Coursework 2019

Team Pursuit Cycling

How to win

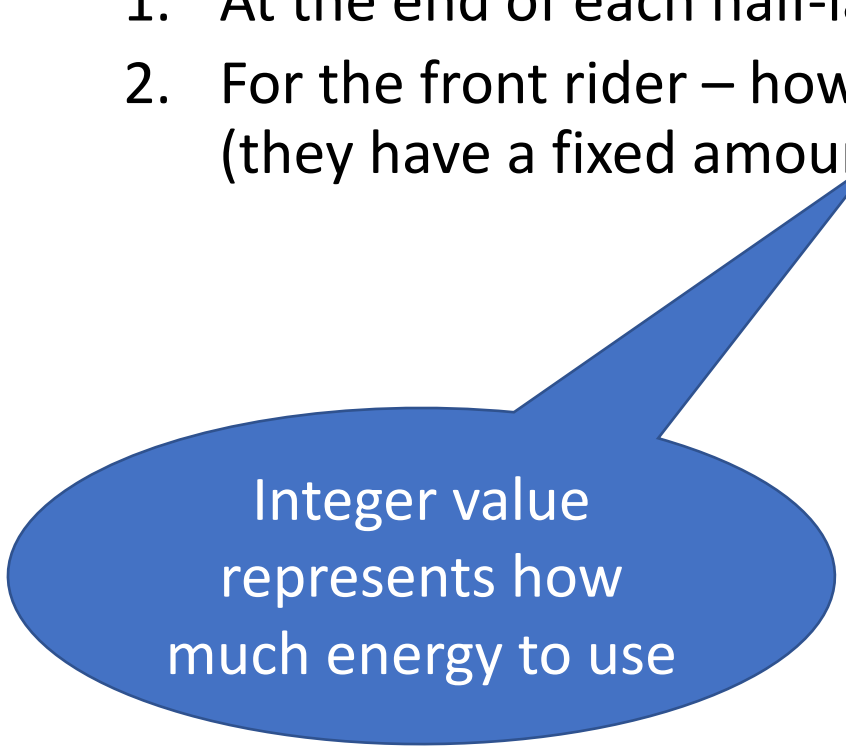
- Train well! Improve psychological and physiological performance
- Improve the specification of the bike (mass, aerodynamics)
- Optimise the transition strategy:
  - When to change the front rider
  - How fast the front rider cycles

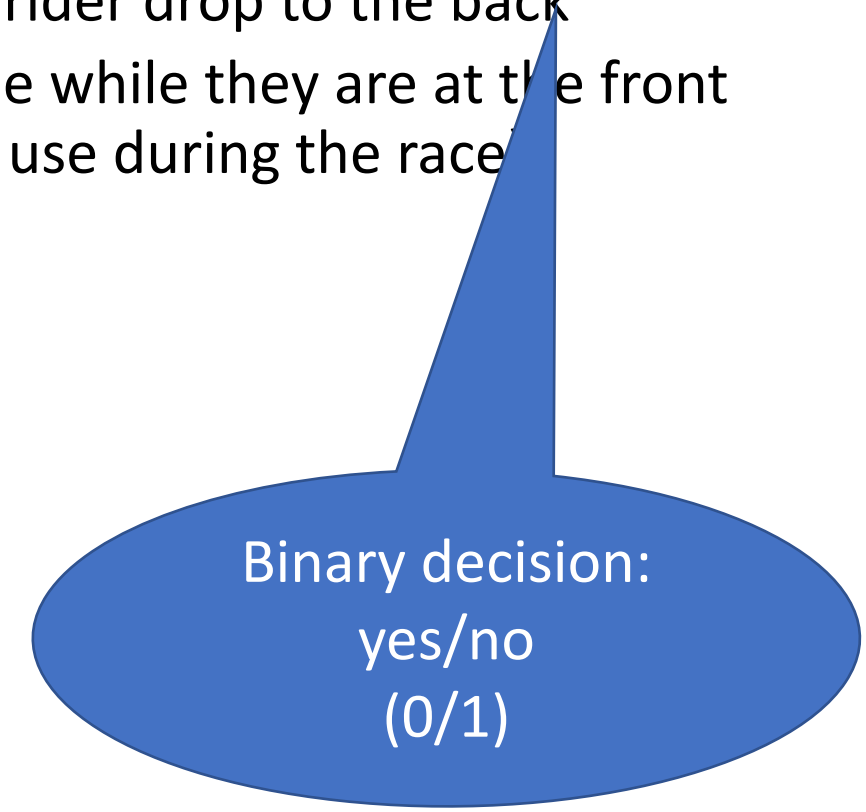
How to win

- ~~Train well! Improve psychological and physiological performance~~
- ~~Improve the specification of the bike (mass, aerodynamics)~~
- Optimise the transition strategy:
  - When to change the front rider
  - How fast the front rider cycles

Coursework….

- There are two things to optimise:
  1. At the end of each half-lap – should the front rider drop to the back
  2. For the front rider – how fast should they cycle while they are at the front (they have a fixed amount of energy they can use during the race)





Chromosome representation

1 0 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1

One bit for each possible transition (number of transitions is fixed to 22)

234 860 345 573 ……..884 204

One integer value representing the power during each of the 23 half-laps
Max power=1200
Min power =200

Two chromosomes:
Each has one value that refers to a half-lap

The half-laps are listed in order (e.g half-lap-first, half-lap2, …, half-lap-last)

Fitness Function

- This is a **minimisation** problem – you have to find the minimum time possible
- If the team runs out of energy, then the race stops and a time cannot be assigned
  - The default code returns a fitness of 1000 for **any** unfinished race (but you can modify this)
  - *But you can also get information from the system as to how close to the end of the race the team was*
- If the team finishes, the fitness is the time taken
  - The shorter the better!

What can you do?

- Optimise just the transition strategy and keep the pacing strategy fixed
- Optimise just the pacing strategy and keep the transition strategy fixed

- Optimise one strategy then the other (in an order you choose)
- Optimise both strategies simultaneously

How can you do it?

- *Tune the parameters of the algorithm supplied in a systematic manner without changing any code*
- Use another type of optimisation algorithm (Hill—climber, simulated-annealing)
  - Should be something that is inline with the module
  - If you want to use something not covered in class, please discuss with me first
- Modify the EA given in some way(s)
  - As per previous slide, decide what to optimise
  - Add new operators (crossover, mutation, selection)
  - Modify the fitness function
  - Change the representation:
    - However, you must convert it back into the format supplied before you call the simulator to get a fitness
    - The simulator expects to strings as shown on the previous slide
- Tune your new EA

What **can't** you do…

- Don't..change the number of half laps (the length of the race)
- Don't… change the min/max power
- Don't change the physics parameters (drag etc.)
- Don't change the parameters of the cyclists
- Don't change the parameters of the bicycles

Marking

- The marks are based on the report – not on how efficient/pretty your code is
- Marks for good ideas that reflect an understanding of EAs
  - Even if they don’t work well
- Marks for a methodical experimental design
- Marks for good presentation of results
- Marks for using statistics to evaluate your results