

**Status:**

The given current structure at iteration : **{iteration}**/25 did not meet all constraints. Generate a closed 2D truss structure that satisfies the constraints. Current max(abs(stress)) is **{c\_max\_stress}** and total mass is **{total\_mass}**.

**Inputs:**

- Loads: Force information as (node\_id: (magnitude, direction)): **{load}**
- Supports: Information about supports on nodes **{supports}**
- node\_dict: Node position of current structure **{node\_dict\_c}**
- member\_dict: Member connection of current structure **{member\_dict\_c}**
- area\_id: Area ID values to choose cross section from **{area\_id}**
- FEM Results: Fem result of current structure **{fem\_result}**
- Structure Generation History: Ranked Structure-score pair for optimization: **{history}**

**Objectives:**

1. Generate the optimal closed truss structure that minimizes stress, while satisfying all given constraints.

**Constraints:**

1. Max stress (compressive(-) and tensile(+)) <= ±**{max\_stress}**
2. Total mass: Mass (sum of member length × area from **{area\_id}**) <= **{max\_weight}**

**Instructions:**

1. Treat this as a single-shot optimization task aimed at resolving constraint violations.
2. Begin with a step-by-step analysis of inputs, current design, and structural generation history. Identify which constraints are violated and why.
3. Use available history, current design, and your structural reasoning to decide whether to improve the existing structure, redesign it, or reuse a historic structure.
4. Prioritize feasibility: Ensure that the final design satisfies all constraints before considering optimization.
5. Design a closed (fully connected) 2D truss — all nodes must be structurally connected, with no isolated nodes.
6. You may add, remove, or fine tune positions of nodes as needed.
7. Plan member connections and node positions to satisfy all constraints and minimize stress.
8. Use precise float values for node coordinates. Avoid rounding or snapping.
9. Use only area\_id values from **{area\_id}** as string keys. No duplicates in node or member definitions.
10. Estimate stress and mass using basic structural principles (e.g., static equilibrium, trigonometry). Clearly state any assumptions made.
11. Ensure the final output is a fully consistent and valid Python dictionary.
12. Return only a structure that fully satisfies the constraints and is optimal within your reasoning capability.
13. Do not change or modify node positions with loads or supports.

**Output format (python dictionary):**

Return a single valid Python dictionary in the format below. Keep all string fields concise. Use floats for coordinates.

```
{}  
  "preamble": "Briefly describe the problem, current design status, and the optimization objective.",  
  "scratch_pad_1": "Initial interpretation of given structure objectives and constraints. This is the first iteration",  
  "analysis": "Detailed analysis of the structural problem based on current design, structure generation history and current status. Include interpretation of load paths, boundary conditions, possible failure modes, and efficiency of current structure and structures in provided in generation history.",  
  "scratch_pad_2": "Reflect on the analysis. Note any patterns, areas of concern, or insights that may guide the design.",  
  "proposal": "Based on current design, history and analysis, decide whether to optimize current design or redesign or use historic structures. Provide a brief description of the proposed structure or modifications to the current design.",  
  "reasoning": "Use design principles (e.g., symmetry, triangulation, load distribution) to justify the design direction. Propose structural changes or node repositions to improve performance.",  
  "scratch_pad_3": "Outline key assumptions, set-up equations (e.g., force balance, mass calculation, trigonometric equations etc.), and calculations to be performed. Include considerations for member selection or node movement.",  
  "calculation": "Perform required calculations such as updated member forces, stress, and structure mass. Quantify constraint violations if any, and iterate on modifications if needed",  
  "scratch_pad_final": "Final scratch pad before you generate the structure verify adherence to objective, constraints and Instructions. Perform modifications to ensure the design meets all requirements.",  
  "node_dict": {},  
    # Node coordinates must be precise floats (rounded to 1e-2) based on structural calculations. Do not snap to 0.5 or 0.25.  
    # Each node entry should include design reason, that will help future optimization.  
  "node_1": (x.xx, y.yy), # thought, reason, design choice, calculation  
  "node_2": (x.xx, y.yy), # thought, reason, design choice, calculation  
},  
  "member_dict": {example_member},  
}  
}
```