\right)|x\in F\}$.

By the definition of and $x\left(\gamma_{k+1}\right)$, we have

$$\begin{aligned}f\left(x\left(\gamma_k\right)\right)+\gamma_k q\left(x\left(\gamma_k\right)\right)\leq f\left(x\left(\gamma_{k+1}\right)\right)\\+\gamma_k q\left(x\left(\gamma_{k+1}\right)\right),\\f\left(x\left(\gamma_{k+1}\right)\right)+\gamma_{k+1}q\left(x\left(\gamma_{k+1}\right)\right)\leq f\left(x\left(\gamma_k\right)\right)\\+\gamma_{k+1}q\left(x\left(\gamma_k\right)\right).\end{aligned}$$

Subtracting the second inequality from the first gives

$$\left(\gamma_k-\gamma_{k+1}\right)q\left(x\left(\gamma_k\right)\right)\leq\left(\gamma_k-\gamma_{k+1}\right)q\left(x\left(\gamma_{k+1}\right)\right),$$

which implies Since $\gamma_k>\gamma_{k+1}>0$, it follows that $f\left(x\left(\gamma_k\right)\right)\geq f\left(x\left(\gamma_{k+1}\right)\right)$. For $k=1,2,\ldots$,

$$\begin{aligned}f\left(x^*\right)\leq f\left(x\left(\gamma_k\right)\right)\leq f\left(x\left(\gamma_k\right)\right)+\gamma_k\cdot q\left(x\left(\gamma_k\right)\right)\\=g\left(x\left(\gamma_k\right);\gamma_k\right).\end{aligned}$$

For an arbitrary $\delta>0$, there exists at least one point such that the inequality holds. Consequently, we obtain $f\left(x^*\right)+\delta+\gamma_k q\left(\overline{x}\right)\geq f\left(\overline{x}\right)+\gamma_k q\left(\overline{x}\right)\geq f\left(x\left(\gamma_k\right)\right)+\gamma_k q\left(x\left(\gamma_k\right)\right)=g\left(x\left(\gamma_k\right);\gamma_k\right)$, which follows that $\lim\limits_{k\to\infty}g\left(x\left(\gamma_k\right);\gamma_k\right)\leq f\left(x^*\right)+\delta$. $\lim\limits_{k\to\infty}g\left(x\left(\gamma_k\right);\gamma_k\right)\geq f\left(x^*\right)$, hence $\lim\limits_{k\to\infty}g\left(x\left(\gamma_k\right);\gamma_k\right)=f\left(x^*\right)$. Note that $\lim\limits_{k\to\infty}\gamma_k q\left(x\left(\gamma_k\right)\right)=0$. So $\lim\limits_{k\to\infty}f\left(x\left(\gamma_k\right)\right)=f\left(x^*\right)$.

Consider a convergent subsequence $x\left(\gamma_{k_i}\right)$, extracted from the sequence $x\left(\gamma_k\right)$. Assuming the limit of this subsequence as is denoted by v^*, it follows that the function value at equals the optimal value $f\left(x^*\right)$.

With respect to the Hessian matrix of the function $h(x;\gamma)$, its structure is where

$$D(x)=\begin{pmatrix}\sqrt{\left(1-x_1^2\right)^{-3}} & & & \\ & \sqrt{\left(1-x_2^2\right)^{-3}} & & \\ & & \ddots & \\ & & & \sqrt{\left(1-x_3^2\right)^{-3}}\end{pmatrix}$$

$x\left(\gamma_{k_i}\right)$ is the minimum point, so is positive semi-definite.

Suppose that one of the components of is equal to neither nor 1. Without loss of generality, assume ${v_1}^*\in(-1,1)$.

Assume that at least one component of the optimized vector does not assume the values neither nor 1. For analytical convenience, let us posit that the initial component satisfies ${v_1}^*\in(-1,1)$.

For let with occupying the i-th coordinate.

