The Journey of a Lost Mathematician in Search of Cheap Gas

Sam Florin

February 2019

1 The Setup

At the start of this $1,000$ week escapade, two factors must be chosen: the number of gas stations, G, and the cost of gas, p. The price of gas at each of these gas stations is chosen independently at random in $[p, 2p]$. Every week for $1,000$ weeks, he can visit any of these gas stations and buy a gallon of gas. Because these gas stations are in such high need for customers, at the end of each week, they will reduce their prices by \$1. These gas stations also have some nice customer perks. If you have built a streak with a gas station and have gone there for the last n weeks, the next week, in addition to the regular 1\$ discount, the gas station will reduce the price by an extra \$$n$. The catch is, if at the end of any week a gas station reduces their price below the cost of gas, p, they will realize they are no longer making a profit and double their asking price.

2 The Strategy

Our mathematician has to buy gas for $1,000$ weeks and wants to spend the least amount of money. At the beginning of this trip, he knows none of the prices at the stations but he does know G and p. He also knows the process the gas stations use to compute next week's prices and will therefore know any future prices at a gas station once he visits it once. Our strategy depends on a bound, b to be picked before hand. This essentially represents how much he is willing to spend before he searches an unknown station. After picking a random station to visit during the first week, his thought process works as follows:

1. Consider what the price would be next week for each of the stations he knows if he were to pick it this week.
2. Of the stations that would drop below p and therefore double were he to pick them, determine the one that is the cheapest this week.

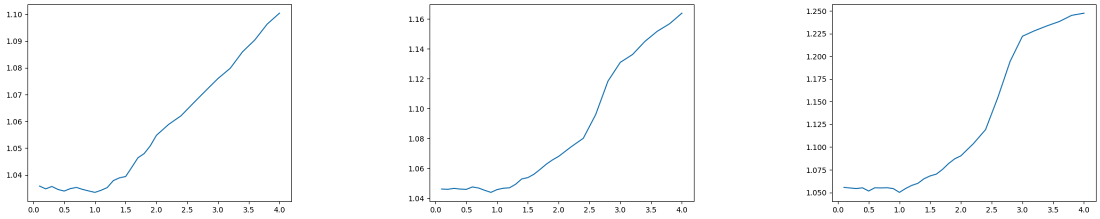
3. If the price at this station is $\leq bp$, the product of our bound and the cost of gas, pick it.
4. If the price is $> bp$, search a random unseen gas station.
5. If the price is $> bp$ but he has viewed all the gas stations already, pick the cheapest station.
6. If everything we've seen doubles, pick a new station.

3 Results

Although these rules are very simple and do not care much about streak or look ahead very far, they still perform quite well. In addition, interesting patterns emerge when looking at the performance of different bounds as the number of gas stations and the cost of gas varies. In analyzing the results of applying this strategy, some interesting patterns emerge. Throughout this section, we will score a bound b given a G and p by defining $score(G, p, b)$ as the average ratio of the amount he pays to the amount he would pay if he knew the price at every station to begin with and picked the cheapest each time. Also, we can define $cheap(G, p)$ as the minimal value of $score(G, p, b)$ as b varies from 1 to 2. In other words, $cheap(G, p) = \min\{score(G, p, b) | b \in [1, 2]\}$. We define $bestBound(G, p)$ to be x where $score(G, p, x) = cheap(G, p)$. So $bestBound$ is the bound our mathematician picks such that, on average, be pay the least.

3.1 Our Favorite Gas Cost

In graphing $cheap(G, p)$ for a fixed G and a varying p, an intriguing pattern occurs. The graphs below have, as their horizontal axis p/G and as their vertical axis $bestBound(G, p)$. It appears that when $p < G$, $bestBound(G, p)$ is consistently relatively low after which it increases quickly.



(a) $G = 50$

(b) $G = 100$

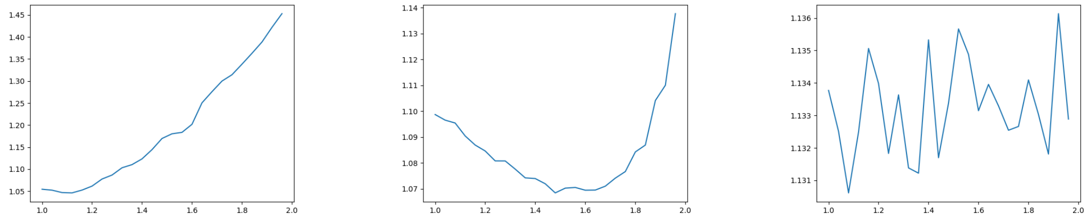
(c) $G = 200$

Figure 1: The comparison between p/G and $bestBound(G, p)$

I believe this occurs because our mathematician is more likely to go to the same relatively cheap gas station over and over again before it doubles, whereas, when the gas stations are few, he is able to search around more without getting too greedy. What this means for our mathematician is that if, after moving to this town, he sees that the cost of gas is less than the number of gas stations, he should rejoice because he can spend the least. But what bound should he pick?

