

Grocery Gus or: How I Learned to Stop Worrying and Love the Log Card

Author: Gus Lipkin

Date: February 22, 2021

Table of Contents

Section
<u>Introduction and Background</u>
<u>Author's Note</u>
<u>My Origin Story</u>
<u>The Spinoff Series</u>
<u>The Dilemma</u>
<u>Analysis</u>
<u>Profit Maximization</u>
<u>Expected Marginal Utility</u>
<u>Conclusion</u>

Introduction and Background

Author's note

Defense contractors generally don't like when you go into too much detail about what you do or how their company works, even if it is for a paper for school. As such, I will be describing where I work and my job as a supermarket.

My Origin Story

I was originally hired as an intern with MiGrocery with the express purpose of collecting information on items that were delivered to be stocked in our canned goods department. Each canned good is supposed to have a log card with it. Each log card describes where the product was made and who handled it along with any climate conditions so we know if the product is safe for the sales floor. I was given a list of SKUs and a login for a piece of software we will call GrocerDocs which allows for realtime tracking of each SKU and a serial number for each unit of that SKU.

Unfortunately, the people who stock the warehouse and are not very organized and items of the same SKU do not always end up stocked together. In addition, just because GrocerDocs says an item is in a location does not mean it will actually be there when I look for it and each item may not have its required log card with it. Someone else could have taken the item without updating GrocerDocs or taken the log card and not uploaded it to GrocerDocs, or both.

GrocerDocs will tell me where each item is, but there's no way to check and see all of the locations I will need to visit without checking the location attribute for each SKU which could have several hundred serial numbers scattered about the warehouse where I work and on the sales floor that I do not have access to. I quickly realized that going SKU by SKU and searching all the different locations that for each specific serial number was inefficient. I found a way to export GrocerDocs' complete inventory and used R to extract a list of every serial number for every SKU I needed to find and order the list by location and export this information to a spreadsheet. I also added some columns to help me better keep track of product as it moved around the warehouse and so I knew which items I had already looked for and which ones I had uploaded. With my new program, I could now sync with GrocerDocs every morning to get the newest locations. Once I showed this to my supervisors, they thought I was a "tech wizard". Thus began the onslaught of spreadsheet generator requests.

The Spinoff Series

My supervisor, the canned food department head, wanted a spreadsheet that had the item names and SKUs down the left and the item serial numbers listed horizontally with each serial number cell highlighted based on its status. She had recently spent several days creating this spreadsheet but since item status can change, it was no longer up to date and she needed a new one. Creating it by hand would take me several days each time, but automating it would only take several days once. Once completed, she showed the other department managers and the bakery manager decided he wanted one too. Then, the district manager decided he wanted me to make a spreadsheet that was

very similar but only listed items on the sales floor. The catch was that he wanted a single file for each department with a worksheet for every store. While there is no list of what each store should have on the sales floor, I could make a good effort at figuring it out by comparing across stores and making sure every store has the same number. Unfortunately for him, it was my day off and I was not going to accept my regular hourly rate. Ultimately, we agreed on a bonus of five hours of pay for coming in and creating the document.

At this point, my supervisor, the canned goods department head, realized that I had not done any work for her in several weeks because I was working on generating spreadsheets of all sorts for other department heads. She wants me back on the log card project. By creating the location sorted spreadsheet of every item I need to find, I have already optimized the process as much as I can.

The Dilemma

Is it more economical to keep me on the log card project full time, keep me on the log card project and loan me out occasionally, or switch me to special projects full time?

Analysis

Expected Marginal Utility

As we will establish in [Profit Maximization](#), I provide no added value to the log card project over any other person performing the same task. I will concede, of course, that I already know which buttons to press to update my spreadsheets and no one else does, but those tasks are easy to teach in under fifteen minutes and can therefore be ignored. Because I provide no added value to the project, when my supervisor asks me to work on log cards the company has a marginal utility of zero while I am doing that work.

What is the company's expected marginal utility for each unit of time I am on a special project? This is relatively easy to answer because of the nature of the special projects I am assigned to. Each special project has progress that can be measured by tasks completed or percent completion when done by hand such as transcribing part numbers into a spreadsheet or trying to find duplicate part numbers. However, when writing a program that does these tasks it is exceptionally rare that a computer running the program would take as long as a human would to do the same task.

