# Optimization Competition Report: Method on Topology Optimization of Specific Heat Conduction Problems Based on Genetic Algorithm

***Name: Hongze Song ID: 124280910034***

## Abstract

• **Problem:** For a two-dimensional "surface-point" thermal conductivity topology optimization benchmark problem discretized into a 51×51 grid, the material layout is optimized to minimize the average temperature of the system under the dual constraints that the proportion of high thermal conductivity materials is 15% and their distribution is a single connected domain.

• **Method:** After comparing multiple heuristic algorithms such as simulated annealing and particle swarm optimization, the genetic algorithm (GA) was finally selected as the core framework, and an innovative connectivity repair operator was designed for it to handle strong constraints. Through systematic analysis and tuning, the key hyperparameters of the algorithm (such as crossover/mutation rate, population size) were determined.

• **Results:** The algorithm successfully converged and obtained a topological structure with high performance and tree-like fractal features. The report deeply analyzes the convergence process of the algorithm, the morphological characteristics of the optimal solution, and explores the sensitivity of the key hyperparameters to the optimization results.

• **Conclusion:** This study verifies the effectiveness of the proposed genetic algorithm with constraint repair mechanism in solving such problems, and provides theoretical analysis and practical guidance for the parameter selection of the algorithm.

## 1 Introduction

### 1.1 **Research Background**

Topology optimization is an advanced structural design methodology that seeks the optimal material distribution within a given design domain to achieve superior performance. Compared to conventional sizing and shape optimization, topology optimization offers the greatest design freedom, often leading to novel and highly efficient innovative configurations.

Driven by the rapid advancement of computational power, it has become a research hotspot in engineering and material science, with wide-ranging applications in the lightweight design of aerospace components, acoustic metamaterials, and the thermal management of microelectronic devices. Particularly in high-power-density electronics, the efficient dissipation of internally generated heat has emerged as a critical bottleneck determining device performance and reliability. Topology optimization provides a powerful tool for designing solutions to such thermal challenges.

### 1.2 Problem Description

This study focuses on a classic "volume-to-point" heat conduction problem, the physical scenario of which is depicted in the provided figure. Within a two-dimensional rectangular domain, heat is generated uniformly, and it must be conducted to a fixed boundary region acting as a heat sink. The domain can be filled with two materials of vastly different thermal properties: a material with very high thermal conductivity, which may be costly or heavy, and a low-conductivity base material.

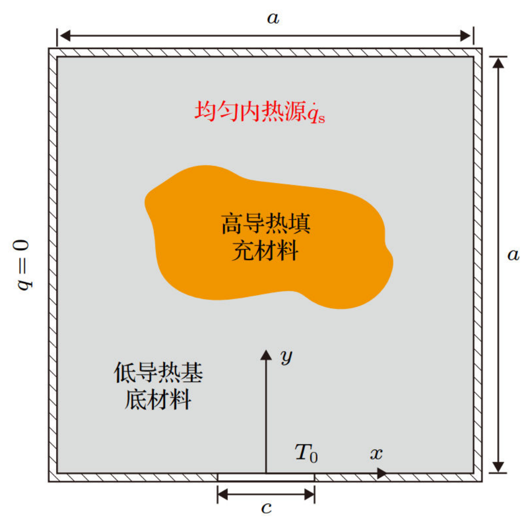



**Figure 1: The schematic diagram of the VP problem.**

The optimization objective is to find the optimal spatial layout of the high-conductivity material to minimize the steady-state average temperature of the entire system. This optimization process, however, is governed by two strict engineering constraints:

1. **Volume Fraction Constraint:** For reasons of cost or weight, the total amount of high-conductivity material is fixed, specified in this problem as 15% of the total domain area.
2. **Connectivity Constraint:** To ensure the formation of an effective heat-conducting pathway, all units of the high-conductivity material must be interconnected, forming

a single, continuous topological domain that efficiently channels heat from all parts of the domain to the heat sink.

### 1.3 Research Purpose and Report Structure

To address the aforementioned challenges, this study aims to develop and apply a Genetic Algorithm (GA) based optimization strategy to efficiently solve this volume-to-point heat conduction problem by tailoring a specialized constraint-handling mechanism. The core of this research lies in the design of an operator that automatically repairs the connectivity of solutions and in the systematic analysis of the algorithm's performance and the resulting topology.

The remainder of this report is organized as follows: Chapter 2 establishes the mathematical model for the optimization problem. Chapter 3 details the selected Genetic Algorithm framework, with an emphasis on its encoding scheme, genetic operators, and the crucial connectivity repair algorithm. Chapter 4 presents and provides an in-depth discussion of the optimization results, including the algorithm's convergence, the morphological features and evolution of the optimal topology, and a hyperparameter sensitivity analysis. ly, Chapter 5 concludes the paper with a summary of findings and an outlook for future work.

