# Spoon Theory Algorithmic Optimization with Diminishing Marginal Returns

Emily Ng

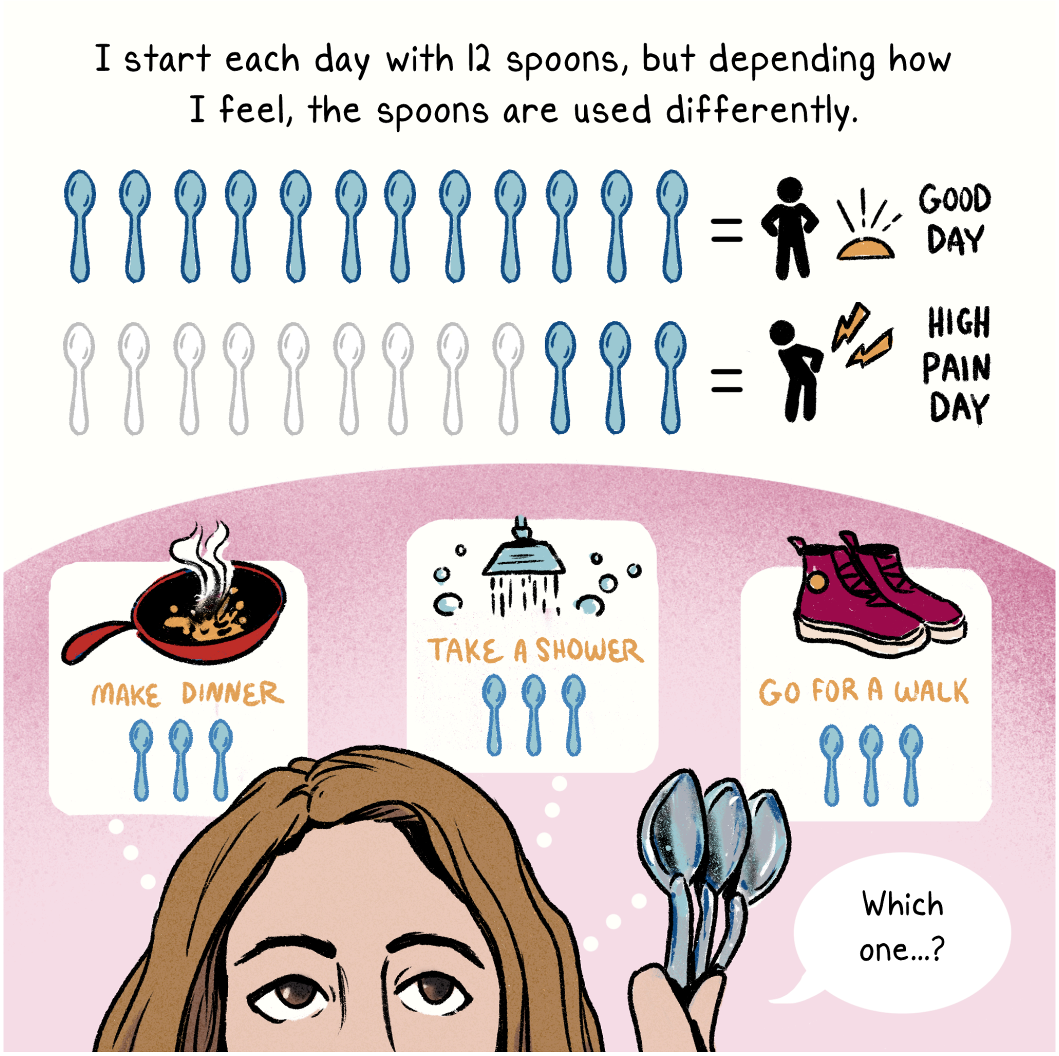
**TOC:**



## Introduction:

This project focuses on modeling diminishing energy efficiency over time, abstracting the real-world Spoon Theory into a computational optimization problem. Inspired by concepts of diminishing marginal returns in economics, the model simulates how task completion becomes increasingly costly as fatigue accumulates throughout the day. Neurotypical or able-bodied people are thought to have a flexible energy budget that can replenish and deplete in a given day. Spoon Theory, developed by Christine Miserandino in 2003, is used to describe the ability of people to complete tasks who are differently abled, chronically ill, or neurodivergent. The core idea is that people with these limitations have a limited amount of non-replenishing energy each day: 12 “spoons”. Each activity (getting dressed, cooking, working, socializing) costs a certain number of spoons. Once spoons are used up, they are gone. Extending past the 12 allotted spoons can lead to burnout, flare-ups in symptoms, or pain. These groups of people can’t “borrow” energy without serious consequences. The objective of Spoon Theory is to maximize the number of spoons used in a day. **NOTE: There were findings presented in class based on maximizing the number of tasks completed rather than the maximum energy. These findings are different as the code base has been altered to use maximum energy rather than complete maximum number of tasks.**

Figure Introduction:




We seek to provide algorithms for the spoon optimization problem from two algorithmic approaches: Knapsack and Greedy methodologies. We generate a random dataset (described in subsection The Data Set). Then, we compare two algorithmic approaches, Greedy and Knapsack, and analyze the following deliverables.

- Deliverable 1: The relationship between the number of tasks in a day and the proportion of completion with respect to maximum energy (6 spoons complete / 12 total spoons = 0.5).
- Deliverable 2: The relationship between elapsed time and the completed proportion of task energy with respect to maximum energy
- Deliverable 3: Average time to complete a “day” versus number of tasks
- Deliverable 4: Average ending proportion of energy in total completed versus alpha (the parameter controlling energy decay described in The Model)

