Daisy Intelligence 2019 Hackathon Problem Statement

Background

When it comes to racing, there are a lot of factors that need to be considered to get the best possible time. These factors include tire pressure, what speed to maintain, how hard to turn a corner, and even the exact line you should follow while racing. In the following problem, students will work with a simplified simulation of a race car that goes along a simple 1-dimensional track line. Students will need to figure out the best configuration for their car, as well as what actions to do while on the racetrack. They will test their configuration against multiple racetracks of varying complexity and will be scored using a weighted sum of track times.

Problem Statement

Teams will be provided with a set of racetracks in the form of csv files. The csv file represents a series of equidistant points on the racetrack, where each row is a specific point on the racetrack. These points are in order such that they would follow the direction of the track. For example, point 3 would be right before point 4 while driving across the racetrack, and be right after point 2.

Each point on the racetrack has a radius of curvature. This radius represents the curvature of the line and is the radius of a circle that perfectly fits the curvature. The closer the curve is to a straight line, the higher the radius, with a straight line being an infinite radius.

There are several constraints a solution must keep in order to be considered valid. The first and most important is curve speed. The radius, along with the specifications of the car, determine the maximum speed the car can traverse a curve. If the speed exceeds this amount, the car crashes and you have failed that track. The car must also manage its limited resources, mainly gas and tire wear. Whenever the car accelerates, it consumes gasoline, and whenever the car breaks, it wears out the tires. If the gas runs out, the car is no longer able to accelerate. If the tires wear out, you have failed that track.

The car can be customized in several ways. You will be given a certain point budget and will be able to spend this budget to alter different attributes of the car. These attributes include acceleration, braking, gas capacity, tire durability, top speed, and handling. The effect of these attributes increases linearly, but the cost increases at a faster rate, meaning a compromise needs to be made for different attributes.

Objective

The students are to submit a car configuration, along with a set of actions for each track provided. The actions detail what the car does at every point along the race track. The problem is described as an optimization problem, in which the students are to find the optimal solution to the problem (fastest possible time) in a reasonable amount of time (24h). Due to the complexity of the problem, it is unreasonable to simply try every single possible solution. As such, students are expected to utilize optimization algorithms and methods to find an optimal solution. The solution will be scored by simulating their actions and configuration on every track, and then taking a weighted sum of the track times.

Presentation

The teams with the top race times will be selected to present their solution and how they came up with it. As such, each team should also provide a presentation of their solution and method, as well as provide any source code used to generate their solution. Teams are expected to be able to present their solution and method if they are chosen as one of the top teams. The format of the presentation will be a power point of 6-7 slides. This power point will be included with the solution.

Submission

Teams will email their submission to hackathon@daisyintel.com This submission will include the names of the team members, an instruction csv file for each track, a car configuration file, a power point presentation, and the source code for the algorithm used to generate the solution. You do not need to include any libraries used. Please zip together your files. You have until 10:00am to submit and you can only submit once.

File Format

The format of the race track is a csv file. There is 1 file per track (named track_1.csv, track_2.csv, ...) The first row of the csv file are the column headers. The subsequent rows of the csv file consist of each point along the race track in sequential order. The distance between 2 points sequentially is 1 meter. The radius column contains the curvature of radius for that particular point. You will use this radius to calculate the maximum speed for that point. A radius of -1 represents infinity.

The format of your optimization consists of 1 file per race track, and 1 car configuration file. The race instruction file (named instruction_1.csv, instruction_2.csv, ...), will be of the same length (number of rows) as the respective race track file. The first row will contain column headers. The columns you should include are an acceleration column (called "a"), and a pit stop flag (called "pit_stop"). The acceleration column contains how the car accelerates at that particular point to the next point (between the min and max acceleration determined by car), while the pit stop flag is used to check if the car decides to take a pit stop (0 for no pit stop, 1 for pit stop). The race ends at the very last control point, so acceleration there doesn't matter. The car file (named car.csv) will contain 1 column row and 1 row filled with parameters. The column names should be "acceleration", "breaking", "speed", "gas", "tire", and "handling". The contents of each column should be the chosen tier for the respective attribute, a number between 1 and 5 (with 1 being the lowest tier and 5 being the highest tier). The car's total cost of attributes must not exceed 18.