Assumption 1: A computer program will always take less time to complete a given task than a human will

Of course, that program does take time to write. It takes me about a week to write a fully-fledged solution to any special project. If I am able to re-purpose existing code for a new project, that time is cut down to just one or two days.

Assumption 2: I will take a week to finish writing a program to complete any given task

It should now be clear that if any special project is expected to take more than a week to complete, I should be tasked with writing a program to do the task. If the task is expected to be repeated, I should be asked to write a program if the total time spent on the task is more than a week. It may take me more time in the beginning, but at some point the programming time and task time will reach a breakeven point at which point the program will be more efficient.

These assumptions, however, assume that MiGrocery has only one person working on these projects, either me or someone else. I will always take a week to complete a given task, but should they choose to hire a small army, essentially mult-threading the process, the task could be completed in a matter of hours. This problem has a natural solution that the total amount paid to employees to complete the task should be minimized regardless of it is paying one person for many hours or many people for one hour. Alternatively, if a task is crucial to company operations, they could choose to pay more people to complete it more quickly rather than wait for me to construct a solution.

Profit Maximization

At first glance, this presents itself as a constrained optimization problem where the total payoff is a function of my time spent on the log card project, my time spent on special projects, the time it would take someone else to complete the same work, and the urgency of each project. Traditionally, we would need to account for differences in pay, but the problem is much simpler if everyone is paid the same. It is also very easy to add that element back in later as a function of time spent on the project.

We can use the standard π as profit. We can also define my time on log cards and special projects as M_L and M_S , respectively. The time someone else would take is O_L and O_S . The urgency of each project should be a ratio that sums to one which we can call W_L and W_S as we are weighing our options. The constant value of each finished project is defined as V_L and V_S . However, each project is not yet finished so D_L and D_S denote the percent progress of the project on a zero to one scale. Interns at MiGrocery are paid \$13 per hour. Thus, expected profit is written as

$$\pi = W_L V_L D_L + W_S V_S D_S - 13(M_L + O_L + M_S + O_S)$$

Quick pause here. I know the draft is super late but I haven't fully worked out how I'm doing this so I'm going to put some math and thoughts down here instead of trying to get words in which might be wrong. I know this will also change the math farther down the page.

- get in function of time to completion and NPV
 - $NPV = \sum_{t=1}^n \frac{R_t}{(1+i)^t}$
 - R_t cash flow during time t
 - See my point on the discount rate. But again, I'm not really sure how to apply that here
 - i discount rate
 - The time scale of these projects isn't really long enough for interest or discount rates to have an effect. But the project does become more useful and therefore have more

value the more complete it is. I'm not sure how I can apply that here.

- t time periods
 - The scale is in weeks
- Let's say time to completion is 10 weeks. Then I have $NPV = \sum_{t=1}^{10} \frac{R_t}{(1+i)^t}$. Couldn't I calculate the discount rate if I assume that the value of \$1 today is the same as \$1 10 weeks from now? Then I could pinpoint the discount rate for any given project.
 - If I know the average discount rate, could I then change cash flow from week to week to simulate how much value is added by completing a certain percentage of the project that week so long as the total cash flow sums to the same value in the end?
 - Would it be better to calculate future value and current value separately then subtract to get present value? I think this might be the easier way to do it. Would present value just be a flat number while future value uses the NPV equation?

Because the weight adds to one, the equation can be quickly rewritten as

$$\pi = W_L V_L D_L + (1 - W_L) V_S D_S - 13(M_L + O_L + M_S + O_S)$$

The portion completed of each project, D , can be re-written as a function of the total time taken to complete the project for each person and the total time it will take to complete the project. We can call this total time for each project T_L and T_S . I have already optimized the log card project so D_L becomes $\frac{M_L + O_L}{T_L}$. For any special projects, we can say that it will take me one-tenth the time it will take someone else. D_S can become $\frac{M_S}{.1T_S} + \frac{O_S}{T_S}$ which is simplified to $\frac{O_S + 10M_S}{T_S}$. The initial equation is then rewritten as

$$\pi = W_L V_L \frac{M_L + O_L}{T_L} + (1 - W_L) V_S \frac{O_S + 10M_S}{T_S} - 13(M_L + O_L + M_S + O_S)$$

