F1 Strategist

Fernando Javier López Cerezo

December 14, 2021

You are the strategist of a Formula 1 racing team. You need to decide the strategy for an upcoming race. After talking to the data analysts and track engineers, you gather some important information about the behaviour of your team cars in the next circuit:

- The time to complete a lap is L/v, where L is the lap length and v is the average speed during that lap. The maximum speed attainable in each moment will depend on two factors: the weight of the car (which will decrease during the race due to fuel comsumption) and the degradation of the tyres (which will increase on each lap). Using the model built by the engineers, let $s(f, d)$ be the maximum speed at which a lap can be taken with f kg of fuel in the tank and the tyres having a degradation d (in %) at the beginning of said lap.
- The speed and weight of the car will determine fuel consumption. Let $g(v, f) \geqslant 0$ be the fuel consumption in a lap at speed v with f kg of fuel at the beginning of said lap.
- The tyre degradation depends on speed, weight, and current state of the tyres. let $h(v, f, d)$ be the degradation state of the tyres after a lap at speed v with f kg of fuel and tyres with degradation d at the beginning of said lap.

The scenario in which you are working is the following: the car has just completed the last pit stop and N laps are remaining. Therefore, you have to determine at which speed the car should be driven on each lap, so as to minimize the total time needed to complete the race. Note that at the beginning of this scenario the car has F kg of fuel left in the tanks, and the tyres are brand new ($d = 0$). Of course, if the car runs out of fuel ($f = 0$) or the tyres get completely degraded ($d = 100$) before the N laps are completed, the car will have to stop (i.e., you can assume $s(0, d) = s(f, 100) = 0$, for all f and d) and it will be a failure.

Provide a dynamic programming solution to this problem (you can assume that any variable you need for the state of the problem is an integer when expressed in suitable units). Upload a single pdf file describing each step in the resolution.

The solution to the problem is a list $V =< v_1, ..., v_N >$ where v_i is the speed at which the car should be driven on lap i. We are going to construct this list using the following method :

Multi-way decision - At which speed should the car go on the next lap?

Lets assume that starting with F kg of fuel and brand new tyres (d=0), $V^* =< v_1^*, ..., v_N^* >$ is the optimal solution to this problem. Then $v_i^* > 0 \quad \forall i \in \{1, .., N\}$. Otherwise the solution wouldn't be valid. Our objective function is to minimise the total time needed to complete the race, therefore $T^* = \sum_{i=1}^{N} \frac{L}{v_i^*}$ is optimal and therefore minimum. Now lets show that this problem fulfills the optimal substructure property by proving that given V^* then $V_{v_2^* -> v_N^*} =< v_2^*, ..., v_N^* >$ is the optimal solution for completing $N-1$ laps in the minimum time possible starting with $F - g(v_1^*, F)$ kg of fuel and a tyre degradation of $h(v_1^*, F, 0)$. We are going to prove this by contradiction.

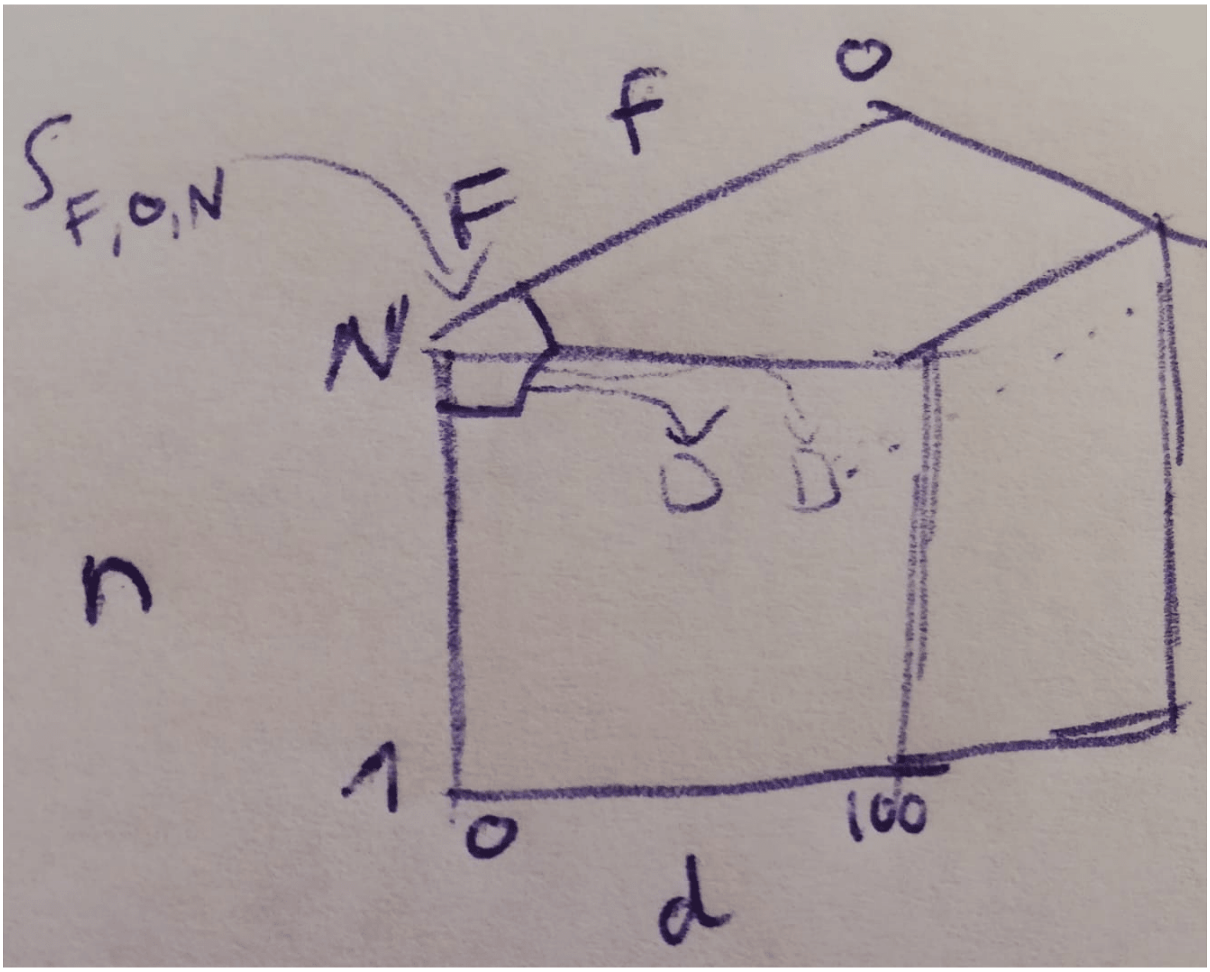
Let's assume that there exists a solution $V' =< v_2', ..., v_N' >$ which is more optimum than $V_{v_2^* -> v_N^*}$. In other words, $\sum_{i=2}^{N} \frac{L}{v_i'} < \sum_{i=2}^{N} \frac{L}{v_i^*} = T^* - \frac{L}{v_1^*}$. Therefore we could construct $V'' =< v_1^*, v_2', ..., v_N' >$. This would be a solution to our original problem such that $T'' = \frac{L}{v_1^*} + \sum_{i=2}^{N} \frac{L}{v_i'} < \frac{L}{v_1^*} + \sum_{i=2}^{N} \frac{L}{v_i^*} = T^*$. This is a contradiction since T^* is optimal. Therefore our problem has the optimal substructure property.

