

Status:

The given current structure at iteration : **{iteration}**/25 satisfies all the constraints. Minimize stress while ensuring the generated closed 2D Truss remains within 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}**
- members_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 an optimization task. Use current status, current design, and generation history to decide whether to improve or redesign.
2. Begin with a self-reflection on current structure and step-by-step analysis of inputs, current design, and performance. Identify constraint violations or inefficiencies.
3. Decide based on history and current design: either improve the current design or generate a new, more optimal one from scratch, or stop.
4. Fine-tune positions of nodes as needed — except for nodes with fixed loads or supports.
5. Ensure all nodes are part of a single connected (closed) structure. No floating or isolated nodes.
6. Plan member connections and node placements to minimize stress while satisfying all constraints.
7. Use precise float values for node positions. Avoid rounding unless justified by geometry.
8. Estimate mass and stress using valid structural reasoning. Clearly state assumptions and methods.
9. Use only area_id values from **{area_id}** (string keys). Avoid duplicates in nodes or members.
10. Ensure the output meets **all constraints** before finalizing.
11. Output must be a valid, fully consistent Python dictionary. Do not leave placeholders or partial entries.
12. If you are confident no further improvements are possible, return the best design and set "stop": True in the output dictionary with comment as why.
13. Only stop if constraints are satisfied and the design is already optimal within your reasoning ability.
14. 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 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.",  
    "stop": "True/False" (str(bool)) if you cant optimize further and want to stop the optimization at current step return best design so far # provide comment why stop,  
    "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},  
}
```