**The Model:**

We utilize the parameter α, with respect to time, to model energy decay. $\alpha \in [0.05,\ 1]$ where a lower value represents faster decay rates and a value of 1 indicates no decay of energy. That is, no decay means energy expenditure remains constant throughout the day. Energy depletes in the model by $E(t + 1) = (E(t) - \text{energy cost of task})\,\alpha^{t}$, where t is the current timestep in the given day per algorithm, and E(t) is the energy available to you at timestep t. The decay function is intended to capture the compounding effect of fatigue, simulating how a task later in the day may disproportionately consume more energy.

The goal is to understand how to maximize total energy expenditure under fixed-budget, depleting resource systems. For our purposes, we assume a 12-unit energy allocation with a 12-hour day. This duration and energy expenditure is fixed to model the needs of differently-abled, chronically ill, or neurodivergent peoples.

**The Data Set:**

Choosing integers from [1,12], we choose the number of tasks (N) that a person has each day and 12 hours to complete these tasks in. For each value of N, we generate 1,000 samples. For each task, we independently assign using uniform random distributions:

- Energy Requirement $\epsilon$ [1, 12] discrete, integer-valued
- Time Duration $\epsilon$ [0.0001, 12] continuously
- Priority level $\epsilon$ [1, 3] discrete, integer-valued

We randomize the energy allocation and time requirement as well as a random priority on [1,3] where 1 is the most important and 3 is the least important. The addition of priority allows for variability in the model. Without it, both the Knapsack and Greedy approaches would tend to favor tasks requiring the highest energy and lowest amount of time, as these would appear most 'valuable' by default. Factoring in priority, now there is the chance of variability where lower energy or tasks with a lower energy-to-time ratio could be accomplished or may take precedence over higher energy expenditure tasks, adding realism to scheduling complexity dynamics. It is important to note that these variables are chosen independently. Therefore, more time does not imply more energy, nor does more time imply higher priority, etc. The result is a list of tuples that contain randomly and independently chosen energy allotment, time requirement, and priority.

**The Algorithms:**

**Knapsack Approach:**

We model the problem as a graph traversal where each node represents a state defined by the current task index, remaining energy, remaining time, and elapsed time. At each point, two options are explored:

1. Skip the current task and move to the next
2. Attempt to use the current task, depleting remaining energy by applying the decay function

We implement this as a depth-first search (DFS) over the task sample space (all possible orderings of tasks in the given 12-hour day), where the algorithm recursively explores all possible task sequences from the current state of choosing a given task. Prior subproblems are stored in a memoization dictionary.