## 2 Mathematical Formulation of the Opotimization

### 2.1 Discretization Model and Decision Variables

The continuous two-dimensional design domain $\Omega$ is discretized into a uniform rectangular grid, comprising $N = n_x \times n_y = 51 \times 51 = 2601$ square cells. For each cell $e$ in the grid (where $e = 1, 2, ..., N$), a binary decision variable $\rho_e$ is introduced to represent the material property at that location:

$$\rho_e = \begin{cases} 1, & \text{if } e \text{ is filled with high thermal conductivity material} \\ 0, & \text{if } e \text{ is filled with low thermal conductivity material} \end{cases}$$

Consequently, the topology of the entire domain can be uniquely defined by a binary vector of length 2601, $\boldsymbol{\rho} = \rho_1, \rho_2, ..., \rho_N$ . This vector $\boldsymbol{\rho}$ constitutes the solution space for our optimization problem. The thermal conductivity of each cell $k_e$, can be assigned based on the decision variable as follows:

$$k_e(\rho_e) = (1 - \rho_e)k_0 + \rho_e k_1$$

### 2.2 Objective Function and Constraints

- **Objective Function**

Our goal is to minimize the steady-state average temperature, $\overline{T}$, of the entire domain. For a given topology $\boldsymbol{\rho}$, the corresponding temperature field $T_e(\boldsymbol{\rho})$ is obtained by solving the steady -state heat conduction equation. The objective function $f(\boldsymbol{\rho})$ is defined as:

$$\min_{\boldsymbol{\rho}} f(\boldsymbol{\rho}) = \overline{T}(\boldsymbol{\rho}) = \frac{1}{N}\sum_{e=1}^{N} T_e(\boldsymbol{\rho})$$

- **Constraints**

Volume Fraction Constraint: The total number of cells occupied by the high-conductivity material must be strictly equal to a preset value, which is 15% of the total number of cells.

$$\sum_{e=1}^{N} \rho_e = n_{high} = 390$$

Connectivity Constraint: All cells with a value of 1 (i.e., high-conductivity material) must form a single, continuous entity in space. In this study, we adopt the 8-connectivity criterion, meaning two cells are considered connected if they share an edge or a vertex.

## 3 Algorithm Selection and GA Design

### 3.1 Algorithm Selection and Comparison

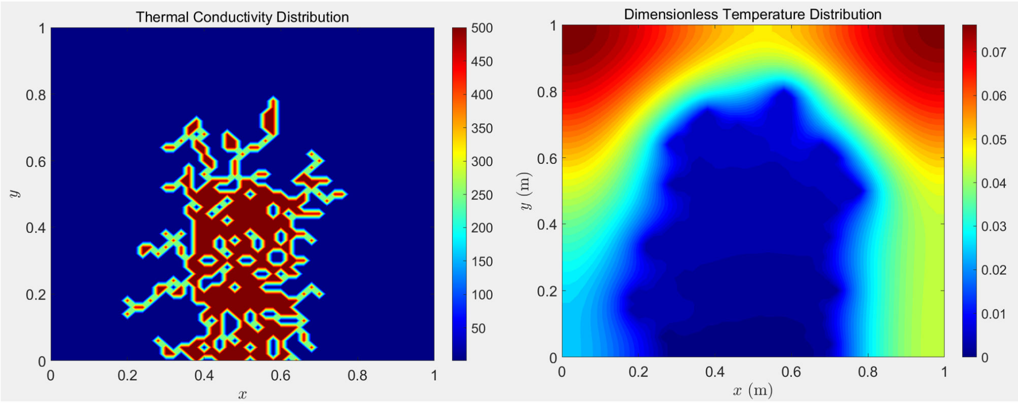
In selecting the core algorithm, we evaluated several classic heuristic methods, including Simulated Annealing (SA), Particle Swarm Optimization (PSO), and Genetic Algorithm (GA).

① **Candidate 1:** Simulated Annealing (SA)

**Advantages:** The algorithm's principle is simple and easy to implement. Its probabilistic "jumping" mechanism can theoretically help escape local optima.

**Disadvantages and Unsuitability:** SA is a single-point search algorithm. For this problem's vast search space of $51 \times 51 = 2601$ dimensions, its serial search is extremely inefficient. More critically, SA's neighborhood search operations (e.g., swapping a pair of 0 and 1 cells) would frequently violate the connectivity of the high-conductivity material distribution. Each violation would necessitate a call to a complex and time-consuming repair function, causing the algorithm to spend the vast majority of its time repairing solutions rather than exploring better structures, leading to an exceedingly slow convergence process.