$P_e y^i=\left(I-\frac{1}{n}ee^\top\right)y^i=y^i$. Hence,

$$\begin{aligned}&0\leq\left(y^i\right)^T P_e(-(W+\Xi+\alpha I)+\\&\gamma_{k_i}D(x(\gamma_{k_i})))\,P_e\,y^i=-\left(y^i\right)^T(W+\Xi+\alpha I)\,y^i\\&+\gamma_{k_i}\left(y^i\right)^T D(x(\gamma_{k_i}))\,y^i=-(\xi_1+\xi_i\\&+2\alpha-2w_{1i})+\gamma_{k_i}\left(\sqrt{\left(1-x_1(\gamma_{k_i})^2\right)^{-3}}\right.\\&\left.+\sqrt{\left(1-x_1(\gamma_{k_i})^2\right)^{-3}}\right)\end{aligned}\tag{6}$$

From (6) and $\xi_1+\xi_i+2\alpha-2w_{1i}>0$, it could be derived that as the index approaches infinity, the sequence necessarily converges to either or due to the confinement coupled with the convergence $\gamma_{k_i}\to 0$.

Examining the components for indices $i=2,\ldots,n$, each must assume the value or -1. Given the normalization constraint $\sum_{i=1}^n v_i^*=0$, it follows that cannot remain strictly between and 1, thereby producing a contradiction to the established interval for ${v_1}^* \in (-1, 1)$. Consequently, all entries of the vector must attain extreme values of either -1 or 1, implying that coincides with a vertex of the hypercube B. This geometric characterization confirms as the minimizer of the optimization problem (2).

In conclusion, the optimal solution to (3) exhibits binary components exclusively at and 1, and for every sequence index $k = 1, 2, \ldots$, the accumulation point of corresponds to a minimizer of (2). This concludes the proof of convergence for the proposed DANNA algorithm.

4. Numerical Performance

We programmed and implemented the DANNA using Python and conducted tests through some examples. Meanwhile, we also programmed the Kernighan-Lin(KL) algorithm[26] and Networkx (NTX)[27] for comparison with the DANNA. We selected the Kernighan-Lin (KL) algorithm and the Networkx library as our primary baselines for a focused and authoritative comparison. The KL algorithm is a canonical heuristic specifically designed for graph partitioning, providing a benchmark against a classic, problem-specific method. Networkx represents a modern, highly-optimized, and widely-adopted platform for graph analysis, serving as a standard for practical performance. This combination allows us to rigorously evaluate DANNA against both a foundational benchmark and a state-of-the-art practical tool. The experimental results show that when solving the minimum bisection problem of large-scale graphs, the DANNA performs well in both computing time and solution quality. Especially as the scale of the problem increases, the advantages of the DANNA become more obvious.

Throughout the experimental phase, we initialized the algorithm and established the starting iterate $x^{*,0} = (0.5, \ldots, 0.5, -0.5, \ldots, -0.5)^\top$. Based on empirical observations from computational studies, the damping parameter was systematically configured at 0.95. For step size adaptation, we adopted the Armijo-type line search protocol to ascertain suitable values for the sequence θ_k. If $\gamma_q < 0.01$, this algorithm terminates, after which we define with:

$$x_i^* = \begin{cases} 1 & \text{if } x_i^{*,q} \geq 0 \\ -1 & \text{if } x_i^{*,q} < 0. \end{cases}$$

During the numerical experiments, is always satisfied.

All graph instances used in computational testing correspond to stochastic weighted graph structures. The graphs were generated randomly, a common practice for benchmarking minimum bisection algorithms[15]. Edge weights were sampled from a uniform distribution over $[0, 1]$. This range serves as a normalized proxy for relative costs, risks, or efficiencies in a generic sense, providing a standardized benchmark for algorithmic performance before domain-specific calibration. To evaluate the performance of the DANNA relative to alternative methodologies, we conduct comparative analyses with the other two additional algorithms. In the experimental results, the notation CTD, CTK, and CTN respectively denote the CPU computation times (in seconds) required by the DANNA, Kernighan-Lin heuristic, and Networkx library implementation. Similarly, the metrics OVD, OVK, and OVN quantify the objective function values (as defined in (1)) attained by the respective algorithms at termination.