**In each recursive step, the algorithm:**

1. Evaluates if it has reached the stopping criterion or has already solved the subproblem

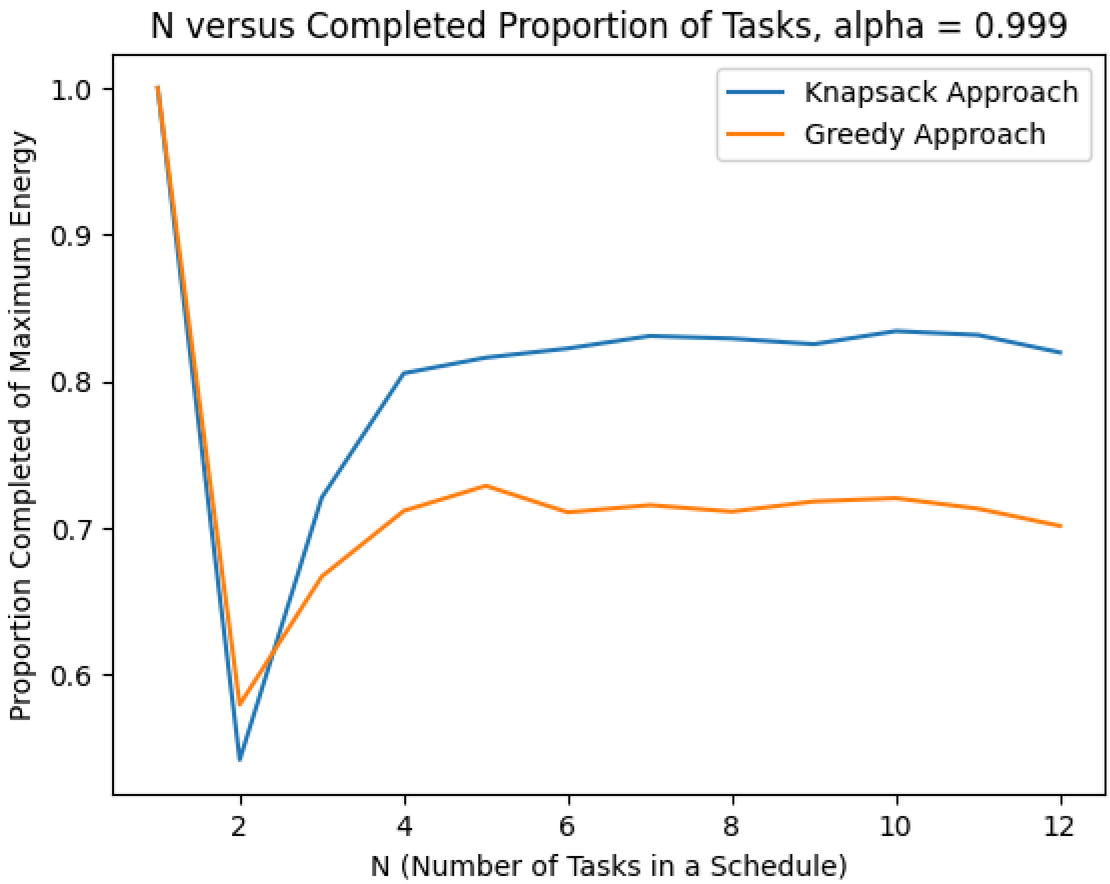
2. Evaluates the best possible task sequence if the current task is skipped
3. Compares it to the best sequence if the current task is taken, factoring in
    a. Immediate cost in energy and time
    b. Diminishing energy capacity due to time elapsed
    c. Maximizing the total energy used in a given timestep

The agent always selects the path that results in the highest number of completed tasks. Contrary to a classical Knapsack problem, the algorithm applies a graph-style exhaustive search over possible orderings of tasks. Each task can be thought of as a node or a state on a graph with state properties of the task index, remaining energy, remaining time, and elapsed time. We build a path to connect as many nodes as possible within the allotted timeframe constraint to utilize as much of the maximum energy as possible. This approach favors sequences that maximize task count completed by favoring high-energy or high-value tasks over a 4D space: task, energy, remaining time, and current time. Evident by the algorithmic constraints, there exists a dependency between remaining time and current time that could lead to reaching a suboptimal solution. In classical Knapsack problems, independence is necessary for reaching optimality of maximum payoff at every subproblem.
The algorithm is accordingly O(len(tasks) * maximum energy * maximum time * maximum time).

