1. Module number	*SET11108/SET11508*
2. Module title	*Emergent Computing for Optimisation*
3. Module leader	*Emma Hart*
4. Tutor with responsibility for this Assessment Student's first point of contact	*As above*
5. Assessment	*Report*
6. Weighting	*60% of overall module total:*
7. Size and/or time limits for assessment	*Report – 6 pages*
8. Deadline of submission Your attention is drawn to the penalties for late submission	**Report submission (via Moodle) : Monday 2nd December 9am** **Demo: Thursday 28th November (during the pratical class 1pm-3pm)**
9. Arrangements for submission	Moodle (+demo in lab)

10. Assessment Regulations All assessments are subject to the University Regulations**.**	
11. The requirements for the assessment	*Please see attached document*
12. Special instructions	*See attached document*
13. Return of work	*within 3 weeks of submission.*
14. Assessment criteria	*See attached document* *Normal academic conventions for acknowledging sources should be followed.*

SET11108/SET11508 Coursework

Track cycling is raced on elliptical tracks called velodromes. These are usually 250 metres in circumference, which includes two straight sections and two banked turns. Track cycling comprises a number of sprint and track endurance events for both individuals and teams but here, we focus on a women's team event called the *Team Pursuit* - an event in which multiple cyclists work together to complete a given race distance in the minimum possible time. Team members take turns riding in the front position thus allowing the other team members to draft closely behind for maximum aerodynamic benefit. This rotation allows the *team* to maintain a significantly higher velocity than would be achievable by a single cyclist. Changes to the relative position of the cyclists are most efficiently performed on either of the two banked turns of the track, where the cyclist in first position sweeps up the bank of the turn and then rejoins in last position.

In the women's event, there are three cyclists in each team and the race is over twelve laps for a total of 3000 metres. All three cyclists in the team must complete the entire race distance, which typically leads to an even distribution of workload amongst the team.

There are four main ways by which to improve performance in this event. The first is to improve the physiological and psychological capabilities of the cyclists through training. The second is to improve the technical specifications of the bicycles used, in terms of their mass and aerodynamic properties*. The third is to change the pacing strategy used by the team such that the power applied results in maximal benefit. Finally, the transition strategy may be changed to achieve a more even distribution of energy usage by the team and the most aerodynamically effective velocity profile.*

In this coursework, you will design, implement and test an algorithm to try and improve the performance of a women's team of fixed composition over a particular race compared to a default strategy you are supplied with. You can optimize either the pacing strategy, the transition strategy or both components. You are given a software library that computes the time taken for the team to finish given a supplied pacing and transition strategy. The library simulates a race, using a model of the relevant physics of airflow, resistance etc.

A background document giving technical information about the software is supplied.

WHAT TO DO

The goal is to find a strategy that improves the race time for the given team with respect to a default supplied strategy. You can choose to do one of the following:

- Optimise only the *pacing strategy* assuming a fixed transition strategy

- Optimise only the *transition strategy*, assuming a fixed pacing strategy
- Optimise *both the pacing and transition strategy (either simultaneously or consecutively)*

Read the following very carefully:

1. You can be as creative as you wish – the aim is simply that you improve on the default strategy by as many seconds as possible.

2. You can use any optimization algorithm that you want to – you are not limited to those covered in class.

3. The simulator that returns the time for a race is written in Java. If you choose to write your algorithm in another language, then it is your responsibility to ensure that you convert the simulator correctly. All solutions will be tested in a demo using the Java program to ensure they are correct.

4. As this is a design problem, there is no limit to the time for which your algorithm can run or the number of runs you do in order to find the best possible solution. However, I expect to see evidence of using a search algorithm in line with the focus of the module – do not simply use the time between now and the hand-in to do an exhaustive search of the solution space!

5. You **cannot** change the characteristics of the riders. It is assumed that each rider in the team has the same profile and ability. These values are hard-coded in the program and should not be modified.

There are two parts to the hand-in: a **demo** and a **report**

Demo
At the demo, you will need to demonstrate your code, and you be able to explain what you did. Your final pacing and transition strategy will be tested outside of your program to verify the race time.
The demo does not carry a mark – however, if the demo is not completed satisfactorily, then no mark will be awarded for the report.

Report
You need to write a report describing what you did, and the results of your experiments. It should have a **maximum length of 6 pages, font size 11 minimum. Do not use an Appendix. Material on additional pages will NOT be marked.**

The report should cover:

1. **Introduction. (5 marks)**

Give a brief introduction that explains at a high level the approach you took to solving the problem – this section does not need to contain technical detail. It should explain what you decided to optimise and why.

It does NOT need to give a description of track-cycling or a general overview of what an evolutionary algorithm is.

2. **Method (20 marks)**
   This section should cover the technical details of your algorithm:
   - how you represented a solution to the problem
   - what fitness function was used by your algorithm
   - what algorithm you used to optimize the problem
   - any special operators you designed as part of the algorithm

   You should justify your choices in each case. Higher marks will be given for approaches that demonstrate clear insights into the problem in terms of what is optimised, and that use algorithms that are well adapted to the problem. Extra marks will be given where you refer to relevant literature to justify your choices.

3. **Experiments & Analysis (20 marks)**

   Describe any experiments that you undertook in order to investigate the performance of your algorithm and to tune any parameters. Be careful to give sufficient detail that another reader could reproduce your experiments. Use graphs or tables to summarise your results - do not simply give the result of every experiment you ran. Comment on any trends or observations.

4. **Conclusions (10 marks)**
   State the best race time that you achieved, identifying exactly what specification of your algorithm achieved this (i.e list the appropriate parameters).

   Comment on the success (or not) of your approach, reflecting on decision made such as the choice of algorithm etc, and noting any particularly interesting observations made during your experiments. Where possible, reflect on your results in the context of existing literature or in light of the course material.

5. **Future Work (5 marks)**
   Explain what you could do in the future to improve the algorithm. This might include taking a completely different approach, or modifying your current one.

Important detail

- **TOTAL MARKS: 60**

- **Report submission (via Moodle) : Monday 3rd December 9am**
- **Demo: Thursday 30th November**