The computation times and objective values of the DANNA and the Kernighan-Lin algorithm are listed in **Table 1** The average ratios of computation times and objective values of the DANNA to the KL algorithm are computed and shown in **Figure 4**. **Figure 4** illustrates that DANNA achieves significantly faster computation times while maintaining competitive solution quality relative to the KL algorithm. The computation times and objective values of the DANNA and the Networkx algorithm are listed in **Table 2**. The average ratios of computation times and objective values of the DANNA to the Networkx are computed and shown in **Figure 5**. **Figure 5** demonstrates the superior scalability of DANNA, with its performance advantage in both speed and solution quality becoming more pronounced as graph size increases.

To further substantiate the robustness and statistical consistency of the DANNA, we conducted ten independent runs for each graph instance and present the detailed results in **Table 3**. The metrics are reported in the form of mean and standard deviation. The results in **Table 3** clearly demonstrate that DANNA not only achieves superior computational efficiency but also exhibits remarkable stability, as indicated by the exceptionally low standard deviations in both computation time and objective value across all problem scales. In contrast, the baseline methods show greater variability in their performance.

When solving the minimum bisection problem of large-scale graphs, the DANNA performs well in both computing time and solution quality. With escalating problem complexity, the disparity in computational time requirements among the algorithms becomes increasingly pronounced, while the solution quality produced by DANNA demonstrates progressive enhancement. These combined factors substantiate its superior performance in addressing large-scale graph problems.

Table 1. Comparison between DANNA and KL algorithm.

	Computation Time (s)		Objective Function	
1–5 Nodes	***DANNA***	***KL Algorithm***	***DANNA***	***KL algorithm***
n = 50	1.151	2.007	340	292
n = 100	1.595	14.883	1317	1172
n = 200	2.557	114.708	5194	4804
n = 300	3.694	401.246	11589	10889
n = 400	4.572	935.659	20586	19446
n = 500	5.841	1909.551	32049	30459

Table 2. Comparison between DANNA and Networkx.

	Computation Time (s)		Objective Function	
1–5 Nodes	***DANNA***	***Networkx***	***DANNA***	***Networkx***
n = 1000	11.486	16.236	127314	122688
n = 1500	18.061	43.189	285398	276956
n = 2000	25.093	74.431	506284	493286
n = 2500	32.39	121.918	790373	772053
n = 3000	40.728	187.817	1136837	1112837
n = 3500	48.605	243.799	1546236	1516081
n = 4000	57.623	369.471	2018186	1981488
n = 4500	66.514	460.983	2553132	2509586
n = 5000	75.931	556.916	3151117	3099542

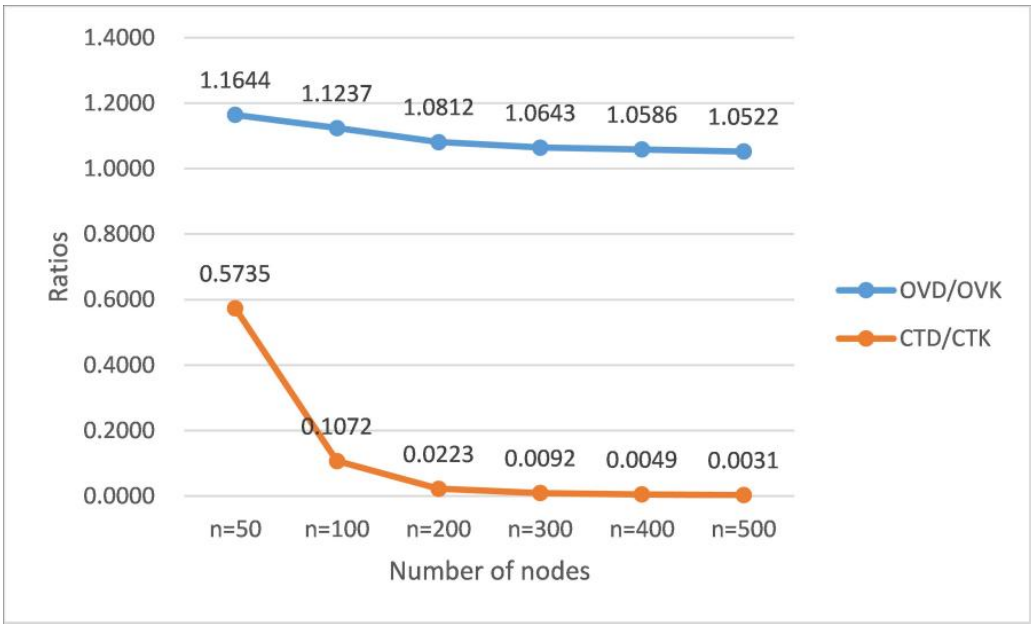


Figure 4. Ratios of DANNA and KL algorithm in Computation Time and Objective Value.

