Analysis and Solution Of Markov Decision Problems With A Continuous, Stochastic State Component

Shruthi Sukumar
B.E., Anna University, 2014

A thesis submitted to the
Faculty of the Graduate School of the
University of Colorado for the degree of
Master of Science
Department of Electrical, Computer and Energy Engineering
2017

This thesis entitled:
Analysis and Solution Of Markov Decision Problems
With A Continuous, Stochastic State Component
written by Shruthi Sukumar
has been approved for the Department of Electrical, Computer and Energy Engineering

____
Jason R. Marden

____
Michael C. Mozer

Date ________________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

IRB protocol # 14-0750

Abstract

Sukumar, Shruthi (M.S., Electrical Engineering, Department of Electrical, Computer and Energy Engineering)

Analysis and Solution Of Markov Decision Problems With A Continuous, Stochastic State Component
Thesis directed by Associate Professor Jason R. Marden

Markov Decision Processes (MDPs) are discrete-time random processes that provide a framework to model sequential decision problems in stochastic environments. MDPs lend themselves to being solved easily using dynamic programming algorithms like value iteration. However, the use of MDPs to model real-world decision problems is restricted, as these algorithms rely on the countability and finiteness of the state space. Real-world decision problems often have continuous variables as part of their state space. Common approaches to extending the use of MDPs to solve these problems include discretization of the state space which, even in low-dimensional cases, suffers from problems like inefficiency and inaccuracy. In the absence of general approaches, methods have been proposed that rely on assumptions about the domain being modelled, like the form of the reward and transition functions. In this thesis, we focus on using MDPs to model sequential decision problems with continuous and discrete state variables and solve them using a modified value iteration approach. This is facilitated by assumptions about the domain, like the presence of piece-wise linear reward structure, binary action space, continuous transition function with infinite support and stochastic dynamics unaffected by the action sequence. Specifically, we solve an MDP used to model human behaviour in a specific task called delayed gratification, using an efficient value iteration based approach. This approach to modelling involves augmenting the state space of the decision-making system to reasonably reflect environmental and psychological factors, and explain the apparent irrationality in human behaviour. Results include the simulation outputs for the value function and comparison of our method to the naive discretization approach. The optimal policy is characterized by hazard curves which determine the probability of an event happening at a particular time, given that it hasn't happened up to that point. To validate our solution, we generate synthetic data and try to infer the parameters used in the generative model. Results for this inference are shown via fits to the synthetic hazard curves by the model's hazard function estimate. Further, to justify this modelling approach for human behaviour fits from the model output to empirical data are shown.

Acknowledgments

I am grateful to my thesis supervisor Dr. Michael Mozer for giving me the opportunity to work in his lab since Spring 2015. My interactions with him have taught me a great deal about how to do research and most of my work in this thesis and other projects have been mostly shaped by his advice. The work presented in this thesis was part of our paper titled "Overcoming Temptations: Incentive Design For Intertemporal Choice" in collaboration with Camden Elliott-Williams, Shabnam Hakimi and Adrian Ward. I am also thankful for the guidance from my thesis advisor Dr Jason Marden, whose dynamic programming course in Spring 2015 taught me the first thing I know about optimal control and gave me the background required for me to be part of this project. I extend my sincerest thanks to all my other professors at CU-Boulder who have have been extremely supportive anytime I needed any guidance. Dr Behrouz Touri took the time out to be part of my committee which is much appreciated.

I would be remiss not to thank my friends in Boulder, especially my lab colleagues Brett Roads, Mohammad Khajah and Shirly Montero and my good friend Santhanakrishnan Ramani for always giving me valuable feedback on the work I have presented in my thesis.

I would like to also thank my colleagues at Answeron Inc., especially Don Kainer, COO and Eric Johnson, CEO, for their always valuable advice and support, and also for giving me the opportunity to work there.

