

ARTIFICIAL INTELLIGENCE TRAINING PROGRAM DESIGNER

Baran Yiğit Aladoğan - 041702005

Yasemin İzel Çangal - 041701021

Ertuğrul Yılmaz - 041701030

Introduction

This project is about programming a Training Program Designer in Python. It is a constraint satisfaction problem which means that we need to create a program according to the given rules. According to the manual, we must design a program that has 9 different subdomains that needs to be improved in score wise in a 10-week period. Initial scores of those lessons given randomly. Rules to be considered are:

- We can train 7 days in a week.
- There are 5 possible slots each day.
- You can't train the same type of lesson day after day. However, transition from Sunday to Monday is acceptable.
- User should be trained one type of training at least once a week.
- Algorithm should try to improve the performance of the least scored lessons.
- Every lesson that has been taken can improve the score by 2-4 points.

In our code, we designed a program that has three different algorithms in it. The names of these algorithms-approaches are: "With At Least 5", "With Probability", and "With Fixed Training Days According to Levels". All these algorithms have different attributes and different ways to solve the problem. Program asks the user to select a designer algorithm at the start of the process. Detailed explanation of these problems is given in the continuation of this paper. Program also has a simple UI for choosing the algorithm which is going to be used to make the schedule.

There Different Approach

1- With At Least 5

In this approach, we used all the slots in our program. Firstly, it sorts the lesson list from smallest to largest according to their scores. Then, creates two lists called as: "FiveLesson" and "FourLesson". Later, the program put the worst scored 5 lessons into the FiveLesson list and best scored 4 lessons into the FourLesson list. Then what the program does is that it puts a FiveLesson array on Monday, Wednesday, Friday, and Sunday. Then put a FourLesson array on Tuesday, Thursday, and Saturday. With this type of program, we are not violating any of the rules and generating a program that has the best progressive score possible. With that we can prioritize the 5 lessons that have the least point.

Later on, in the program, we add a random number between 2-4 to the scores of the lessons every week, which calculates a new random number in every lesson differently. Program will work like that until the goal week (10) has reached. Lastly, the code is working well.

2- With Probability

In this approach, the principles of probability were very important. Algorithm selects a random index from the schedule and checks if it is violating training program rules. If it is not violating the rules then it sends that index to a function called probabilityScores. In this function it specifies random indexs' training according to its row value. Then it calculates the level of this score. Since the program has seven training levels, function has a variable named trueFalseArray of length seven. Array has seven True values inside. Function changes some of the true values to false according to level score and shuffles the array then given a random element from the array. This element decides to

put one to that index if it is true, otherwise one is not going to be put. By using this approach high level trainings has less probability to give permission to put a one to the random index even if the index satisfies the program rules. Low level trainings will have higher chance to put one before the schedule is fully filled. But when the program stops looking at the random indexes, the answer is if there is no available cell to put a one then means that the program completed the schedule. Lastly, the code is working well.

3- With Fixed Training Days According to Levels

In this approach, there are numbers of days according to levels which are decided before. We are giving the scores randomly and then assigning the levels according to their scores. For example, level 1 needs to do training for 4 days, level 2:3 days, level 3:3 days ,...,level 7:1 days in levels_to_days array. Time to time but rarely, we can't find a proper day according to rules which are consecutive days or at most 5 subdomains can be in a day. So, we are going next step at that time without giving that day. There are fixed days but if no day is proper at that time, then we jump another step without giving a day. So, we are checking the rules as well every time. Also, we are calculating every week the post_scores and incrementation is randomly between 2-4 for every session for a subdomain. At the end, we are printing post scores with their levels too. Lastly, the code is working well.

Everybody has coded and written the report. The code is working well with all approaches and an user can see all these approaches by running once.