**Results of SA**




***The average dimensionless temperature is: 2.932414e-02***

**Figure 2: Thermal conductivity and temperature distribution results of SA**

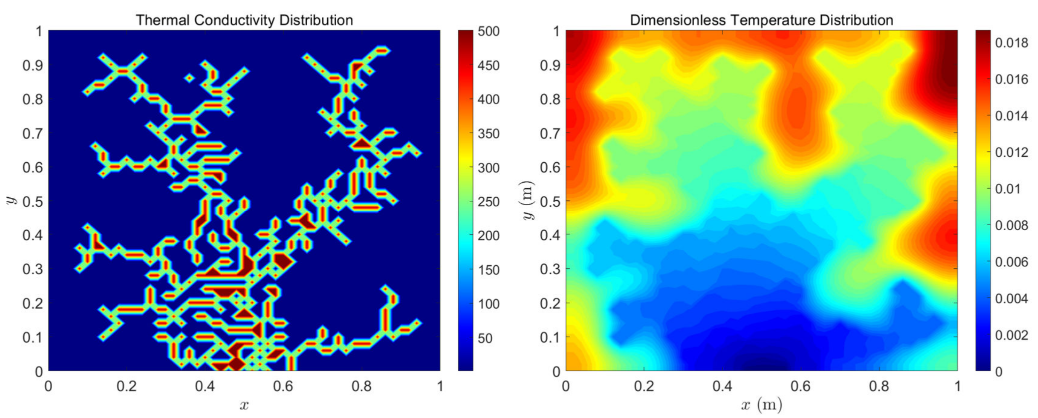
This form is a typical manifestation of the algorithm converging prematurely and falling into a local optimum. Although the SA algorithm can easily form these simple cluster structures locally, its neighborhood search mechanism is difficult to overcome the huge "energy barrier" to make large-scale, structural changes, so it cannot evolve to a better tree structure, which ultimately leads to poor heat dissipation performance.

② **Candidate 2:** Particle Swarm Optimization (PSO)

**Advantages:** As a swarm intelligence algorithm, it often exhibits fast convergence through collaboration and information sharing among particles.

**Disadvantages and Unsuitability:** The classic PSO algorithm is designed for continuous optimization problems. Although variants like "Binary PSO" exist, its core concepts of "particle velocity" and "position update" are difficult to map intuitively and effectively to the discrete topological changes in this problem. A particle's position update is inherently global, which would completely destroy the existing topology, making the connectivity constraint nearly impossible to maintain or repair effectively.

**Results of PSO**




***The average dimensionless temperature is: 9.333457e-03***

**Figure 3: Thermal conductivity and temperature distribution results of SA**

### ③ Final Choice: Genetic Algorithm (GA)

**Suitability and Advantages:**

1. Naturally Suited for Discrete Optimization: The "chromosome-gene" encoding of a GA perfectly matches the binary decision vector $\boldsymbol{\rho}$ of our problem.
2. Powerful Global Search Capability: Through its Crossover and Mutation operators, GA can perform an efficient parallel search across the entire vast solution space.
3. Ease of Integrating Constraint Handling: GA's "generate-evaluate" generational evolution model provides great convenience for handling strong constraints. We can apply the **connectivity repair operator** centrally and efficiently after generating a new population of offspring, ensuring that all individuals entering the next round of selection are **feasible solutions** that satisfy all constraints. This structural decoupling and convenience are unparalleled by SA and PSO.

**Table 1: Summary table of the best results of each algorithm**

| Algorithm | Best results (lowest temperature) |
| --- | --- |
| Simulated Annealing | $29.3\times10^{-3}$ |
| Particle Swarm Optimization | $9.33\times10^{-3}$ |
| **Genetic Algorithm** | $\mathbf{6.34\times10^{-3}}$ |