I would like to also thank my undergraduate advisor Dr P V Ramakrishna for providing me with the opportunity to work in the Integrated Systems Laboratory during my undergraduate degree. It is because of his advice and guidance that I first wanted to get involved in research.

Last but not the least, I am thankful to my mother Sreedaya Sukumar and my father Sukumar Seshadri for supporting me morally and financially for the last twenty-four years, and for supporting my every decision and never losing faith in me.

Contents

List of Tables

List of Figures

1 Introduction

Sequential decision problems are faced ubiquitously by intelligent systems that interact with their environments to optimize returns. In such decision problems, these systems need to carry out a sequence of actions that optimize not only their immediate rewards but their overall expected future payoffs. To optimize future payoffs, sufficient *planning* is required, by which the system is able to foresee the implications of current actions on not only immediate rewards but also future returns. Hereafter, we will use the term *agent* to refer to the systems interacting with their environment.

The study of sequential decision making is of interest in different areas of science from robotics to economics. With the advent of artificial intelligence and the rise in automation of various tasks, the study of optimal decision making by systems has become increasingly important. Decision theory is also studied widely among psychologists and behavioural economists to observe, rationalize and even optimize human behaviour by better mechanism design. In this thesis, we present a method that uses optimal control theory to model human behaviour, but can easily be extended to other domains.

There is a wide range of examples of sequential decision tasks that are faced in the real-world by any decision-making agent. One common example is the shortest path problem— what is the path that needs to be taken in order to incur minimum costs. The costs, in this case, could be time, distance, or anything else that depends on the agent and the environment. Another specific example is solving for the optimal strategy to play a game like Tetris. The agent that plays the game (Figure 1) must optimize their strategy to eliminate as many rows as possible to gain the maximum number of points. The agent could be a human or a computer that is learning to play the game. Later in the thesis, we will formalize our approach in the context of human decision tasks called *delayed gratification*, where the decision-maker is faced with the choice of waiting for a large reward for a period of time or choosing a small reward available immediately. This is an example of a sequential decision task as at every time step, the human agent needs to make a decision to either continue to the large reward or quit waiting in favour of the small reward.

In the above examples, it is entirely possible and is more often the case, that there is stochasticity in the environment. In Tetris, the stochasticity could present in the form of uncertainty about what piece will arrive next and in what orientation. Now, the agent is