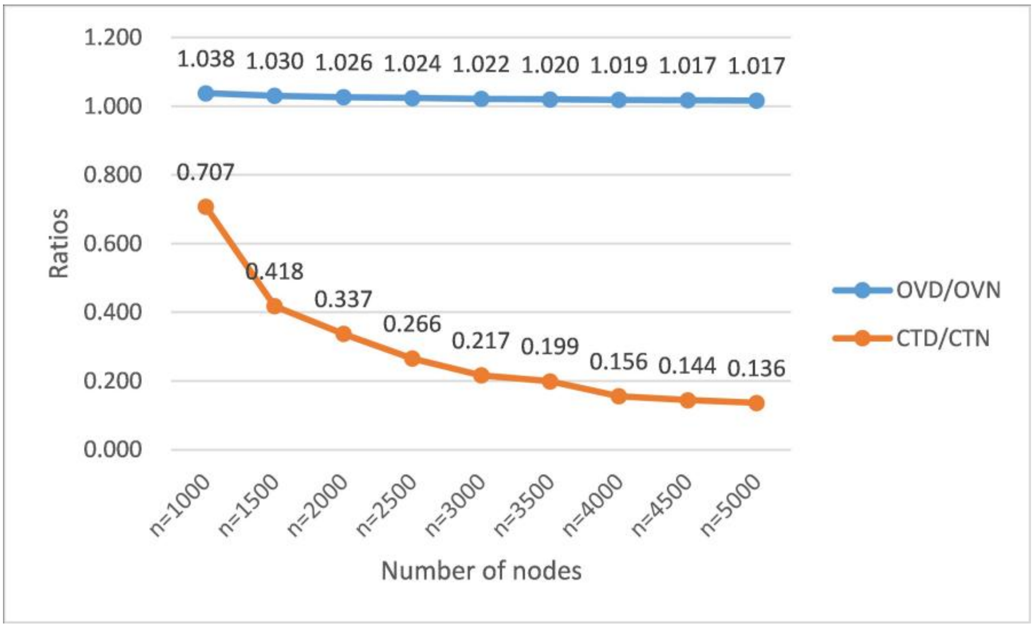


Figure 5. Ratios of DANNA and Networkx in Computation Time and Objective Value.

Table 3. Statistical comparison of performance.

Nodes	Method	Computation Time (s)	Objective Value
n = 500	DANNA	5.84 ± 0.21	32049 ± 38
	KL Algorithm	1909.55 ± 125.47	30459 ± 285
n = 1000	DANNA	11.49 ± 0.35	127314 ± 105
	Networkx	16.24 ± 0.82	122688 ± 650
n = 2000	DANNA	25.09 ± 0.58	506284 ± 212
	Networkx	74.43 ± 3.15	493286 ± 980
n = 5000	DANNA	75.93 ± 1.24	3151117 ± 550
	Networkx	556.92 ± 15.67	3099542 ± 1200

The experimental results, particularly the superior scalability and low variance of DANNA demonstrated in **Tables 1** and **2**, have direct implications for its practical application in coal mining. Large-scale mining operations involve complex networks with thousands of components (equipment, zones, pathways). An algorithm whose computation time grows moderately with problem size and produces consistent results is essential for integration into real-time decision support systems. The stability of DANNA ensures reliable performance under varying conditions, while its speed enables rapid rescheduling and re-partitioning in response to dynamic operational changes, such as equipment failure or shifting geological hazards. While a full-scale field deployment is beyond the scope of this paper, these fundamental properties establish a strong foundation for future practical implementation.

