Evolution Is All You Need

Kai Liao Haoyuan Feng Weixin Guo
New York University
{kl3199,hf934,wg826}@nyu.edu

Abstract

In this paper, we propose a model to solve the interview scheduling problem for NYU Shanghai, which is inspired by Genetic Algorithm in Heuristic Algorithm. To overcome the NP-Hard nature of a typical timetable scheduling algorithm, we propose this model to evolve the current best solution without extra information or brute force searching in extremely large space. We apply our model on the given setting of 379 students and 24 faculties, and prove the feasibility of the model with quantitative results. Finally, we conclude our work and point out some insufficiencies and potential improvements for our work.

1 Introduction

The question we aim to solve is Topic 3, a traditional NP-Hard problem where the solution is hard to find by brute-force searching in extremely large space. Topic 3 sets the solvers near NYU Shanghai's interviewing day, where N new candidates of the undergraduate program are entering the interview process and an interview committee of M faculties is formed to question the students. Each student will be interviewed by 4 faculty members, and the five together combined as an interview group, where the faculty come up with their own questions during each interview. The task is to design an interview process subject to certain constraints.

We start by analyzing Q1 of Topic 3 in which there exists only one constraint: no interview group containing more than 1 identical interviewing faculty. Figure 1 shows the graphical representation of least number of faculties M needed for $N = 1, 2, 3, 4$. Each dot represents a faculty and each square represents a student. By induction, we get $M = 4N - \binom{N}{2}$

However, it is challenging to build a model by induction when there are more constraints, so we introduce genetic algorithm.

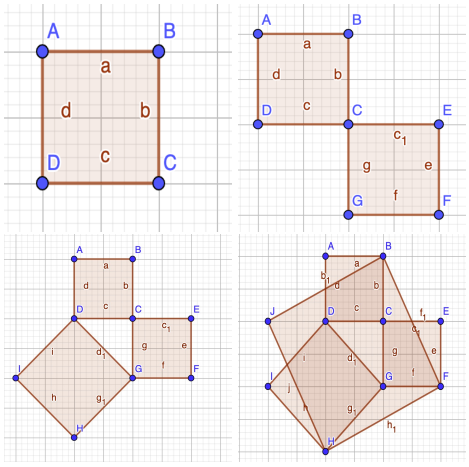

Figure 1: (a) M=4 (b) M=7 (c) M=9 (d) M=10

Natural selection is a crucial process in every species' evolution. Originally proposed by Charles Darwin, it reflects a process where the fittest of a population are selected naturally by environment and survival factors to reproduce the offspring of the next generation. In computer science, the concept of natural selection is used to create genetic algorithm. The genetic algorithm utilizes necessary elements of natural selection such as selection, mutation, crossover, and computes the fitness of the predefined population scoring the outcome of every parameter of a model in the end intending its optimization.

In this work, we propose a model to solve a more complicated problem in which, to ensure fairness, the following 4 requirements are set for the structure of the interview groups:

- R1: Each faculty member interviews similar amount of students;
- R2: Two interview groups cannot have 4 identical interviewing faculty;
- R3: Two interview groups having 2-3 identical interviewing faculty should be avoided;

- R4: The number of students being interviewed by any pair of faculty members should be as small as possible;

2 Model Description

2.1 Problem Formulation

We first describe how we formulate the problem. Since there is no information regarding interview time, availability of classroom, and faculty availability, we formulate the problem as a typical combinatorial optimization problem subject to the 4 constraints mentioned above. To proceed, we make the following assumptions: there are plenty of available classrooms to conduct interviews in parallel, all faculty members are available throughout the interview season, and each interview lasts for a fixed amount of time. For each student, we assign four different faculties, which forms an interview group. An interview schedule is represented as an $N \times 4$ matrix, where each row is an interview group. In the end, we output a list of clusters, such that faculties in a cluster are all unique, allowing all interview groups in one cluster to take place in a parallel fashion.

2.2 Theory

Our model is largely based on genetic algorithms. We define an individual as a schedule, mathematically an $N \times 4$ matrix, and a population consisting of a certain number of individuals. To simulate natural selection, we define a customized loss function to measure the fitness of each schedule in our population. We also define mutation of one schedule and cross over of two schedules to simulate the process of producing offspring. Repeating the process for a fixed number of iterations, we get the best schedule with the lowest loss. Finally, we perform bottom-up hierarchical clustering to produce a list of clusters, where each cluster consists of some number of interview groups that can take place in parallel.