Calculations

To calculate the velocity when the car travels from point x to point x+1 using acceleration a, use the following formula (set d to equal 1):

$$\mathrm{V}_{\mathrm{x}+1} = \sqrt{V_x^2 + 2a_x d}$$

If the radicand is negative, this means that the car is breaking so hard that it would never reach the second point. The acceleration at point x should be adjusted such that you reach the second point. In our scoring program, we automatically adjust the acceleration such that you reach the second point with 0 velocity ($V_{x+1}=0$)

To calculate the time it takes to travel from point x to point x+1, use one of the 2 following formulas (set d to equal 1):

If not accelerating/breaking:

$$\Delta\mathrm{t} = \frac{\mathrm{d}}{\mathrm{V}_{\mathrm{x}}}$$

If accelerating/breaking:

$$\Delta\mathrm{t} = \frac{\mathrm{V}_{\mathrm{x}+1} - \mathrm{V}_{\mathrm{x}}}{\mathrm{a}}$$

To calculate the maximum speed that you can drive between two control points, use the following formula

$$V_{max} = \sqrt{\frac{r_x H}{1000000}}$$

Where r is the radius of curvature, and H is the handling coefficient of the car. When using the radius of point x, both V_x and V_{x+1} must keep this constraint.

To calculate how much gas is consumed when accelerating, use the following formula

$$G = 0.1a^2$$

To calculate how much wear has been done to the tire while breaking (deaccelerating), use the following formula

$$W = 0.1a^2$$

Keep in mind that you only use gas while accelerating, and you only wear the tires while breaking. If you run out of gas you can no longer accelerate[1], where as if you wear out your tires you fail the race. If you run out of gas during the scoring simulation, you will no longer accelerate but you will still break as normal.

By using a pit stop, you can refill your gas and replace your tires. This is done by having the car stop (reach 0 velocity[2]) at the same control point as setting the pit stop to 1. Whenever you take a pitstop, you also take a 30 second penalty.

Below is a table containing the attributes and their costs. You will have (18) points.

Attribute	Tier 1	Tier 2	Tier 3	Tier 4	Tier 5
Acceleration	+10 m/s^2	+15 m/s^2	+20 m/s^2	+25 m/s^2	+30 m/s^2
Breaking	-10 m/s^2	-15 m/s^2	-20 m/s^2	-25 m/s^2	-30 m/s^2
Top Speed	10 m/s	20 m/s	30 m/s	40 m/s	50 m/s
Gas Cap.	500	750	1000	1250	1500
Tire Dur.	500	750	1000	1250	1500
Handling	9	12	15	18	21
Cost	**0**	**2**	**3**	**4**	**6**

[1] Note if you would run out of gas between 2 control points (i.e. current gas – gas consumed is negative), the simulation will be generous and give you enough gas to reach the next control point with 0 current gas.
[2] Note that when you decelerate to the point that you stop between two control points, the simulation automatically adjusts your acceleration to reach the control point at 0 velocity.

Material

You can find all the material for this hackathon in the following google drive.

https://drive.google.com/open?id=1WyiOKLbns1tbqE_DXZk33jocDIvSX8XF

It will contain this document, the track files, and a sample instruction and car file. Later on, a zip/tar will be provided that contains the scoring executable for testing purposes. There is a file for the windows executable and a file for the linux executable. Please unpack these archives and leave the executable in the directory.

Scoring

Your total score will be calculated using a weighted sum of each track time against the simulation program. The weighting will be as followed

Tracks 1, 7, and 8 are multiplied by 1.

Tracks 2, 3, and 4 are multiplied by 0.25.

Tracks 5 and 6 are multiplied by 0.5.

Testing

You will be provided with an executable that can be used to test the validity and score of your instructions/car. This will be provided later on during the hackathon. The executable can be ran as followed:

Windows:

fitness.exe {directory_with_instructions_and_car}

Linux

./fitness {directory_with_instructions_and_car}

Example:

fitness.exe test/

Where test/ contains car.csv, track_1.csv, track_2.csv, track_3.csv ...