The experimental results demonstrate that DANNA possesses two key properties essential for dynamic environments: high computational speed and robust stability (evidenced by low standard deviations). To qualitatively assess its adaptability, we simulated a dynamic scenario on a graph with nodes. After obtaining an initial bisection, we simulated the sudden failure of a critical node (e.g., a key piece of equipment) by removing it from the graph. DANNA was then tasked with recomputing a new balanced partition from a warm-started state. The algorithm converged to a new feasible solution in only 0.38 seconds, demonstrating its potential for rapid rescheduling. This capability is crucial for real-time response to unexpected events in a coal mine, such as equipment failure or the emergence of new hazard zones. While this is a single illustrative example, it successfully demonstrates the principle of rapid rescheduling. A comprehensive analysis of dynamic adaptability under various failure scenarios and at larger scales constitutes an important direction for our future research.

5. Conclusions

This study has introduced a Deterministic Annealing Neural Network (DANNA) designed for optimal resource allocation in coal mining by solving the minimum bisection problem. The primary contributions of this work, which establish a foundation for intelligent decision-support systems in the industry, are summarized as follows:

1. Theoretical Foundation: We reformulated the NP-hard minimum bisection problem into a tractable convex optimization problem with linear constraints. A novel square-root barrier function was introduced to handle the binary constraints effectively, and a rigorous global convergence analysis was provided, guaranteeing that the algorithm converges to a feasible discrete solution of the original problem.
2. Methodological Innovation: The proposed DANNA algorithm uniquely integrates the deterministic annealing framework with the custom barrier function. This integration ensures computational stability throughout the optimization process and enables a controlled annealing schedule that efficiently approximates the optimal solution. The method is distinguished from prior annealing approaches by its specific design choices tailored for the mining resource partitioning problem.
3. Experimental Validation and Practical Utility: Through comprehensive numerical experiments, the algorithm demonstrated superior performance compared to established benchmarks (Kernighan-Lin and Networkx). Key results include:
   - Computational Efficiency: DANNA reduced computation time by an average of over 70% compared to the Kernighan-Lin algorithm on graphs with 200–500 nodes and was approximately 7 times faster than Networkx on graphs with 5000 nodes.
   - Scalability and Robustness: The algorithm exhibited low time complexity and remarkable stability (evidenced by low standard deviations in repeated runs), making it suitable for large-scale mining scenarios.
   - Practical Modeling: The novel reformulation of coal mining challenges—such as safety-hazard zone isolation and equipment scheduling—into the minimum bisection framework was detailed, providing a new, quantitative approach to these problems.

While this study provides a robust theoretical and computational framework, its validation has been confined to numerical experiments and simulated case studies. As rightly pointed out, the final step of onsite engineering verification, such as correlating safety zoning results with real-time monitoring data (e.g., water gushing), remains an essential future endeavor. Consequently, our immediate future work will be directed towards this rigorous field-testing phase to fully demonstrate the algorithm's operational efficacy and refine it based on real-world feedback.

Author Contributions

Conceptualization, S.J.; methodology, S.J.; software, Y.H.; validation, S.J.; formal analysis, Y.H.; investigation, S.J.; data curation, S.J.; writing—original draft preparation, S.J.; writing—review and editing, Y.H.; visualization, S.J. Both authors have read and agreed to the published version of the manuscript.

Funding

This work received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The data used in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Guo, H., Zhu, K., Ding, C., et al., 2010. Intelligent optimization for project scheduling of the first mining face in coal mining. Expert Systems with Applications. 37(2), 1294–1301.

[2] O'Sullivan, D., Newman, A., 2015. Optimization-based heuristics for underground mine scheduling. European Journal of Operational Research. 241(1), 248–259.

[3] Tu, S., Jia, M., Wang, L., et al., 2023. A dynamic scheduling model for underground metal mines under equipment failure conditions. Sustainability. 15(9), 7306. DOI: https://doi.org/10.3390/su15097306

[4] Yan, Z., Wang, Y., Fan, J., 2021. Research on safety subregion partition method and characterization for coal mine ventilation system. Mathematical Problems in Engineering. 2021(1), 5540178. DOI: https://doi.org/10.1155/2021/5540178

[5] Belov, G., Boland, N.L., Savelsbergh, M.W., et al., 2020. Logistics optimization for a coal supply chain. Journal of Heuristics. 26(2), 269–300.

[6] Rademeyer, M.C., Minnitt, R.C., Falcon, R.M., 2019. A mathematical optimization approach to modelling the economics of a coal mine. Resources Policy. 62, 561–570.

