INITIAL PLAN

OPTIMISING FOR ENTERTAINMENT IN THE VOTE-REVEAL PROBLEM

CM3203 – One Semester Individual Project – 40 Credits

Author – Aric Fowler

Supervisor – Richard Booth

PROJECT OVERVIEW

DESCRIPTION

In many elections or competitions, a set of voters will rank a set of candidates from best to worst, or will give scores to some of the candidates, with the winner then being the candidate that gets the highest total number of points. When it comes to revealing the result after all votes have been cast, some competitions proceed by having a roll-call of all the voters in which each announces their own scores. This is often done for entertainment purposes (see, for example, the Eurovision Song Contest, in which each voting country announces their vote in turn). This raises the following question: which ordering of the voters should be chosen for the roll-call in order to maximise the entertainment value? For example, a roll-call for which the eventual winner becomes obvious very early on is probably not going to be very entertaining. There are 2 main parts to this project:

- Come up with some function(s) to measure the "entertainment value" of a roll-call.
- Solve the optimisation problem of finding a roll-call that maximises this function.

MOTIVATION

The vote-reveal problem is a combinatorial optimisation problem where the number of possible solutions is vast (but not infinite), and it is possible to evaluate the quality of a solution in polynomial time. It is not possible to find the globally optimal solution in polynomial time, however, and a brute force approach that considers all possible solutions would be infeasible.

In many optimisation problems, it is possible to create a well-defined cost function to be minimised or maximised. The travelling salesman problem is an example of this, where the cost function to be minimised is the sum of all edge weights in the solution. It is not as simple to define a cost function for the vote-reveal problem since the factor in question – entertainment value - is subjective. The need to find functions that can correctly model entertainment value adds an additional layer of complexity both to defining the problem and to solving it; the objective function needs to be simple enough to allow possible solutions to be evaluated efficiently, but complex enough to model the nuances of keeping an audience engaged.

AREAS OF FOCUS

CORE FEATURES

I have labelled the following goals as core features, which I deem to be essential parts of my project.

BASIC ENTERTAINMENT FUNCTION

My first task in this project will be to define a function that generalises some factor that contributes to an entertaining vote reveal. The earlier example mentioned the point at which the eventual winner becomes obvious; I aim to define a cost function that can estimate where this point lies for a given reveal order, so that by maximising the function the optimal reveal order will delay this for as long as possible. Another area I aim to investigate is the audience for the vote reveal. If most of the audience are supporters for specific candidates, it may be useful to prioritise these candidates in the cost function so that the vote reveal order is highly entertaining for those supporters, with a possible downside of being less entertaining to the rest of the audience.

Deliverable: A mathematical definition of the basic entertainment function on paper.

OBTAINING EXAMPLES

To test my program, I will need examples of candidates and voters with the associated voting data. I hope to have several examples with different types (e.g. a competition where each judge ranks all contestants versus a competition where each judge selects a single favourite) so that I can test the flexibility of my entertainment function. Manufactured data may be useful to test specific scenarios, although a mx of real and manufactured data is preferred.

Deliverable: Example data sets.

MODELLING AS A LINEAR PROGRAMMING PROBLEM

Once I have a well-defined function to optimise, I need to be able to set up and solve the problem to find a good solution. My goal is to represent the vote-reveal problem as a linear programming problem consisting of a set of variables that can be assigned values in some range, a set of linear constraints on these variables, and an objective function that must be maximised.

Deliverable: A set of constraints and objective function for each data set.

IMPLEMENTATION USING GUROBI

Gurobi is a powerful optimisation solver that can solve many types of optimisation problems including linear programming problems. It implements the branch and bound algorithm with additional features to improve performance such as pre-solving to reduce the problem complexity, parallelism, heuristics (to find good incumbents quickly), and cutting planes to trim fractional (infeasible) solutions out of the solution space. I will be able to quickly find good solutions to my linear programming problem by passing the constraints and objective function to Gurobi and analysing its output.

Deliverable: One or more optimal or near-optimal solutions for each problem.

ADDITIONAL FEATURES

After completing my core features, I would like to expand my project to investigate the entertainment function in more detail. I would also like to be able to give a comprehensive and meaningful analysis of the solutions that I find.

VISUALISING SOLUTIONS

It is difficult to analyse solutions by hand given only the ordering of votes and the value of the objective function. To facilitate the evaluation of my model I would like to develop a system that can visualise a given vote reveal order. An intuitive example would be a bar chart that shows the current score for each candidate with a button to reveal the next vote, updating the bars to reflect the new scores.

Deliverable: A tool that visualises vote reveal solutions.

EXTENDING THE ENTERTAINMENT MODEL

Another optional area that I aim to explore is the entertainment function. My basic function will consider the current state of entertainment based on the revealed votes; however, it is often the case that people speculate about results yet to come (e.g., sports tournaments where fans may consider some matches to be an easy win for their team while others have an unknown outcome). Time permitting, I would like to expand my function to consider the odds of each candidate winning overall in the hope that this can improve the entertainment value in cases where certain candidates are assumed to have an advantage. Consider a competition where an underdog gets more points overall than a favourite. In this case, it is possible to reveal a few votes for the underdog early on to establish a slight lead over the favourite. People may assume that the favourite has won more of the remaining votes and will eventually overtake, but if the underdog maintains their lead (or if they switch places several times) the audience will be more invested in finding out the eventual outcome.

Requirement: Extension to original base data to include odds of winning for each candidate

Deliverable: Entertainment function that takes odds into account.

WORK PLAN

This plan shows my intended workflow over the next 12 weeks with milestones to indicate points where major parts of the project should be completed. I have agreed with my supervisor to have a meeting every Friday to cover the progress made in that week, with a break between weeks 8 and 9 for easter.

Week	**Focus**	**Activities and Milestones**
1	**Initial Report**	Initial Report Setup work such as applying for Gurobi academic licence *Milestone: Initial Report is complete*
2	**Background**	Research on function(s) to model entertainment value Obtain example data
3	**Core features**	Model examples as linear programming problems with constraints derived from entertainment function Write software to solve these using Gurobi *Milestone: Optimal solutions have been found*
4		
5		
6	**Additional features**	Develop visualisation tool Extend entertainment function to incorporate additional features *Milestone: Visualisation of solutions*
7		
8		
Easter		(Buffer weeks to finish incomplete tasks)
Easter		
Easter		
9	**Evaluation**	Investigate solutions. Find evidence that functions are being modelled correctly.
10	**Report First Draft**	Compile information from dev log into first draft. Start writeup.
11	**Report Final Draft**	Completed report with all information included. Proofreading. *Milestone: Final report is complete*
12	**Submit Report**	