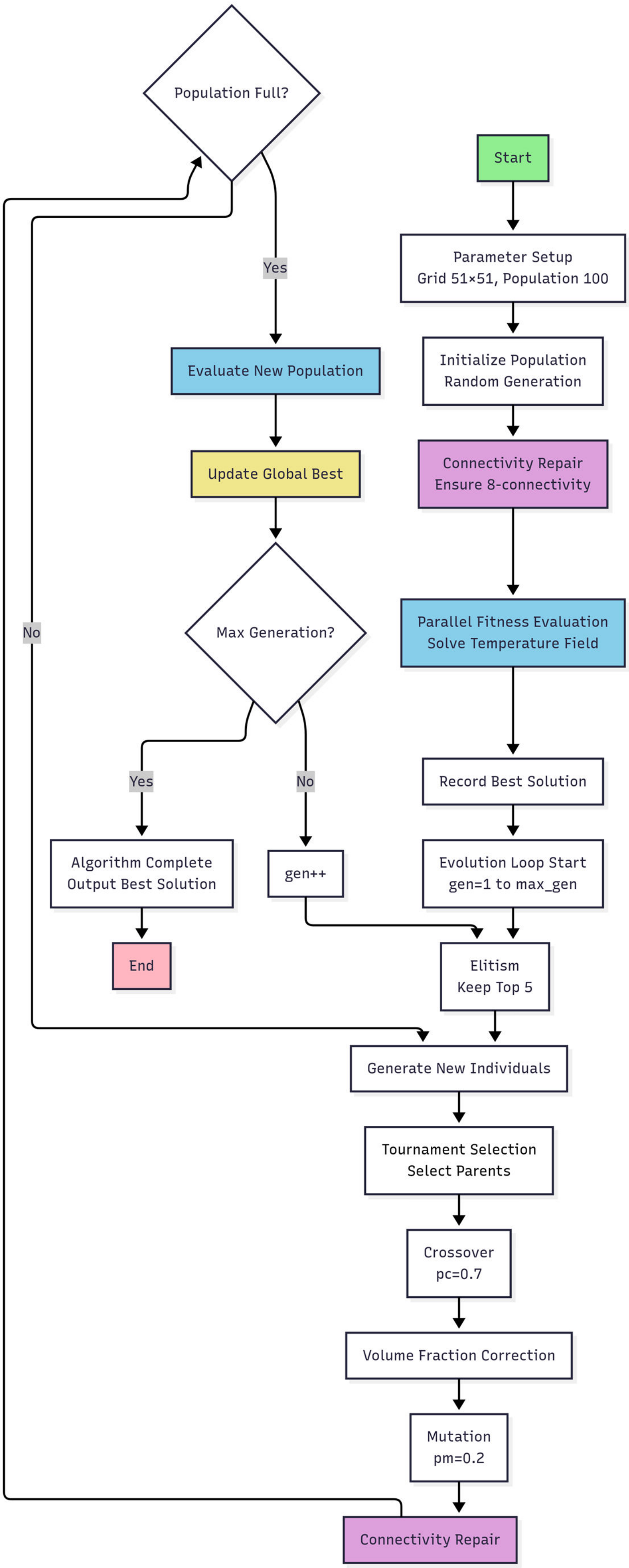



**Figure 4: Flowchart of the Genetic Algorithm**

### 3.2 Encoding and Population Initialization

**Encoding:** Each individual (a potential topology) is encoded as a binary chromosome of length $N = 2601$, corresponding directly to the decision vector $\boldsymbol{\rho}$. The $i$-th gene on the chromosome being 1 or 0 represents the corresponding cell being made of high- or low-conductivity material, respectively.

**Population Initialization:** We create an initial population of $pop\_size$ individuals. For each individual, we randomly set $n_{high} = 390$ genes to 1 and the rest to 0. This procedure ensures that all individuals in the initial population satisfy the volume fraction constraint from the outset.

### 3.3 Genetic Operators

**Selection:** We use *Tournament Selection*. In each selection round, *k* individuals (where *k=3* in this study) are randomly chosen from the current population to form a group. The individual with the best fitness (i.e., the lowest average temperature) in this group is then selected as a parent for the next stage.

**Crossover:** A *Two-dimensional Block Crossover strategy* is employed. For two parent individuals, with a probability of $p_c$, several rows are randomly selected, and all their genes in these rows are swapped. This method preserves the spatial proximity of genes within a row. After crossover, the volume fraction of the offspring may no longer be 390. We correct this by randomly flipping excess 1s to 0s or insufficient 0s to 1s.

**Mutation:** An individual undergoes mutation with a probability of $p_m$. The operation involves randomly selecting one gene with a value of 1 and one with a value of 0, and swapping their values. This step ensures that the volume fraction constraint remains satisfied after mutation.

### 3.4 Key Innovation: The *repair_connectivity* Algorithm

To ensure that every individual participating in fitness evaluation and selection is a feasible solution that satisfies the connectivity constraint.

**Algorithmic Steps and Strategy Analysis:**

① Identify and Preserve the Largest Connected Component: First, the algorithm iterates through all cells with a value of 1 in an individual and uses *Breadth-First Search (BFS)* or *Depth-First Search (DFS)* to find all 8-connected components. It then preserves only the largest connected component (the one with the most cells) and sets the value of cells in all other components from 1 to 0. This step provides the most stable "topological skeleton" for the subsequent repair.

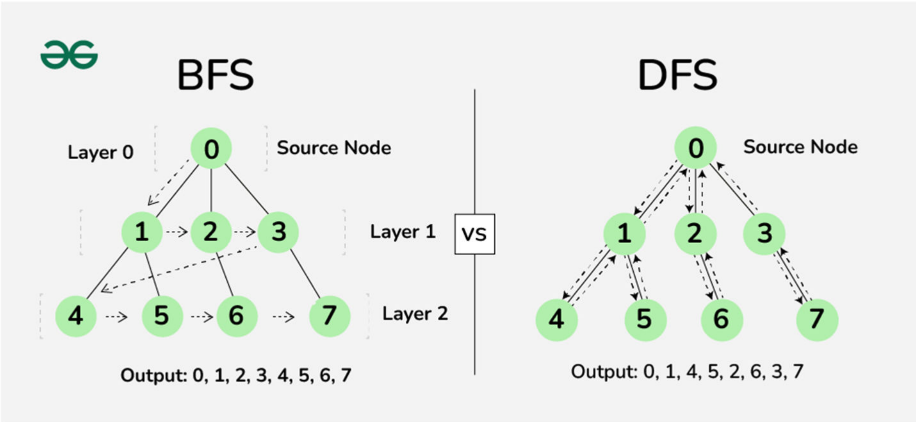



**Figure 5: 8-connected components finding algorithm**

② *Heuristic Growth Strategy*: After the previous step, the number of cells with a value of 1 is typically less than 390. The algorithm needs to "grow" new high-conductivity cells to make up the deficit.

③ *Boundary Point Identification*: It identifies all boundary points of the current largest connected component—those cells with a value of 1 that are adjacent to at least one cell with a value of 0.