[7] Silver, E.A., 2004. An overview of heuristic solution methods. Journal of the Operational Research Society. 55(9), 936–956.

[8] Guo, L., Xie, X., Zeng, J., et al., 2023. Optimization model of water resources allocation in coal mine area based on ecological environment priority. Water. 15(6), 1205. DOI: https://doi.org/10.3390/w15061205

[9] Darby-Dowman, K., Wilson, J.M., 2002. Developments in linear and integer programming. Journal of the Operational Research Society. 53(10), 1065–1071.

[10] Hirayama, M., Guivant, J., Katupitiya, J., et al., 2019. Artificial intelligence in path planning for autonomous bulldozers: Comparison with manual operation. International Journal of Innovative Computing, Information and Control. 15(3), 825–844.

[11] Han, M., Gao, Z., Li, A., et al., 2022. An overview of high utility itemsets mining methods based on intelligent optimization algorithms. Knowledge and Information Systems. 64(11), 2945–2984.

[12] Cristóbal, J., Guillén-Gosálbez, G., Jiménez, L., et al., 2012. Multi-objective optimization of coal-fired electricity production with CO_2 capture. Applied Energy. 98, 266–272.

[13] Wu, Y.-C., Feng, J.-W., 2018. Development and application of artificial neural network. Wireless Personal Communications. 102, 1645–1656.

[14] Zhang, B., Gong, X., Wang, J., et al., 2022. Nonstationary fuzzy neural network based on FCMNET clustering and a modified CG method with Armijo-type rule. Information Sciences. 608, 313–338.

[15] Karpinski, M., 2002. Approximability of the minimum bisection problem: An algorithmic challenge. In: Diks, K., Rytter, W. (Eds.). Mathematical Foundations of Computer Science, Vol 2420. Springer: Berlin, Germany. pp. 59–67.

[16] Dang, C., Ma, W., Liang, J., 2009. A deterministic annealing algorithm for approximating a solution of the min-bisection problem. Neural Networks. 22(1), 58–66.

[17] Wu, Z., Karimi, H.R., Dang, C., 2019. A deterministic annealing neural network algorithm for the minimum concave cost transportation problem. IEEE Transactions on Neural Networks and Learning Systems. 31(10), 4354–4366.

[18] Fang, L., Wei, L., Yi, G., et al., 2017. Research of potential safety hazard investigation and risk control system for mine enterprise. In Proceedings of the 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China. pp. 523–527.

[19] Goodman, G.V., Sarin, S.C., 1988. A mathematical programming approach for scheduling equipment in a surface coal mining operation. International Journal of Mining and Geological Engineering. 6(4), 327–341.

[20] Zhang, L., Yang, W., Hao, B., et al., 2023. Edge computing resource allocation method for mining 5G communication system. IEEE Access. 11, 49730–49737.

[21] Jiang, S., Dang, C., 2021. A more efficient deterministic annealing neural network algorithm for the max-bisection problem. Neurocomputing. 458, 428–439.

[22] Xiao, W., Belta, C., 2021. High-order control barrier functions. IEEE Transactions on Automatic Control. 67(7), 3655–3662.

[23] Deb, S., Fong, S., Tian, Z., et al., 2016. Finding approximate solutions of NP-hard optimization and TSP problems using elephant search algorithm. The Journal of Supercomputing. 72, 3960–3992.

[24] Sazlı, M.H., 2006. A brief review of feed-forward neural networks. Communications Faculty of Sciences University of Ankara Series A2–A3: Physical Sciences and Engineering. 50(1).

[25] Oliveira, I.F., Takahashi, R.H., 2020. An enhancement of the bisection method average performance preserving minmax optimality. ACM Transactions on Mathematical Software. 47(1), 1–24.

[26] Patil, S.V., Kulkarni, D.B., 2021. Graph partitioning using heuristic Kernighan-Lin algorithm for parallel computing. In: Deshpande, P., Abraham, A., Iyer, B., et al. (Eds.). Next Generation Information Processing System: Proceedings of ICCET 2020, vol 1162. Springer: Singapore. pp. 281–288.

[27] Hagberg, A., Conway, D., 2020. NetworkX: Network analysis with Python. Available from: https://networkx.github.io (cited 9 July 2025).