Generative design of bionic partition for airplane cabin interiors by Monte Carlo tree search and finite element analysis

by Georgy Tskhondiya

Project GitHub page: https://github.com/gigatskhondia/Reinforcement_Learning_and_FEA
Project Facebook page: https://www.facebook.com/GigaTsk/

1. Background and Rationale

Please see appropriate chapters in [1, 2].

2. Methodology

a) Finite element model

For the finite element model I have used the Space Frame Element from [3].

b) Monte Carlo tree search

In computer science, **Monte Carlo tree search** (**MCTS**) is a heuristic search algorithm for some kinds of decision processes, most notably those employed in game play. MCTS has been used for decades in computer Go programs. It has been used in other board games like chess and shogi games with incomplete information such as bridge and poker, as well as in real-time video games (such as Total War: Rome II's implementation in the high level campaign AI), [4].

c) Finite Element Environment to MCTS

The finite element model represents an environment to which an agent applies actions and from which it gets rewards. The agent uses MCTS to decide on its actions. Actions change geometry of the structure of aircraft component; the resulting geometry is then subjected to FEA. The agent gets rewards (win) if it meets the optimization objective of minimizing (weight of a component) and maximizing (stiffness of the structure) target values. The outcome of the modeling (produced “online” during the “game”) is an optimized design of the component.

3. Model Details

In this work, I made an agent do actions of drawing lines between grid nodes. For 5x5 grid, there were 72 possible lines (actions). I applied MCTS in a game between two players. Rules of the game were as follows: whoever drew a structure that passed through certain nodes (checkpoints), had at least two neighbors for each node and an improved strength and weight compared to the previous move, won.

I decided to apply algorithm in a form of a game between two players because of the ability to replace one player with a human engineer to assist an agent in engineering design.

The overall objective of the agent was to minimize the structure's weight while maximizing its strength. It took a while to complete the game if the number of playouts was big. The bigger the number of playouts the more intelligent the agent were.

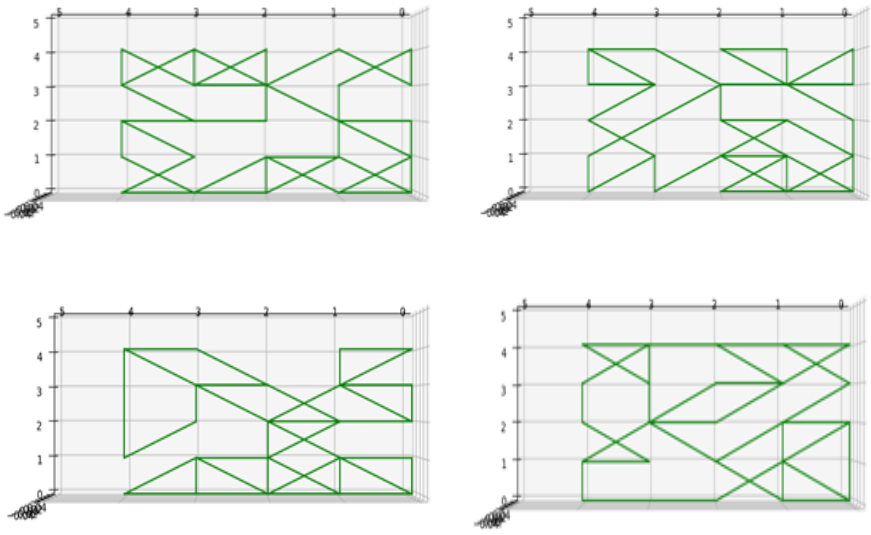

Figure 1. Sample structures obtained by MCTS and finite element analysis with 500 playouts

4. Final Words

Results of the modeling show that an agent is usually capable of producing valid outcomes within 40 game steps (I incentivized the agent to do so by adjusting the parameter of win limit) with 500 playouts .

Applying ideas from AlphaZero to generative design seems to be a plausible direction of the future research.

Codebase for the model can be found at my GitHub page (see the link at the beginning of the paper). I took the code for the pure MCTS from this GitHub repo [5].

5. Bibliography

[1] On Engineering Design by Finite Element Analysis and Deep Reinforcement Learning, Tskhondiya Georgy, https://github.com/gigatskhondia/Reinforcement_Learning_and_FEA/
[2] Generative design of bionic partition for airplane cabin interiors by reinforcement learning and finite element analysis, Tskhondiya Georgy, https://github.com/gigatskhondia/Reinforcement_Learning_and_FEA/
[3] MATLAB Guide to Finite Elements. An Interactive Approach, Peter I. Kattan, 2nd edition
[4] https://en.wikipedia.org/wiki/Monte_Carlo_tree_search
[5] https://github.com/junxiaosong/AlphaZero_Gomoku