The loss of a schedule consists of 4 four components, corresponding to the 4 constraints mentioned above respectively.

$$L = L_1 + L_2 + L_3 + L_4 \tag{1}$$

We start by handling R1: each faculty interviews similar amount of student. For each faculty in the current schedule, we construct a set that stores the students he/she interviews. We can then define a probability distribution using the mapping

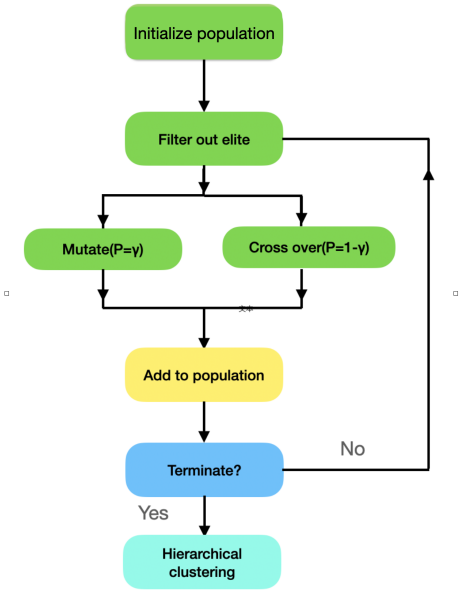

Figure 2: Conceptual Illustration of Our Model

between each faculty and the students he/she interviews,

$$P(i) = \frac{N(i)}{\sum_{j=1}^{M} N(j)} \tag{2}$$

where $N(i)$ is the number of students that the ith faculty interviews. We use clipped KL divergence between P and uniform distribution Q weighted by a factor of 10 to measure L_1

$$L_1 = 10 \cdot clip(KL(P||Q)) \tag{3}$$

where KL divergence is clipped to $[0, 1]$.

To measure L_2, we add a penalty of 1 million each time we find two interview groups with 4 identical interviewing faculty,

$$L_2 = \sum_{i=1}^{N} \sum_{j=0}^{i-1} I(i, j) \tag{4}$$

where $I(i, j)$ is defined as,

$$I(i,j) = \begin{cases} 1 \text{ million}, & \text{if } group(i) = group(j) \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

To measure L_3, we simply sum the overlap faculty count between any two interview groups,

$$L_3 = \sum_{i=1}^{N} \sum_{j=0}^{i-1} N(i, j) \tag{6}$$

where $N(i, j)$ is the overlap count of faculties between group i and j if $2 \leq N(i, j) \leq 3$.

To measure L_4, we make use of the mapping between faulty and students interviewed that we construct to measure L_1. We define L_4 as,

$$L_4 = 0.01 \cdot \sum_{i=1}^{M} \sum_{j=0}^{i-1} |S(i) \cap S(j)| \qquad (7)$$

where $S(i)$ denotes the set of students that the ith faculty interviews.

For each iteration, we filter out K elite schedules in the population with lowest loss. Mutation: for each elite schedule, we then define mutation and cross over to produce offspring schedules for the next iteration. For each interview group in a schedule, we randomly perturb one faculty with probability α, and ensure that all 4 faculties in that group are unique. In addition to the mutation operator, we define cross over operator on two randomly selected schedules. For each pair of interview groups in the two schedules, we randomly assign one faculty in group two to the corresponding faculty slot in group one with probability β, while ensuring that all faculties in each group are unique.

2.3 Algorithm

A detailed algorithm is attached. In each iteration, after filtering out elite schedules, we do the following operation, we apply mutation operator with probability γ or cross over operator with probability $1 - \gamma$ until population size reaches 100 which we set by default. In the end, we perform bottom-up hierarchical clustering to output a list of clusters stored in a csv file.

Algorithm 1 Genetic Algorithm based Scheduler

```
population ← list of size 100
initialize each schedule in the population
for i in 1...X do
    compute L for each schedule
    filter out K elite schedules with lowest L
    while population size < 100 do
        operator ← mutation p = γ
                   crossover p = 1 − γ
        new schedule ← apply operator
        add new schedule to population
    end while
    best schedule ← schedule with lowest loss
end for
clusters ← clustering on best schedule
return clusters
```

3 Experiments

3.1 Hyperparameter Tuning

There are 4 hyperparameters in our algorithm. K denotes the number of elite schedules we filter out in each iteration. γ denotes the probability of applying mutation operator. α denotes the probability of perturbing a faculty in each interview group in one schedule. β denotes the probability of crossing over each pair of interview groups between two schedules.

In this case, we don't have labeled data to tune these hyperparameters like a typical supervised machine learning setting. We apply random search to search for the best setting over 30 random sets of hyperparamters, which outperforms grid search strategy by effectively searching a larger configuration space (Bergstra and Bengio, 2012).

Due to the limited computing power, we run the 30 sets of hyperparameters on the fixed experiment setting: $N = 50$, $M = 15$, population size $= 100$, $X = 500$ iterations, and record the loss of the best schedule. A selected list of hyperparameters incuding the best setting is shown below.