Finally, in order to make this giant mess usable, T_L and T_S must be able to be said in terms of each other. Each special project takes about twenty-five hours to complete. Based on current progress, we can estimate that the log card project would take about ten twenty-five hour weeks to complete. Because I could complete ten special projects in the ten weeks it will take to complete the log card project, $T_L = 10T_S$. This only holds for the present moment in time because as time passes, the amount of special projects that can be completed by the time log cards are completed will decrease until the time remaining on the log card project will be equal to the amount of time it will take to complete a given special project. In order to make sure we are comparing the same amount of time, we must multiply any special project values by ten as well. This means that my and anyone else's time is limited to 250 hours as well. O_S and $10M_S$ become $250 - O_L$ and $10(250 - M_L)$, respectively. The profit equation becomes

$$\pi = W_L V_L \frac{M_L + O_L}{250} + 10(1 - W_L) V_S \frac{(250 - O_L) + 10(250 - M_L)}{250} - 13(M_L + O_L + 10(10(250 - M_L)) + 10(250 - O_L))$$

This cannot be simplified much more unless some of the variables are assigned numeric values. While I do not know the true values of W_L , V_L , or V_S , I can most certainly make something up. Let us say that the log cards are four times as important as any given special project and so $W_L = .8$. However, special projects are considerably more profitable and so $V_S = 5V_L$ and $V_S = \$10,000,000$.

$$\pi = (.8 * 2,000,000) \frac{M_L + O_L}{250} + 10(1 - .8)10,000,000 \frac{(250 - O_L) + 10(250 - M_L)}{250} - 13(M_L + O_L + 10(10(250 - M_L)) + 10(250 - O_L))$$

Simplifying,

$$\pi = 1,600,000 \frac{M_L + O_L}{250} + 20,000,000 \frac{(250 - O_L) + 10(250 - M_L)}{250} - 13(M_L + O_L + 10(10(250 - M_L)) + 10(250 - O_L))$$

$$\pi = 6400M_L + 6400O_L + 220000000 - 80000O_L - 800000M_L - 357500 - 1287M_L - 117O_L$$

$$\pi = 219642500 - 794887M_L - 73717O_L$$

The important part here is not the final profit equation because, as expected, the profit would be higher if MiGrocery did not have to pay employees, but rather that one hour of my time has a value of 794887 while the value of someone else's time is only 73717. My time is 10.87 times more valuable than someone else's for the same task if the constants given above retain their value.

Proper profit maximization requires taking the first derivative of the profit function. With all of the guesses about the values of constants I made and the ratios that allowed me to put one variable in terms of another, this quickly fell by the wayside. With proper numbers, the equation could be put into solver to give the amount of hours per project that myself and someone else should work on to maximize profit.

Conclusion

I may be tooting my own horn, so to speak, but I do not think that MiGrocery could have hired a better and more productive intern. I have shown time and time again that I am able to use skills in my toolset to expertly optimize current workloads and to automate tasks that could take weeks to complete by hand. Through profit maximization techniques with estimated numbers, I have shown that I can be more than ten times as productive as an employee doing the same task by hand. In discussing expected marginal utility, we learned that it makes sense for me to take on any special project that is expected to take more than a week to complete or that is an ongoing project. Because I am so much more productive on special projects, it makes sense for me to be switched to those full time rather than kept on log cards and loaned out to special projects on occasion.

Notes

- OR get in function of time to completion and NPV

- $NPV = \sum_{t=1}^n \frac{R_t}{(1+i)^t}$

- R_t cash flow during time t
 - i discount rate
 - t time periods
- time rate of value matters
 - w = function of time to completion (raised to a negative)
- v is amount of time put into it
- Gus has limited time per week but morons aren't limited
- Make w a function of time to completion and v is a function increasing in my time (2nd derivative is negative), then an internal solution should present
- Get in time to completion in a concrete way
- Explain T
- NPV changes over time when doing things by hand
 - O_s and M_s are a sum
- value moving forward is function of my time put into T
 - M_s asymptotes off to 1
- Talk theory before math
- Can solve by continuous with internal minimum
- -