**Greedy Approach:**
In the Greedy algorithm, tasks are simply chosen based on the highest priority (1 being highest, 3 being lowest) and the highest proportion of energy required / time to complete the given task. Including priority allows for variation in lower energy tasks taking precedence over higher energy tasks. While energy decays using the same function as in the Knapsack problem, the approach is to simply accomplish as many tasks as possible in the remaining time, given the remaining energy. This algorithm is driven by the sorting on the two keys of priority and energy level/time required; therefore, the algorithm runs in O(nlogn), where n is the number of tasks, and O(1) space.

## Deliverable 1: The relationship between the number of tasks in a day (N) and the proportion of completion with respect to maximum energy (6 spoons complete / 12 total spoons = 0.5).

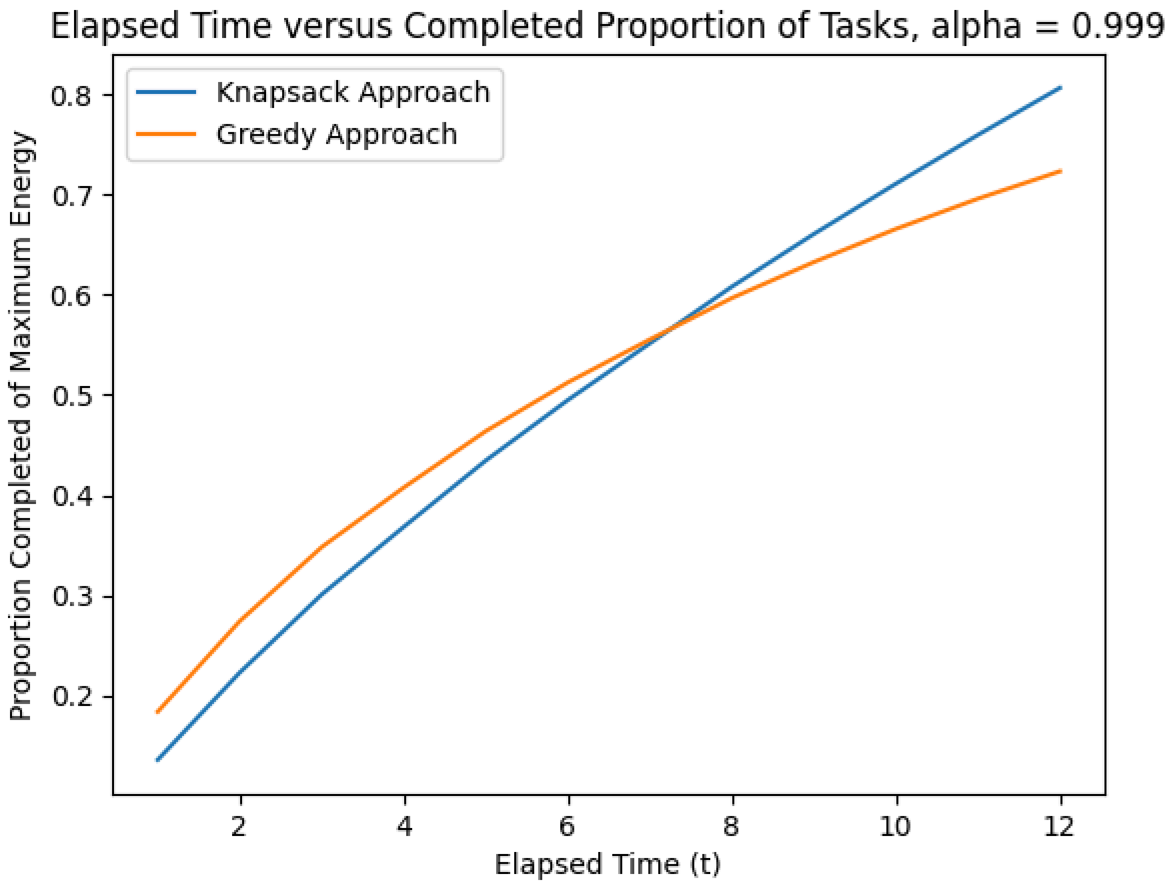
Figure 1:




The results for N = 1 are arguably throw-away values, as it is a given that if you only have one task to complete in a day, energy decreasing with respect to time would still allow for the single task to be accomplished without significant inability to not accomplish the task due to energy decay. At N=2, there is a sharp decline as the time decay takes effect. However, as more tasks are added to the schedule, both algorithms perform better. This could be due to the fact that as the number of tasks increases, the individual energy requirements of each task approach 1 given the fixed energy allocation of 12 in a given day which allows for a more steady rate of energy depletion while accomplishing tasks. The Knapsack approach hovers between 0.8-0.85 of maximum energy being used, and the Greedy approach is between 0.7-0.75. This indicates that there is an advantage to the Knapsack approach over the Greedy approach in the ability to use maximum energy most of the time. When Greedy is at a slight advantage at N=2, this can be attributed to the dependencies in the Knapsack approach, causing a suboptimal output solution.

## Deliverable 2: The relationship between elapsed time and the completed proportion of tasks

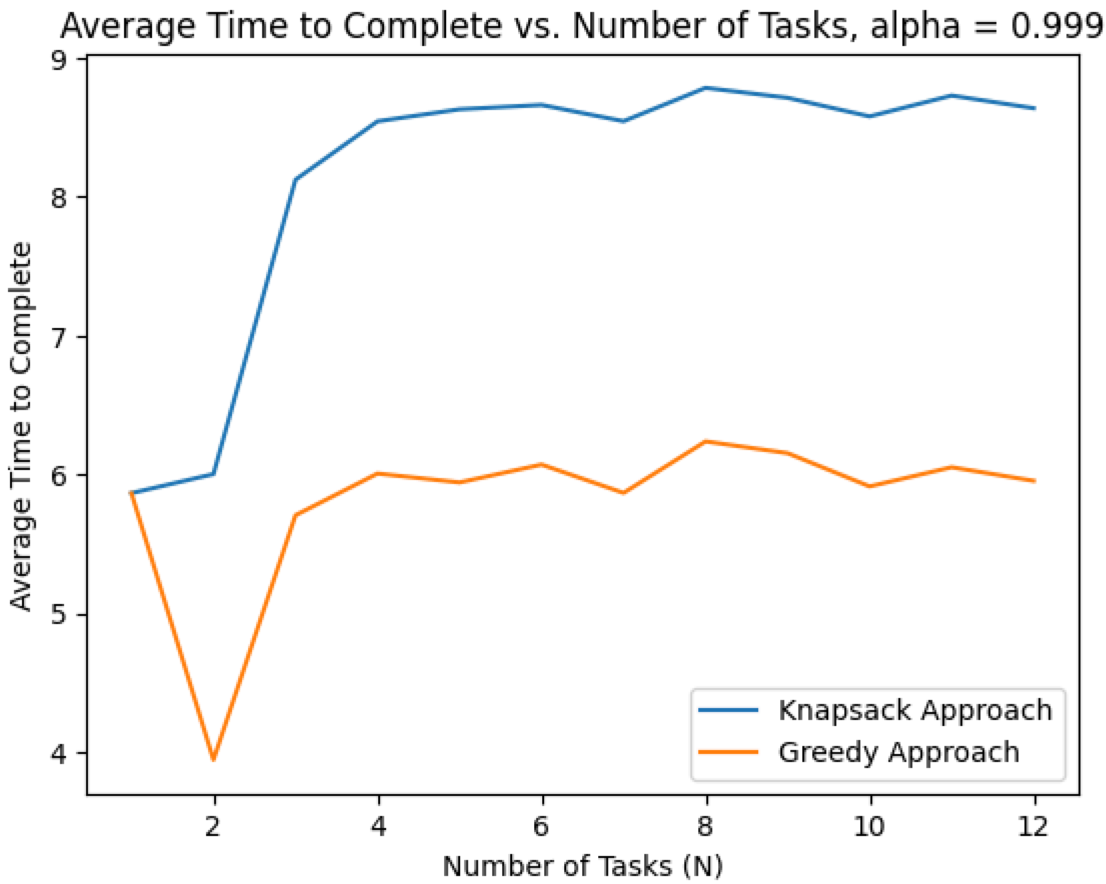
Figure 2:




We simply see that with both approaches, the decay rate of energy used with respect to time influences the ability to accomplish all tasks in a given schedule as the day gets longer. Our decay function is working! Before t = 8, there is an advantage to the Greedy approach over the Knapsack approach. I believe this finding can be attributed to the variation of ordering added by factoring in priority into the calculations, as well as the dependencies in regard to time of the Knapsack approach. Considering priority, there can be more variety added between choosing lower-value tasks over higher-value tasks, which could allow us to proportionally tackle more tasks as time goes on. However, at t = 8 and higher, using the Knapsack algorithm leads to increased ability to use the most energy in a given day, marginally.

## Deliverable 3: Average time to complete a “day” versus number of tasks

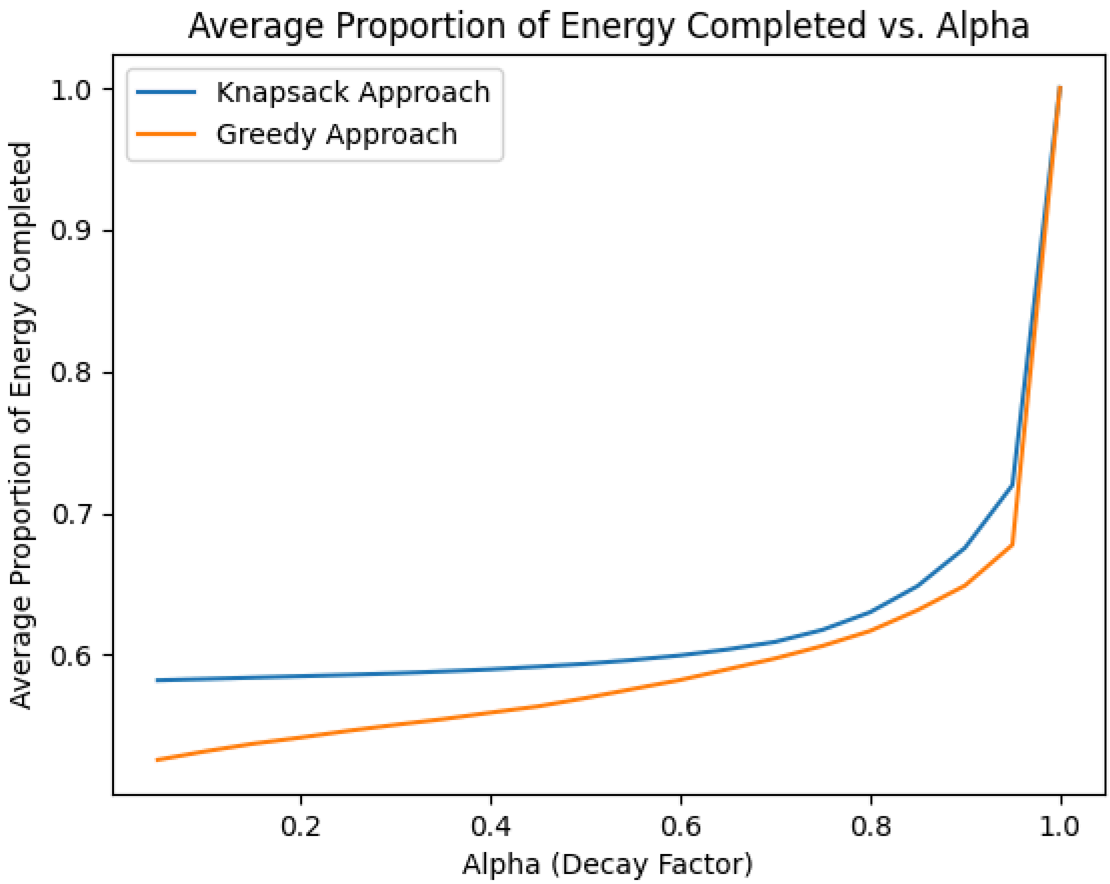
Figure 3:




Again, the results at N = 1 are uninteresting and are reflective of the randomly distributed choice of time on the interval [0.0001, 12]. However, we see that the average amount of time it takes to use the most energy possible in a given schedule is significantly higher in the Knapsack approach than in the Greedy approach. I think this finding can be attributed to what we saw in Figure 1 that more energy is being used; hence, it could imply that we are “active” in completing our tasks for longer. A more thorough peek under the hood could reveal definitive results in regard to the reason for this. But my guess would be simply on the principle of maximizing the most energy used, which can lead to completing the tasks in a given schedule taking longer.

## Deliverable 4: Average ending proportion of energy in total completed versus alpha (the parameter controlling energy decay described in The Model)

Figure 4:




This is, arguably, the most interesting finding of the paper. It shows that even if energy diminishes and does not replenish with regard to time, even by the slightest margins, it is not possible to use all of your maximum energy allotment on a given day. If energy is truly fixed and non-replenishing, this plot could suggest that the model of Spoon Theory and using all 12 spoons is not necessarily an accurate way to model how much you can complete in a day. Perhaps, trying to measure life in these full 12-spoon increments could lead to burnout. And, possibly, the way we are asking people to maximize their energy expenditures is flawed. Computational modeling of this phenomenon could be an interesting area for further study.

## Conclusion:

This paper found that, in regard to maximizing energy usage, a Knapsack approach resulted in more energy being used, more time of activity, and an increased amount of energy used as time went on in a 12-hour day with a 12-unit energy allotment. There was a slight advantage to a Greedy approach in a low number of tasks, $N < 3$, and a low number of active hours in a day, $t < 8$. However, the most interesting finding is that energy decays, even slightly, so that not all the energy of all 12 spoons can be utilized in a given day.

Spoon theory has its criticisms for being a one-size-fits-all approach, and users not being able to show up with consistent energy. These concerns have often been attributed to the nature of being differently-abled, neurodivergent, or chronically ill. But, the findings of this study could suggest that the issue could be misplaced onto the debilitating condition. The model they are taught to use their energy allocation could be flawed for being able to use their maximum energy in a day if these populations' energies are fixed and depleting over the course of a day.

Areas for further study include investigating seeking to complete the most tasks rather than using the most energy (as presented before, this results in the Greedy approach being favored). Also, attempting to approximate a better fitting model for maximizing energy allocation in fixed-resource environments with diminishing ability to accomplish tasks is a topic of interest to be explored. Further, the idea of energy "giving" tasks rather than energy "depleting" tasks is an area of future interest.

## Discussion:

The difficulty I found myself in, in modeling Spoon Theory, is that there is no clear objective. Get the most done you can, sure, but is that in regard to most tasks or maximizing your energy used in regard to productivity given some high level of importance or priority, etc. Since there is so much ambiguity, I can understand the criticism in regard to the tool, though many fiercely advocate for its efficiency and find it helpful.

It is important to note that the time dependencies of remaining time and elapsed time caused issues in the sub-problem optimality of maximizing used energy in the Knapsack approach. The reasonable assumptions I was given were quite muddy and did not maximize productivity in regards to accomplishing all tasks as anticipated. While the results were not presented in this paper, if the objective is maximizing the number of tasks completed rather than the maximum energy used, a Knapsack approach is suboptimal to Greedy in almost every regard, which could suggest that a Knapsack approach is not the right tool for the job given that objective.

## Attribution:

Google Gemini, Grammarly, and ChatGPT assisted in the writing of this project and its deliverables for content checking rather than generation. Dr. Brown was the most valuable and insightful form of consultation.

## Repository:

[https://github.com/emilylienhuang/SpoonTheory](https://github.com/emilylienhuang/SpoonTheory)