④ *The "Minimum-Adjacency" Growth Strategy*: This is the core heuristic idea of this repair operator. The algorithm identifies all candidate growth points (cells with a value of 0 adjacent to the current component) and counts how many times each candidate is adjacent to the component. When selecting the next cell to turn into a 1, the algorithm preferentially selects randomly from those candidate points with the minimum number of adjacencies to the current component.

Mechanism Analysis: The geometric meaning of this "minimum-adjacency" strategy is to inhibit the topology from becoming "fat" and "bloated" inwards. Instead, it encourages exploration outwards along the **weakest connections, forming slender "branches"**. This intrinsic mechanism naturally favors the generation of **tree-like, branching structures** that align with the physics of heat conduction. It not only efficiently repairs connectivity but also embeds morphological heuristics into the repair process, guiding the optimization towards high-quality solutions.

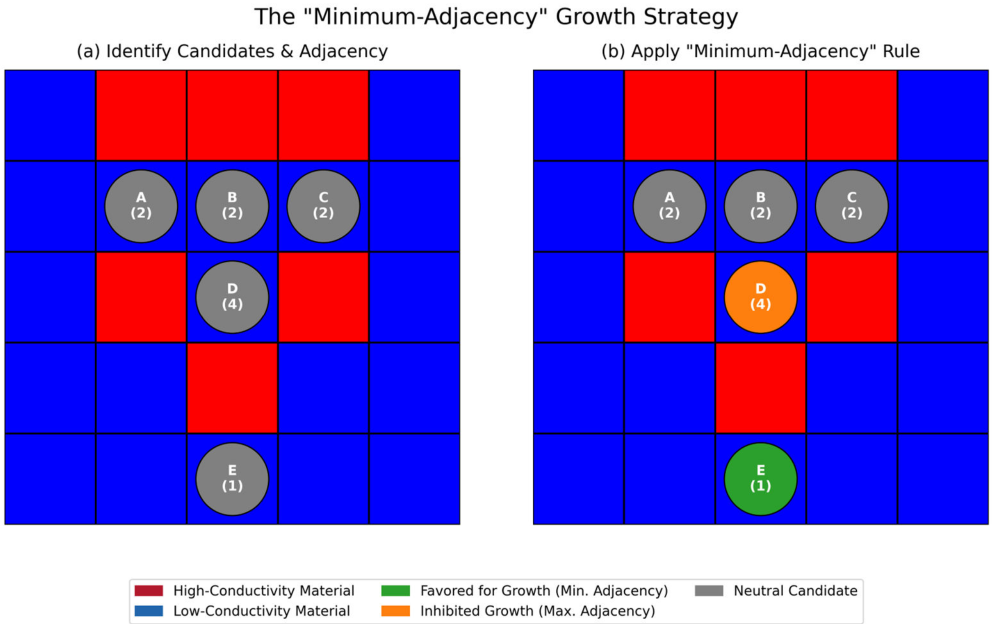



**Figure 6: Illustration of the "Minimum-Adjacency" Growth Strategy. Panel (a) displays the initial state with all candidate growth points (A-E) marked in grey; the number within each circle is its adjacency count to the high-conductivity material. Panel (b) shows the outcome after applying the "minimum-adjacency" rule: candidate E, having the minimum adjacency count (1), is marked in green and is favored for growth to encourage the formation of a new branch. In contrast, candidate D, with the maximum adjacency count (4), is marked in orange and is inhibited to prevent the structure from becoming bloated.**

### 3.5 Hyperparameter Selection and Tuning

The performance of a GA is highly dependent on the setting of its hyperparameters. We tuned the key hyperparameters through a combination of experience and multiple experiments using a control-variable approach, seeking the optimal balance between convergence speed and solution quality.

- **Population Size (*pop_size*):** Affects population diversity. Too small, and it can lead to premature convergence to a local optimum; too large, and the computational cost becomes prohibitive. We found ***pop_size=100*** to be an ideal balance.
- **Crossover Probability (*pc*):** Controls the rate at which new structures are generated. Too high can disrupt good patterns; too low can lead to search stagnation. Experiments showed that **pc=0.7** provides good exploratory power.
- **Mutation Probability (*pm*):** Key to maintaining population diversity. Thanks to our powerful repair operator, a slightly higher mutation rate can be tolerated to escape local optima. **pm=0.2** was confirmed to effectively maintain population vitality.
- **Elite Count (*elite_num*):** Ensures that the best-so-far individuals are not lost.

**elite_num=5** (5% of the population) is a common and effective strategy.

The final hyperparameter configuration is summarized in the table below:

**Table 2: Hyperparameter configuration summary table**

| Hyperparameter | Symbol | Value |
|---|---|---|
| Population Size | *pop_size* | 100 |
| Max Generations | *max_gen* | 10000 |
| Elite Count | *elite_num* | 5 |
| Crossover Probability | *pc* | 0.7 |
| Mutation Probability | *pm* | 0.2 |

# 4 Results and Discussion

## 4.1 **Algorithm Convergence Analysis**

The convergence of the algorithm is the primary indicator of its effectiveness. Figure 5 shows the convergence curve of the fitness (i.e., the global average temperature) of the best individual in the population as a function of the generation number during the optimization process.