K	γ	α	β	Loss
8	0.783	0.105	0.725	234.006
15	0.569	0.529	0.0623	233.973
17	0.559	0.416	0.445	275.970
19	0.575	0.998	0.812	288.022
16	**0.566**	**0.449**	**0.019**	**222.977**

Table 1: Hyperparamter Tuning Results (Partial)

3.2 Results

We plugin the best hyperparamter setting over the 30 sets, and run the algorithm for Q2 with setting $N = 379$, $M = 24$, $X = 500$ iterations. A complete visualization of clusters is available in the attached csv file. Since the result of our model is a list of clusters, we summarize some key characteristics into two graphs.

Figure 3 shows the number of students each faculty interviews. It indicates the result satisfies R1. Each faculty interviews around 60 to 70 students, which closely resembles a uniform distribution. Figure 4 shows the log loss of the best schedule in the population in each iteration. Recall that we add a loss of 1,000,000 as the penalty each time we find a violation of R2. The graph shows log of loss is below 13 ($2^{13} < 1,000,000$), which indi-

cates R2 is not violated in our final $N \times 4$ matrix. Further, the loss is converging to 7,255 as the algorithm approaching 500th iteration, which means R3 and R4 are also satisfied as much as possible.

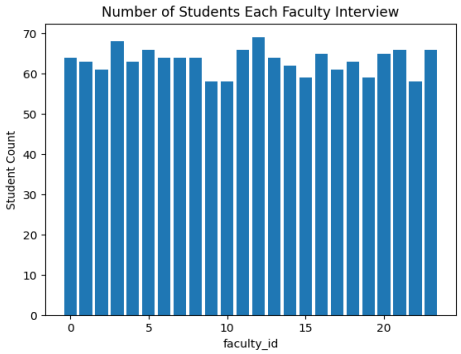

Figure 3: Number of Students Assigned to Each Faculty

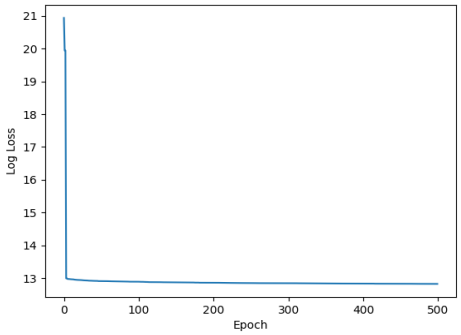

Figure 4: Log Loss versus Iteration

4 Discussion

Although we have successfully solved the problem, one modification could be made based on the existing combination factors of our model. We notice that although the number of students interviewed by each faculty and the repetition of faculties between groups are considered for maximum fairness, the difficulty of each faculty is not considered. Therefore, a difficulty parameter could be included to the model by adding a difficulty rating for each faculty. The rating is from 1 to 5, which represents the average of their historical grading (assuming each faculty gives a specific rating at the end of each interview) to interviewees, with 1 being the most relaxed grader and 5 being the most harsh grader. The interview groups should be having similar difficulty rating averaged from the four faculties within them. This difficulty rating could be easily incorporated into our loss function.

One clear defection of the genetic algorithm should be considered in future work. Genetic algorithms would be too computationally expensive with large N and M even for supercomputers. In this case, we believe the scheduling can be formulated as a reinforcement learning problem by using Q-Learning (Watkins and Dayan, 1992).

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r(s,a) + \gamma \max_{a} Q(s',a) - Q(s,a)] \tag{8}$$

We could define s as current schedule, $a \in A(s)$ as some perturbation for the current state, and $Q(s,a)$ as the sum of discounted future reward. Many work combing off-policy Q-Learning with neural network's strong representational power such as DQN (Mnih et al., 2013) and SAC (Haarnoja et al., 2018) could be applied in this case.

5 Conclusion

We implement genetic algorithm to solve the interview scheduling problem for NYU Shanghai. The algorithm self evolved and efficiently solved the optimized schedule of combinations of students and faculties subject to the 4 restrictions for the proposed NP-Hard problem, by looping selection, mutation, and crossover for the schedules that eventually converges to the result. This model correctly solves the $N = 379$ and $M = 24$ problem of Topic 3, and its empirical result from our experiment proves the applicability of our Genetic Algorithm based Scheduler. Future works are encouraged to modify our model with additional fairness parameters or with off-policy deep reinforcement learning algorithms.

Reproducibility

All code and data are available here: https://github.com/kaitrickster/Math-Modeling-Project

Collaboration Statement

All team members contributed equally in proposing and discussing possible solutions. Kai focused on implementing the model and running experiments. Haoyuan and Weixin focused on solving Q1 in the project. All members contributed equally in writing the paper.

Acknowledgments

This project has benefited from support to KL by Professor Keith Ross who sponsors his access to NYU Shanghai HPC in the Reinforcement Learning class.

References

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*.

Christopher J. C. H. Watkins and Peter Dayan. 1992. Q-learning. *Machine Learning*, 8(3):279–292.