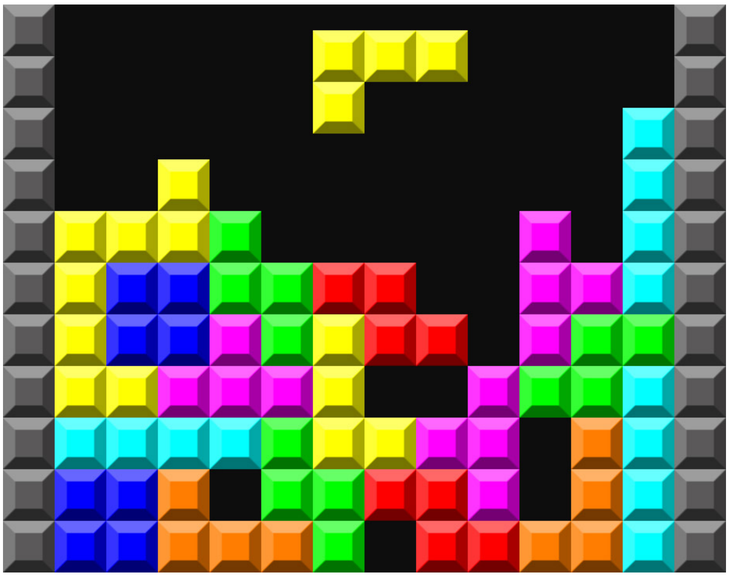


Figure 1: Tetris game board; Souce: internet

tasked with accounting for such uncertain events while computing the optimal strategy, based on their belief about the distribution of over the occurrence of these events. In the *delayed-gratification* case, the uncertainty arises in the form of moment-to-moment fluctuation in the agent's willpower, which is considered to be stochastic (e.g., fluctuations in the agent's level of hunger, exhaustion, the focus of attention), and needs to be accounted for while taking decisions. The modelling details and the reasoning behind them are presented in detail in the next chapter. Since most decision problems happen in uncertain environments, the agent is required to compute a *contigency plan*, which spells out the optimal policy to follow to maximize *expected* future rewards. This implies that the agent performs *closed-loop* control where at every step the agent receives feedback about the current state of the system to evaluate the best action to take that step.

Markov decision processes (MDPs) are random processes that are used to model such stochastic control systems. MDPs are a useful modelling framework that lends themselves to being easily solved using dynamic programming algorithms. One common algorithm we will focus on is *value iteration*. We describe MDPs as well as value iteration below.

1.1 Markov Decision Processes

As stated before, Markov Decision Processes provide a mathematical framework for the modeling of sequential decision tasks in stochastic environments. MDPs are completely defined by five elements of the decision making system [2]:

1. **State Space S:** The state space includes all the variables that are part of the state of the decision making system. Agents transition from one state to another upon taking a decision or applying a control signal. Often, the state space is augmented to include multiple variables which are ideally all discrete variables.

2. **Action Space A:** The action space includes the variable that indicates the decision or control signal of the agents.

3. **Transition function $P(S_{t+1}|S_t, A_t)$:** This function defines the probability of transitioning to a particular state, given the current state and action taken by the agent.

4. **Reward function $R(S_t, A_t)$:** The reward function determines the immediate reward associated with the current state and the action (the state-action pair).

5. **Discount factor γ:** This discount factor of the MDPs determine how much the rewards out in the future are discounted exponentially in time by the decision-making agent. Typically, $\gamma \in [0, 1)$, and is essential ensure tractability of the solution to the MDPs when the planning horizon presented is indefinite.

A problem modeled using the Markov Decision Process is called a Markov Decision *Problem.* In a Markov decision problem, the goal is to identify the optimal policy that is to be followed by a decision making agent, i.e., the strategy that maximizes the total expected reward the agent obtains. To model a decision problem as a Markov decision problem, it is important to identify all the analogues of the decision making system to the attributes described above. Once the problem is formalized using the MDP framework, it can then be solved using optimization techniques like dynamic programming [2]. We now describe a basic algorithm in dynamic programming and optimal control and how it used to solve MDPs.

The overall objective that is optimized by the decision-making agent is shown below:

$$V^* = \max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s, \pi(s))\right] \tag{1}$$

This represents expected future reward that needs to be maximized by the agent. The objective specified here is for the infinite-horizon case. In Equation (1), the objective is maximized over all policies; here π represents a policy and $\pi(s)$ represents the action carried out by the agent under the policy, at state s. V^* denotes the objective obtained under the optimal policy. To solve for the optimal objective in the infinite horizon case, we iteratively compute the value function, until it converges to the optimal value function using the following equation:

$$V_i(s) = \max_a \sum_{s'} P(s'|s,a)\Bigg[R(s,a) + \gamma V_{i-1}(s')\Bigg] \tag{2}$$

Each iteration, represented by Equation (2), computes the *value* of state s under the current estimate of the optimal policy. Once the algorithm converges, the value function is optimal and we have:

$$V_i(s) = V_{i-1}(s) = V^*(s) \quad \forall s \in S$$

When $V^*(s)$ is plugged into Equation (2), it becomes the *fixed point* of that equation, which is called the Bellman Equation (3):

$$V^*(s) = \max_a \sum_{s'} P(s'|s,a)\big[R(s,a) + \gamma V^*(s')\big] \tag{3}$$

$$\pi^*(s) = \arg\max_a \sum_{s'} P(s'|s,a)\big[R(s,a) + \gamma V^*(s')\big] \tag{4}$$