Let $S_{f,d,n}$ be the minimum time it takes to complete n laps starting with f kg of fuel in the tank and a tire degradation of d.

Lets consider the following Bellman equation :

$$S_{f,d,n} = \begin{cases} \infty & if \quad f = 0 \\ \infty & if \quad d = 100 \\ \frac{L}{s(f,d)} & if \quad n = 1 \\ \min\limits_{0<v\leq s(f,d)} \left(\frac{L}{v} + S_{f-g(v,f),h(v,f,d),n-1}\right) & if \quad otherwise \end{cases}$$

We can represent our solution using the following 3d space :



We can see that our solution has a space complexity of $O(fdn)$. The time complexity will depend on how many possible values there are for v between 0 and $s(f,d)$. Since we are not given any further information about this value we can only estimate the time complexity. Lets look at the complexity of finding $S_{f,d,n}$ given f, d and n. This complexity will be between O(1) (if there is only one possibility for the value of v) and $O(n^2)$ (since this is the maximum of possibilities for n-1 laps). Lets therefore estimate this complexity to be $O(n)$. Then the time complexity for our solution would be $O(fdn^2)$.

Algorithm 1 Calculate optimum time for F1 Strategist problem

Input: integers L, F, N
Output: optimum value $t(integer)$
Data: integers i, j, k, v, min, aux, 3d matrix V,S (integers) [0...F,0...100,1...N]
for $i \leftarrow 0$ to 100 **do**
    for $j \leftarrow 1$ to N **do**
        $S_{0,i,j} \leftarrow \infty$
    end for
end for
for $i \leftarrow 1$ to F **do**
    for $j \leftarrow 1$ to N **do**
        $S_{i,100,j} \leftarrow \infty$
    end for
end for
for $i \leftarrow 1$ to F **do**
    for $j \leftarrow 0$ to 99 **do**
        $S_{i,j,1} \leftarrow \frac{L}{s(i,j)}$
        $V_{i,j,1} \leftarrow s(i,j)$
    end for
end for
for $k \leftarrow 2$ to N **do**
    for $j \leftarrow 99$ to 0 **do**
        for $i \leftarrow 1$ to F **do**
            $\text{min} \leftarrow L + S_{i-g(1,i),h(1,i,j),k-1}$
            $V_{i,j,k} \leftarrow 1$
            for $v \leftarrow 2$ to $s(i,j)$ **do**
                $\text{aux} \leftarrow \frac{L}{v} + S_{i-g(v,i),h(v,i,j),k-1}$
                if $\text{aux} < \text{min}$ **then**
                    $\text{min} \leftarrow \text{aux}$
                    $V_{i,j,k} \leftarrow v$
                end if
            end for
            $S_{i,j,k} \leftarrow min$
        end for
    end for
end for
$t \leftarrow S_{F,0,N}$

Algorithm 2 Reconstruct solution of F1 Strategist problem

Input: integers$F, N,$ 3d matrix V (integers) [0...F,0...100,1...N]
Output: array A (integer)$[1, ..., N]$
Data: integers i, f, d
$f \leftarrow F$
$d \leftarrow 0$
for $i \leftarrow 1$ to N **do**
    $A_i \leftarrow V_{f,d,i}$
    $f \leftarrow f - g(S_i, f)$
    $d \leftarrow h(S_i, f, d)$
end for