Planning Football Competitions

15-48 EC thesis proposal

For:	Master Students that know how to program and are interested. If you've ever done a course in heuristics, it is a pre.
Load:	15 - 48 EC, depending on blocks (see below)
Period (approx.):	From September 2021 onward
Supervisor:	Daan van den Berg
Contact:	daan@yamasan.nl, kawarimasen0010@gmail.com
Introduction video:	https://bit.ly/3DWGFrG

World class football involves planning. A lot of it.

About the project

The scientific term for a 'double round robin competition' in sports is the *traveling tournament problem* (TTP): every team plays every other team *twice*: once at home, and once away, with a limit on consecutive homes or aways. Planning such a competition can in some cases be tedious, especially when matches are played in short succession and across large distances. The problem of planning such a competition to minimize travel distances is very difficult, and can not be feasibly de optimally solved.

A way to get a good planning, is to use heuristic algorithms. These algorithms do not however, guarantee an optimal solution. Once such algorithm is Nicholaus Frohner's beam search, which he adapted with some gaussian noise (a random function). He claims the results are good. Let's see how good. Beam search has a few parameters, such as beam width, number of consecutive home and away games, and possibly some noisy parameters too. Furthermore, let's feed it some *real* data, such as a few champions league teams, Eredivisie, or Primera Division. Even though these teams usually return home after a match, testing an algorithm with real data teaches us a lot about performance.

Block #1
Replicate (part of) Frohner's work [1]. Make some publicly available data, and see what is a reasonable amount for with one block of 15-18 EC.

Block #2
Make more data. Larger tournaments, different parameterizations and see how the beam search performs over a *characterized* ensemble of traveling tournament problem instances. Are there differences depending on beam width and noise? Do some runtime analysis before you start, this could get messy. But we'll adjust to a reasonable workload, of course.

Block #3
There are many more heuristic algorithms, like the plant propagation algorithm [2-6] or simulated annealing [7], that can be parameterized for performance. Implement one or two of these algorithms, and let them solve the problem instances from Block#2. It would be great to do some parametrization, but to what extend this is feasible remains to be seen.

Block #4
Let's try to *evolve* really hard TTP-instances. Are hard instances for one algorithm also hard for another, or even other parametrizations?

References

[1]https://www.researchgate.net/publication/340534951_A_Beam_Search_Approach_to_the_Traveling_Tournament_Problem

[2] https://bit.ly/3jH1Eqe

[3] https://bit.ly/3r2fc1D

[4] https://bit.ly/36qLTMQ

[5] https://bit.ly/3e4b4Jg

[6] https://bit.ly/36qNXEr

[7] https://bit.ly/3yHtF5A