Here, π^* represents the optimal policy that is returned by value iteration. The Bellman Equation (3) is the fundamental equation in dynamic programming and is based on the Principle of Optimality which states [1]—"An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.". Essentially, the equation computes the optimal policy from every state until the end, despite sub-optimal starting conditions due to stochasticity of the system. Value iteration iteratively computes the value function under the optimal policy for all the states in the state space. The pseudocode for the algorithm is shown below:

```
Data:
P - transition function $P(s'|s,a)$
R - reward function $R(s,a)$
$\epsilon$ - threshold value $\epsilon > 0$
Result:
$V[s]$ - approximate optimal value function
$\pi[s]$ - approximate optimal policy function
initialization $V_0[s] \equiv 0$ and $k \leftarrow 1$ ;
while $|V_k(s) - V_{k-1}(s)| > \epsilon$ do
    for each state $s \in S$ do
        $V_k(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)\big\{R(s,a) + \gamma V_{k-1}(s')\big\}$ ;
        $\pi(s) \leftarrow \text{argmax}_a \sum_{s'} P(s'|s,a)\big\{R(s,a) + \gamma V_{k-1}(s')\big\}$ ;
    end
    $k \leftarrow k + 1$ ;
end
return $V_k, \pi$
```

Figure 2: General Value Iteration Algorithm

The parameter ϵ, known as the *Bellman residual*, is the maximum difference between any two successive computations of the value function. If the Bellman residual is ϵ at any particular iteration, then the current estimate of the value function is $\frac{2\epsilon\gamma}{1-\gamma}$ away from the optimal value function.

1.2 Literature Review

As stated earlier, modeling sequential decision problems with continuous state variables poses many challenges such as inaccurate solutions, inefficiency and, in problems high-dimensional state spaces, the *curse of dimensionality.* To accommodate continuous state and action variables, Equation (3) can be modified (replacing the summation with integrals), allowing a larger class of models to be modelled. In this section of the thesis, we present an overview of the previous work on efficiently solving MDPs with continuous state variables. Many of the papers reviewed in this section make assumptions about the domain being modeled and present algorithms that are more efficient than some of the general solutions.

Li and Littman [9] present a new algorithm called *Lazy Approximation*, that solves MDPs using value iteration, by making approximations to the value function that are contingent on their assumptions on the domain. These assumptions include reward and transition functions in the model being *piecewise constant* (PWC), while being continuous. The approximation relies on the property that the convolution of two PWC functions, results in a *piecewise linear* (PWL) function. Hence, the new value function computed at step $n + 1$ using the one at step n is PWL. However, using this to compute the value function at time $n + 2$ yields a piecewise quadratic function, and the order keeps increasing as the horizon for computation expands. To retain tractability of the solution, after every value iteration step, the PWL value function is approximated by a PWC function and is used to compute the value function for the next stage. The approximation is carried out by minimizing the L_∞ error, which is used to bound the error between the value function returned by the algorithm and the optimal value function according to Singh and Yee [15]. Lazy approximation provides a solution to performing value iteration for multi-dimensional continuous state variables by avoiding discretization. They validate their method on synthetic data as well as the well-known Mars Rover problem, where the optimal policy for the rover to cover all the targets and take pictures, minimizing time and energy consumed.

Another paper that deals with finding continuous states is [3] , which attempts to find exact solutions for MDPs with time as a continuous state variable. They label these processes as *time dependent* MDPs or t-MDPs. The contribution of this paper is to provide a way to compute *exact* solutions for MDPs with a continuous state space. The focus on the continuous variable being time is to solve specific examples where continuous time variable was part of the state space. The assumptions made by the model are that the transition function needs to be discrete and finite and that reward functions are expected to be piecewise linear (PWL). The biggest contribution of this paper is to compute a *time-value* function exactly and solve for the optimal strategy without making any approximations, for specific situations as mentioned above.