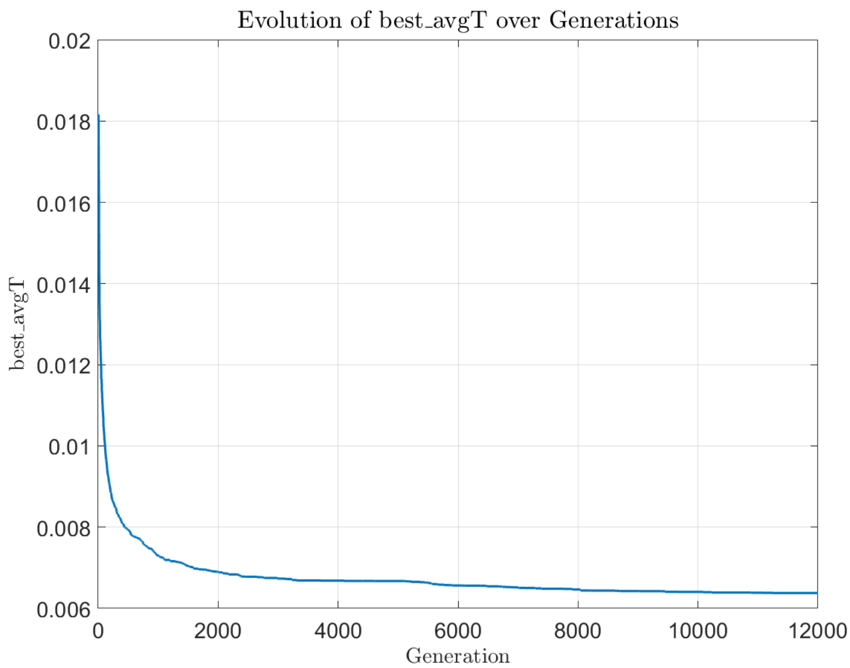



**Figure 7: Algorithm convergence curve. The x-axis represents the generation number, and the y-axis represents the fitness value (normalized average temperature) of the best individual in each generation.**

The convergence curve exhibits clear and logical phases. Initially, the average temperature drops sharply, reflecting the algorithm's process of rapidly eliminating a vast number of chaotic, random topologies to establish a fundamental conductive pathway. Subsequently, the rate of convergence decelerates into a prolonged phase of steady refinement, where the algorithmic focus shifts from building the main trunk to the fine-tuning and optimization of the branching network. ly, the curve flattens, indicating that the algorithm has converged to a high-quality, stable solution, beyond which further iterations yield only marginal improvements.

### 4.2 Optimal structure analysis

After 12,000 steps of iterative optimization, the algorithm finally obtained the optimal topological structure and temperature field distribution as shown in Figure 4.2. In the figure, red represents high thermal conductivity materials, and blue represents low thermal conductivity materials.

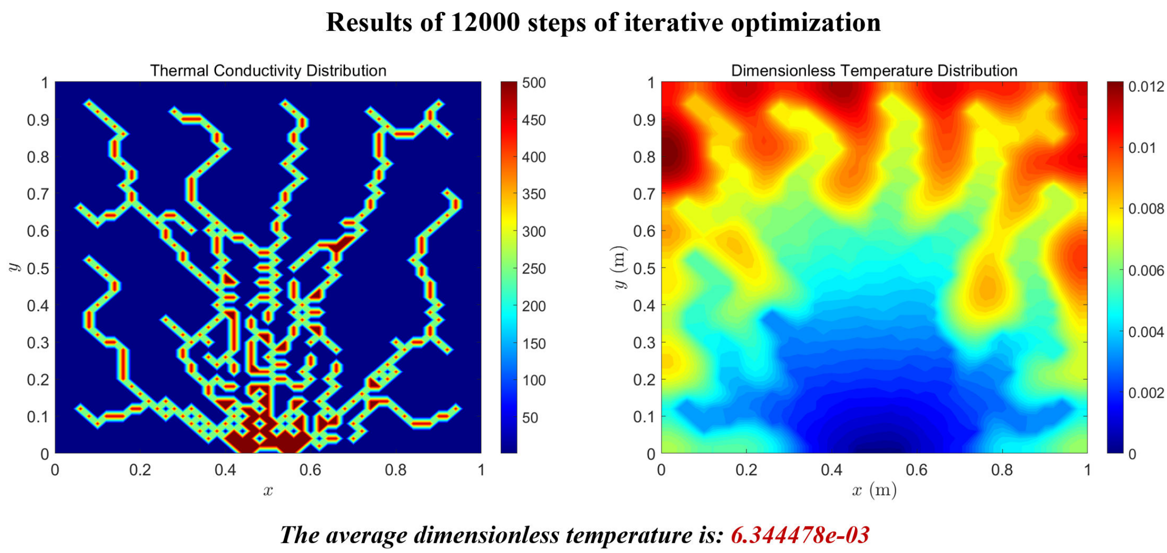



**Figure 8: The final optimal topology obtained from optimization.**

The final topology exhibits highly organized tree-like fractal features. It consists of a thick "main trunk" connected to the heat sink boundary at the bottom, and a complex network of progressively finer "branches" extending upwards and sideways from the trunk.

