EC final report

I have addicted to a game called ‘auto chess’ about one month ago, It’s a game consist of choosing chess and being stronger than your opponents, because I kept losing at the higher rank game, I started to consider maybe I can make an AI algorithm, some neural networks training to understand how to play this game, due to the turn-based feature of auto chess game,

Its states can be divided into discrete time, makes the training feasible.

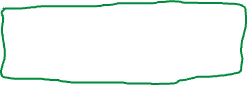


Your level, exp

Chess table and price



Money you have



Waiting field



Battle field



Reset table

**Auto chess game rule**

1.In this game, you have to compete with other seven players.

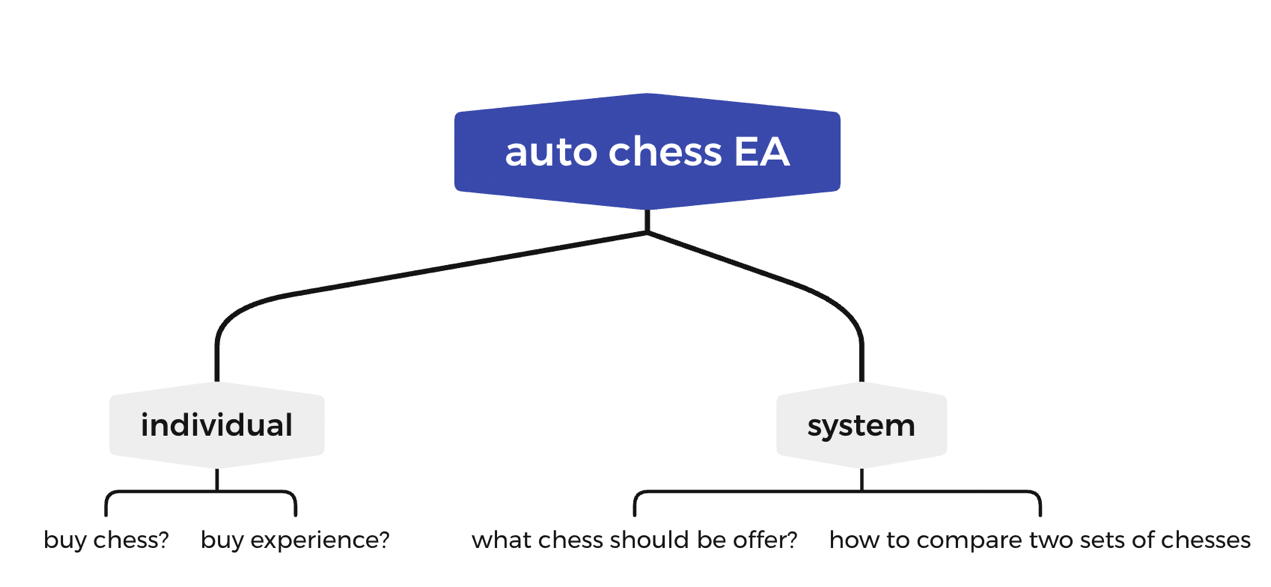
2.Every round, each player will compete with another random player, the player who gets a stronger set of chesses wins.

3.The winner will deal some damage to the loser, every player have some health, if anyone losses all its health, it will be out of the game, so, your final objective is surviving to the end.

4.For every round, the system will give you some money, which you can use to buy chesses, or you can use money to buy experience, and make you level up,

Your levels equal to the number of chesses you can put on the battle field, or you can reset the chess table to find the better chess, the chess table is like a shop that you can buy chesses on it, or, just keep your money is also a strategy.

5. The chess will be level up if you have three identical chesses (it’s an exciting moment), three level 1 chesses make a level 2 chess, three level 2 chesses make a level 3 chesses, and level 3 is the max level, and the price of chess between in 1 to 5 dollar, more expensive the chess is, more powerful and rare it will be.



My plan to accomplish this work

I was wondering, how could evolutionary algorithm applied on this auto chess game, I mean, its final purpose is to training a policy, but the EA I have used so far is always training for a set of number that fits the fitness function most, but if I simplify the problem to a single round of the game, there is always a best choice in that round, depends on several factors, so, I could assumed the single round is a fitness function, what I want is to find an action, which can be represent as a set of numbers, then the whole game will becoming a series of fitness test, the individual that gets the highest total fitness value wins, so it’s a complex version of homework EA, I think it’s better to begin my plan on an easy game setting, like 3 or even 1 chess on the board, make the action space smaller, then see what happens on the computer to decide whether we should move on.

Recent situation 2021/12/07

I have finished the system and game process, game setting part of the auto chess EA code after I wrote the proposal above, about four days ago.

(The code is also uploaded)

I found that the most challenging part is the individual deciding part, I still thinking how to implement this part, maybe I will use the pytorch linear neural network to represent the individual and decide the action it takes, and the mutation part, crossover… Confusing me as well, I will figure out how to implement them if this proposal pass…

Hope it works!