In [4], Feng et al also present an approach to approximately solve MDPs with continuous variables as part of their state space. The goal of this paper is to exploit the structure of the problem to use dynamic programming to efficiently solve for the optimal solution. These structural assumptions made by the authors of this paper are that the reward function must be PWC or PWL and more importantly that the transition function must be discrete and finite. Particularly, the assumption regarding the reward structure is that they are convex and *rectangular* PWC or PWL, which means that within a hyper-rectangular partition of the state space they are either constant or linear. The paper then makes use of properties of the Bellman backup that arise from these assumptions to efficiently perform value iteration and show a significant reduction in computational time when applied the Mars Rover example, as opposed to the discretization method, when the conditions are satisfied.

One paper that does attempt to remove a large number of the restrictions seen so far on the reward and transition functions, and provide a general approach to solving any continuous/hybrid state MDP is by Sanner et al [14]. They propose a *Symbolic* Dynamic Programming based algorithm that uses a continuous version extension of Algebraic Decision Diagrams (ADDs), which is a data structure used to compactly represent the value function. The paper presents a general method to solve continuous state without making assumptions. However, the use of a general algorithm that makes use of decision diagrams may result in significant computational complexity and can be avoided by making assumptions about the domain.

In [13], Munos and Moore deal with continuous variable MDPs by presenting an intelligent discretization scheme for the value function with respect to the continuous state, essentially discretizing based on whether more resolution is required in a particular region or not. They used an approach based on the k-D tree data structure that is used to partition a k-dimensional space. The algorithm starts with a uniform and coarse discretization of the value function, partitioning into a grid of rectangular cells. Each cell is then split using the Kuhn's triangulation technique based on the criteria that check whether the corner values of the value functions are all constant or whether the value function is linear within a cell. Their method is evaluated based on both splitting criteria for a particular problem called the ”Car on the Hill” problem. The goal is to identify an optimal policy for the car to reach the top of the hill as soon as possible and stop there. The method has the merit of overcoming the computational inefficiency of the uniform discretization method and uses domain information to increase resolution in regions of high variation. However, this algorithm makes the assumption of the transition function being both discrete and finite. The finiteness of the transition function is an essential assumption to reduce error considerably, as a transition function that allows the transition to an infinite number of states with non-zero probability will lead to an error where the value function is truncated.

In most of the aforementioned work, to solve MDPs with continuous state variables, there has been some assumptions made about the domain being modelled to increase the efficiency of the value function. Some papers model a larger class of problems at the cost of computational or space complexity. In this thesis, we take the former approach, i.e., using domain knowledge to make assumptions about the modelled domain to be able to perform value iteration better than with the naive uniform discretization of the continuous state variables. We also allow for discrete variables to be present in the state in addition to the continuous variables. Our approach models those domains where the transition function has infinite support but makes use of analytical solutions for the value functions at the extrema to avoid truncation errors from approximation. In regions where it is computationally intractable to obtain the analytical solution, we perform non-uniform discretization based on the transition function to approximate the region. As in some of the above work, there have been assumptions like the piece-wise linearity of immediate reward that have also been made in our modeling. In chapter 3, the method is described in detail.

1.3 Outline

This thesis is structured in the following way. Chapter 2 includes an overview of the particular task we will study, the delayed gratification task, and the rationale behind the proposed model for human behaviour. The chapter is based largely on the work from [12] and presents the formalization of the delayed gratification task as a Markov decision problem, which we will refer to as the DGMDP. In Chapter 3, solutions for the proposed DGMDPs are presented that use a modified value iteration algorithm. Finally, in Chapter 4, simulation results are presented that examine the relationship between model parameters and model predictions. In the concluding chapter, a list of future paths for this work is laid out to improve upon the contributions of this thesis.

