‘I know it when I see it’ – Developing Quality Schedules Considering Subjective or Unspecified Criteria

Douglas Moody

Computer Systems Technology Department,
New York City College of Technology,
Brooklyn NY,11201, USA
dmoody@citytech.cuny.edu

Abstract. Most timetabling problems have a given objective function to measure the quality of a solution. However, users may have a “I know it when I see it” recognition of a quality schedule, without specifying the complete basis for their judgment. In this situation, the objective function cannot be exclusively used as a solution quality measurement. This work presents an AI based approach to aid in categorizing the solution’s quality when the users have not explicitly defined all factors used in their criteria.

Keywords: Timetabling, Artificial intelligence, Image classification

1 Introduction

This work examines when the user community is not able to precisely articulate their preferences, factors and criteria for evaluating the quality of a timetable solution. This may cause the true user’s quality measurement to vary from the objective function. Without the complete knowledge of the user community’s criteria, the scheduler is at a loss to improve the quality of the schedule. The user may employ a “I know it when I see it” approach [6] - where the user, without articulating specific criteria, judges the solution’s quality only after viewing.

This work focuses on a common timetabling problem - the Travelling Tournament Problem (TTP) defined in [1], to illustrate this possible scenario. This problem has been well studied and has a very precise and calculable objective function. We add new criteria to the objective function, though this criterion will be unknown to the timetabling algorithm. With use of AI, this work will still be able to recognize quality solutions considering the unknown criteria. The solution of the TTP has been addressed by a wide variety of integer programming, constraint programming, tiling and search techniques [7][4][3][2]. Our focus is not on improving the optimization of the distance objective function, but rather the incorporation of additional evaluation criteria. Others like Tuffaha et al. [11] have introduced known and measurable criteria to the TTP problem, such as season duration. This work looks to incorporate unknown and subjective criteria.

2 Challenge

Our scenario begins with the user community providing the teams and distances to enable the creation of a schedule according to the hard constraints of the TTP problem.

A scheduler uses software to create the schedule with a minimal total distance according to its algorithm. However, upon showing the schedule to the user community, the schedule receives a poor rating. The scheduler is not sure why, so additional schedules are generated for user review. These additional schedules have slightly higher distance values, but the perceived quality of solution by the user community varies, though no specific criticisms and feedback are provided to the scheduler.

For an example in a real-world scenario, let us assume that the user community would favor each team to have at least 3 "*long road trips*". We define a long road as a trip where the distance travelled is at least 80% of the longest trip taken by any team within the league. The reasoning for this user preference could be as follows:

– For a professional league, all teams should have to bear the burden of a long road trip (perhaps changing time zones) a few times to share the pain of the travel across the league.
– For a youth sports league, the road trips might represent exciting opportunities for hotel stays for the players. Each team should have a few fun weekends.

Hence a high-quality schedule is one that has relatively low total travel distance, but also provides that most teams have at least 3 long road trips. Our scenario is assuming the user community has never explicitly stated or even realizes this preference. This new criterion is known for our analyses and simulation in our project but is not known or used by algorithm evaluating or generating candidate schedules. Our approach consists of 3 major steps:

1) Convert the solution of the TTP algorithm to an image – Rendering the Schedule
2) Train the AI model to recognize good and poor schedules.
3) Generation and classification of candidate schedules.

We have chosen to work on the NFL32 instance of the TTP. This instance has sufficient teams to enable a schedule image to be rendered. The information is provided in a github repository of TTP instances and solutions on the Robin X website [12]. The algorithm selected is the NFL32Sol_ModifiedCircle solution [9]. The next sections describe each step.

3 Rendering the TTP Schedule as an Image

The output of a typical TTP solution is a file indicating the home team, away team and the slot (or week) of their game. The RobinX web site provides for the formats of TTP problems. This solution output must be converted to an image, where each pixel carries information about the solution. Since the TTP is a distance-oriented problem, we construct our grid will use colors to show the distance travelled each week. Our color will be a shade of a grayscale. Each color RGB pixel color will be one value providing example pixel colors of (1,1,1), (2,2,2), (50,50,50), (200,200,200), etc. The exception is those pixels close white (over 200) are colored yellow (255,255,0) to enable distinction with the white background. Pixel values are calculated based on the distance to the next game. The dark blocks indicate light or no travel for the team, while the brighter shades

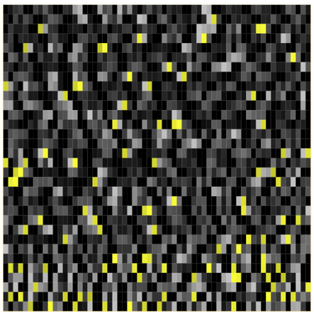


Fig. 1: A schedule image of the base solution for the project.

of gray and yellow (bright white in grayscale) indicate longer trips to the game location. The row order rendering of the TTP (each row is a particular team) is based upon the number of long trips within a season for the team. A sample TTP screen image output based on this coloration is shown in Fig. 1.