3.2 Shifting Best Bounds

If we graph $score(G, p, b)$ with $b \in [1, 2]$, this image looks strictly increasing for small p meaning b works best when it is close to 1. However, as p increases, this graph starts to have a decline at the beginning to a minimum followed by an incline. The location of the minimum of this graph seems to increase as p increases until p is around $2.5G$ at which point chaos ensues.



(a) Smooth increasing at $p = 100$

(b) Nice decline at $p = 200$

(c) Absolute madness at $p = 300$

Figure 2: The results of bounds at different prices showing the effect of b on $score(100, p, b)$

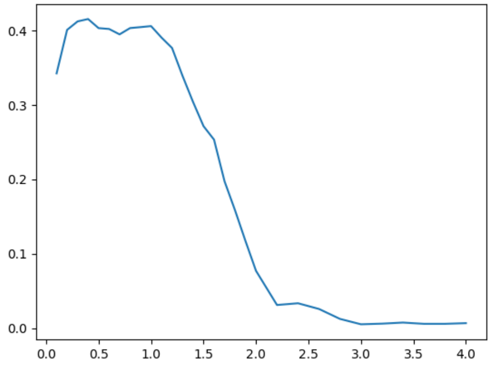
I made these graphs for 200 stations into a gif which I think better shows this process. That gif is at `https://gph.is/g/amzRJkZ`

It makes sense that, as prices increase, higher bounds are better because lower bounds lead to more searching for less overall reward. But where do these crazy graphs at higher p come from.

3.3 Ranges

The clear answer is that the ranges of the graphs drastically increase above a certain point. That can be seen in the graph below.

Figure 3: The range of possible values of $score(G, p, b)$. The horizontal axis represents G/p and the vertical axis is the the range of $score(G, p, b)$ as b varies



This essentially means that, once p exceeds a certain fraction of G, the bounds matter less and less. The reason for this is unclear to me. Perhaps it is because the decrease of prices becomes so irrelevant that these strategies fail to make a difference.

4 Variations

The first variation I considered was a simple one. Instead of the price at a station decreasing by an additional $\$n$ if he has gone there for the last n weeks, the price at that station will not decrease at all and instead increase by $\$n$, but all other rules stay the same. However, even in this new cruel world where companies will push our mathematician until he breaks, the patterns observed with the original case still held.

I was a bit more ambitious afterwards and generalized this process. A gas station game f can be defined by three functions: f_{picked}, $f_{notPicked}$, and $f_{notInBound}$. At the end of the week, f_{Picked} considers the price at the station that was picked as well as the streak there and returns what the the new price there will be. In the example we considered primarily, $f_{Picked}(p, s) = p-(s+1)$ and in the variation, $f_{Picked}(p, s) = p+s$. $f_{notPicked}$ takes in the same information but is applied to all stations that weren't picked. In both examples, $f_{notPicked}(p, s) = p - 1$. The final function, $f_{notInBound}$ is applied if the outputs of the previous functions are not in $[p, 2p]$. It takes in the new price and the cost of gas and outputs the adjusted price. In our two examples, $f_{notInBound}(x, g) = 2x$ if $x < g$ or x if $x > 2g$ because we have only cared if the price is too small.

The process for picking a gas stations is slightly different:

1. Consider what the price would be next week for each of the stations he knows if he were to

pick it this week.

2. Determine all stations such that $f_{notInBound}$ (if applied) does not increase the price there. Figure out the one of those that is the cheapest this week.
3. If the price at this station is $\leq bp$, the product of our bound and the cost of gas, pick it.
4. If the price is $> bp$, search a random unseen gas station.
5. If the price is $> bp$ but he has viewed all the gas stations already, pick the cheapest station.
6. If everything we've seen doubles, pick a new station.

Unfortunately, it doesn't appear that the patterns discovered earlier hold more generally, but it would be interesting to determine when they do. Alas, that is research for another day.

5 Some Code

Some of the not very nice source code is available on my Github here: `https://github.com/sflorin123/Sam-Florin-SPARC-Code`