### 4.3 Structural evolution process

To gain a deeper understanding of how the optimal structure was formed, we traced the evolution of its morphology at different stages of the optimization, as shown in Figure 4.3.

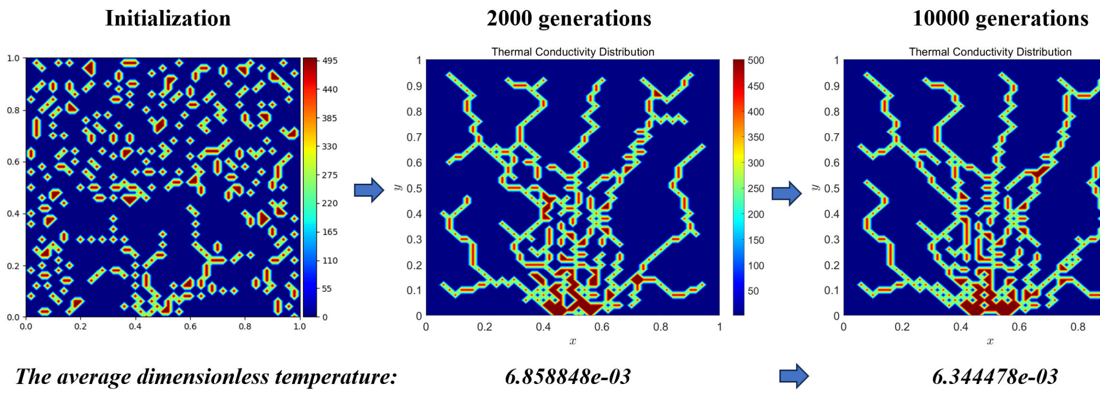



**Figure 9: Evolution path of the topology. It shows the morphology of the best individual in the population at early (e.g., Gen 10), middle (e.g., Gen 500), and final generations.**

**Initialization (Generation 1):** The topology appears as several isolated, randomly distributed "islands" or short "worm-like" structures. At this point, the algorithm's primary task is to establish basic connectivity by linking these scattered material patches.

**Intermediate Phase (approx. Generations 2000):** In this stage, the main trunk structure of the conduction path has been largely established. The algorithm's focus shifts to the "fine-tuning" of the branch structures, optimizing details through mutation and local adjustments to further enhance thermal efficiency.

**Final Phase (approx. after Generation 10000):** This indicates that the algorithm has found a high-quality optimal solution, and subsequent iterations are unlikely to yield significant performance improvements. In fact, if we continue to run the program, the results will be better, but it will take more time and the benefits will be very small.

### 4.4 Hyperparameter Sensitivity Analysis

This section aims to discuss the sensitivity of the optimization results to key hyperparameters to deepen the understanding of the algorithm's behavior.

① **Effect of Population Size (*pop_size*):**

A small population (pop_size=50) would suffer from a severe lack of diversity, making it highly susceptible to "premature convergence" to a simple, far-from-optimal topology.

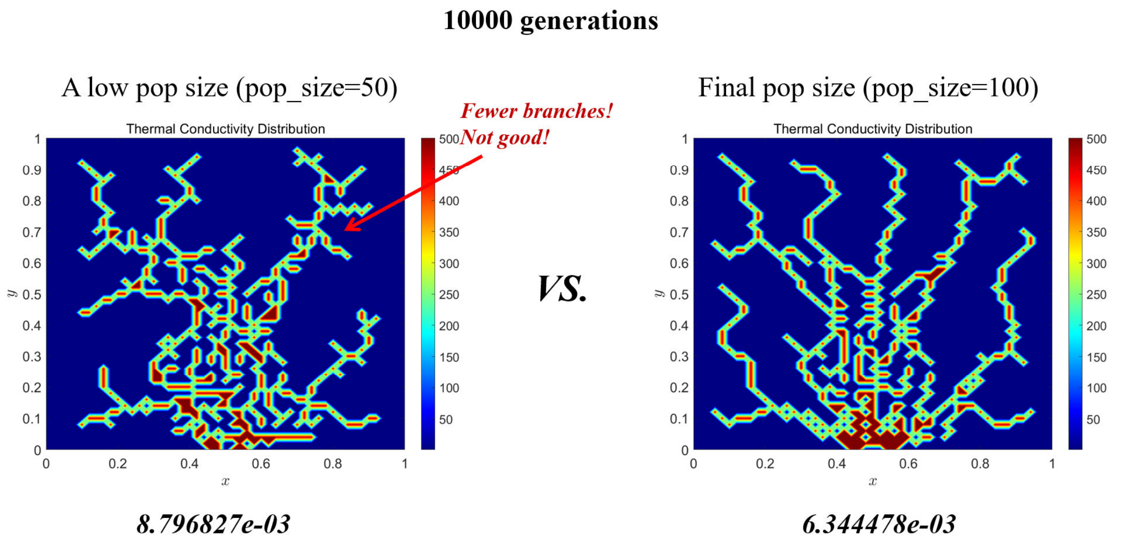



**Figure 10: Comparison of results with different pop sizes under the same generations**

A large population (pop_size=200) would enable a more extensive global search, reducing the risk of getting trapped in local optima and thus evolving more complex and refined topologies. However, this comes at the cost of a sharp increase in computational expense. In the actual running process of the program, one iteration of the GA algorithm takes 5 minutes, which means that **1000 iterations** take **83+** hours!!!