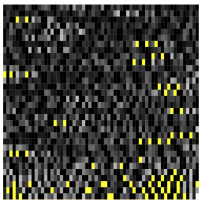
4 Training the AI Image Classification Model

We use the public and free image classification training platform provided by Google's Teachable [10]. The platform allows the AI user to create image via a web cam and mark their classification. We use a webcam to project the rendering of two sets of schedules – each referring to a good schedule or a poor schedule.

Our schedule generation process starts the NFL32Sol_ModifiedCirclec solution provided in the RobinX web site. This solution is based on relaxation algorithms and well-known techniques for creating tournament matchup combinations. [13][5]. It is a high-quality solution according to the TTP objective function. We then performed several perturbations on this solution. A perturbation was done by switching of the physical home locations of two teams. For example, switching between New York and Boston would require a team to travel its distance to Boston, though the schedule states the away team is New York. We utilized the pathway2code.com platform [8] for the calculation of the new schedules and the rendering of the schedule image. The project produced a series of 100 schedules for classification training purposes. An example input to the training model is shown in Fig. 2.

5 Image Classification and Results

After training, the project generated 2 sets of 25 schedule images for classification. The first set had only 2 perturbations each from the best schedule in RobinX, while the second set had 30 perturbations. Schedules from both sets were given to the AI tool for classification. The AI tool returned the probability that the schedule was a good or poor schedule.



Total Distance: 1178493
Distance Variance from Base: 0%
Category Classification: POOR
Actual embedded road trips:
Teams with < 3 Long Road Trips: 22
Teams with 3- 7 Long Road Trips: 7
Teams with > 8 Long Road Trips: 3
(not provided to training model)

Fig. 2: Base Solution Schedule Image and Long Road Trip Counts

The first set of 2 perturbations contained generated schedules that varied 10% from the best schedule. Nearly all the schedules involved more than half the teams with fewer than 3 road trips. The classifier was able to categorize these schedules as "poor" with a greater than 90% probability. The second set of generated schedules with 30 perturbations were on average only 10-25% higher in distance, however the perturbation led to more teams having longer road trips. The classification effort again was successful over 90% in labelling these schedules as good. Some schedules had nearly identical distances in the two groups, but a different distribution of long road trips and were classified correctly. Also, artificial colorings of schedules with extremely high or low coloring regions were classified successfully based on road trip distribution.

6 Summary

The work shows that a solution rendered as an image can used by a classifier to categorize the quality of the schedule after a training phase. The AI model is unaware of the actual criteria. This classifier will recognize patterns in the schedule based on pixel colorings, enabling the classifier to categorize the quality of the solution without criteria knowledge These criteria are not always available in practice, can be subjective, and vary among users in the community. For users who only know a high-quality schedule when they see it, we show quality schedules can still be generated.

References

1. Easton, K., Nemhauser, G., & Trick, M. (2001). The traveling tournament problem: description and benchmarks.
2. Easton, K., Nemhauser, G., & Trick, M. (2002, August). Solving the travelling tournament problem: A combined integer programming and constraint programming approach. In International Conference on the Practice and Theory of Automated Timetabling (pp. 100-109). Springer, Berlin, Heidelberg.
3. Fujiwara, N., Imahori, S., Matsui, T., & Miyashiro, R. (2006, August). Constructive algorithms for the constantdistance traveling tournament problem. In International Conference on the Practice and Theory of Automated Timetabling (pp. 135-146). Springer, Berlin, Heidelberg.

4. Goerigk, M., Hoshino, R., Kawarabayashi, K, & Westphal, S. (2014). Solving the Traveling Tournament Problemby Packing Three-Vertex Paths. Proceedings of the AAAI Conference on Artificial Intelligence, 28(1).
5. Hoshino, R. and Kawarabayashi, K. Generating approximate solutions to the traveling tournament problem using a linear distance relaxation J. Artificial Intelligence. Res., 2012, 45, 257-286.
6. Jacobellis v. Ohio, 378 U.S. 184 (1964)
7. Kim,J.,Han,J. & Jeong,S. ,Solving the traveling tournament problem based on the simulated annealing and Tabusearch algorithm, Journal of Engineering and Applied Sciences,13(21):9204-9212.
8. Pathway2Code Homepage, https://www.pathway2code.com, last accessed 2024/1/21.
9. RobinX web site https://github.com/Robin-X/RobinX/blob/master/Repository/TravelOptimization/Instances/NFL32.xml.
10. Teachable Machine Homepage, https://teachablemachine.withgoogle.com/v1/, last accessed 2024/3/12.
11. Tuffaha, T., Cavdarogu, B. & Atan T, (2021), Timetabling round robin tournaments with the consideration of rest durations – PATAT 2022 – Volume II 2021 p. 389-397.
12. Van Bulck, D., Goosens, D. Sc Schönberger, J., & Guajardo, M. (2020). Robinx: A three-field classification and unified data format for round-robin sportstimetabling. European Journal of Operational Research, 280 (2), 568–580.
13. Westphal, Stephan and Noparlik, Karl ,Miyashiro R., Matsui T. & Imahori, S. An approximation algorithm for the traveling tournament problem Ann. Oper. Res., 2012, 194, 317-324.