② **Effect of Crossover Probability (*pc*):**

A low crossover rate (pc=0.4) would mean that offspring are more likely to be direct copies of their parents. The ability to recombine genes would be limited, making it difficult for good "topological schemas" to propagate and combine effectively within the population, slowing down convergence.

A high crossover rate (e.e., pc=0.9) could be disruptive to well-formed topological patterns. However, because the row-swapping crossover used in this study is relatively less destructive, and is backed by a powerful repair operator, a higher crossover rate (e.g., 0.7) can still achieve good results.

③ **Effect of Mutation Probability (*pm*):**

A low mutation rate (pm=0.05) would lead to a rapid loss of diversity in the later stages. Once trapped in a local optimum, it would be difficult to escape through small perturbations, potentially resulting in final topologies with obvious local defects.

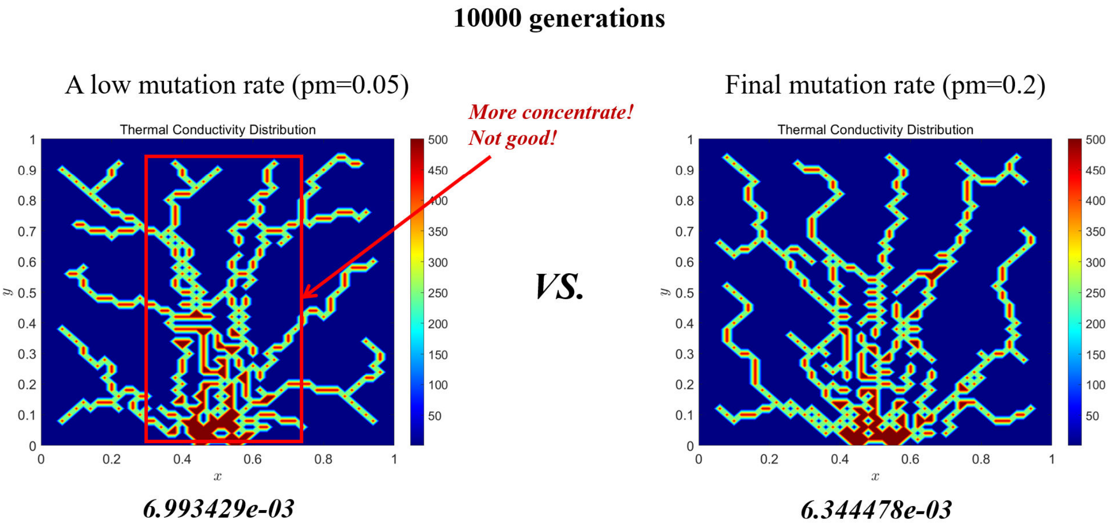



**Figure 11: Comparison of results with different mutation rate under the same generations**

A high mutation rate (pm=0.5) would cause the algorithm to degenerate into a nearly random search. The evolutionary pressure would be overwhelmed by random noise, preventing the formation of stable, ordered structures. The convergence curve would oscillate violently instead of declining steadily.

# 5 Conclusion

## 5.1 Summary of Conclusions

This study successfully applied a genetic algorithm (GA) with an innovative constraint repair mechanism to solve the two-dimensional "surface-to-point" thermal conductivity topology optimization problem under the dual constraints of given volume fraction and connectivity. The superiority of the genetic algorithm in dealing with such problems with strong topological constraints was verified through comparative analysis with simulated annealing (SA) and particle swarm optimization (PSO) algorithms.

In particular, the core ***repair_connectivity*** repair operator can effectively handle connectivity constraints, converge stably and find high-performance feasible solutions.

## 5.2 Future work prospects

Although the current repair operator has achieved good results, its strategy - forcing the preservation of a unique maximum connected component at each repair - may be too strict in some cases. This "winner takes all" mechanism may prematurely eliminate some temporarily separated but highly potential secondary structure "building blocks".After listening to the sharing about optimization competition in class, I think further improvements can be made in the following two aspects in the future.

1. Allow multiple domains to coexist and evolve: In the early and middle stages of the algorithm, multiple unconnected secondary connected domains are allowed to exist within individuals. The role of the repair operator changes from "forcing a single" to "optimizing each", allowing these secondary connected domains to grow and optimize in their respective local areas as independent "seeds".
2. Introducing an intelligent bridging mechanism: When the algorithm reaches a certain stage (for example, when the convergence speed slows down or reaches a preset number of generations), a "bridging" module is activated. The specific task of this module is to intelligently establish connections between these fully optimized secondary connected domains with minimal material cost or minimal performance damage, and finally integrate them into a single, more complex connected network.