2 Delayed Gratification

In the 1970s, Walter Mischel and his colleagues at Stanford University conducted a series of experiments [11] on children to study their ability to delay gratification, using snacks like marshmallows and cookies. The setup for the experiment was to give each child one marshmallow and ask them to wait for a period of time while the experimenter stepped out of the room, after which upon their return they would receive a second treat if the first one was left uneaten. This required children to overcome the temptation of the single marshmallow sitting in front of them while the experimenter was absent so that they don't forfeit their chance of eating two marshmallows, a bigger reward, instead of just the one. Of the children who initially chose to wait for the second marshmallow, many succumbed to the temptation of the one marshmallow at different points in their wait time. A fraction of the children succeeded in waiting for all the way until the experimenter returned.

Figure 3: Picture showing child at different stages of the Marshmallow experiment; Source: Google Images

This failure to delay gratification isn't just observed in children who like marshmallows but is a common phenomenon in human decision making. For e.g., Should you eat the piece of rich chocolate cake, or stick to your diet? Or should you spend money impulse-shopping or save that amount towards your retirement? Tasks which require gratification to be delayed, span a variety of domains from maintaining a healthy lifestyle to financial decision making. In all these cases, it is in the individual's best interest to wait for the large reward as opposed to opting for instant gratification. However, a study of behaviour in these situations reveal that humans are not always disposed to making the obvious *rational* choice, and tend to revert to choosing a small, immediate reward even if they initially chose to wait.

There are models of delayed gratification like [10] that explain the defection rate of the agents via uncertainty in the time horizon and time-fluctuating discount rates. Here, we explain the failure to delay gratification in environments where the horizon is well known, as in real-world tasks like retirement savings. As seen in Figure 3, the children in the marshmallow task need to make a consistent effort to resist the temptation of the marshmallow at any time instant. The child's behavior reflects not just making a decision to wait at the start of the task, but a constant struggle between persisting and defecting during the delay period. The model presented in the following sections aims to capture such characteristics of human agents performing this task.

2.1 Formalizing The One Shot Delayed Gratification Task

In this section, we focus on modelling and representing our delayed gratification task as a Markov Decision problem ($DGMDP$). The assumption made here is that time is quantized into discrete steps at which decisions are made by the agent. Assuming the number of time steps to the large reward is τ, the subject makes a decision at every time step to either $PERSIST$, thereby being one step closer to the large reward, or $DEFECT$ in favour of immediately receiving the smaller reward. Figure 4., represents the finite state machine for the $DGMDP$.

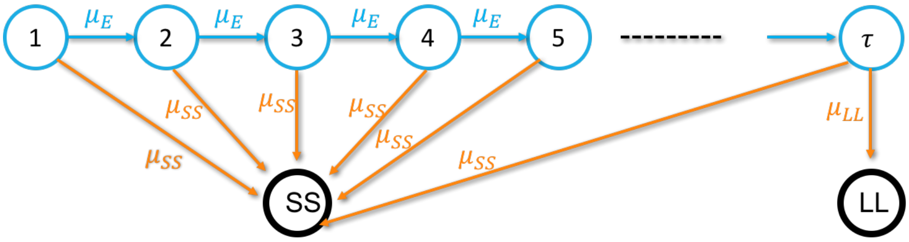


Figure 4: Finite state machine representation of delayed gratification MDP

As noted in Figure 4, the smaller and larger rewards are denoted μ_{SS} and μ_{LL}, respectively. SS and LL are shorthand for 'smaller sooner' and 'larger later'. In addition, there is also a constant reward available for continuing at any time step, μ_E, which can be used to represent some cognitive cost of waiting for the reward at every step. These rewards are associated with the state transitions. With exponential discounting, rewards t steps away are diminished by a factor γ^t, where $\gamma \in [0, 1)$. Using backward induction technique of value iteration the above $DGMDP$ can be solved as follows:

$$V(\tau) = \max_a \begin{cases} \mu_{SS} & \text{for a} = DEFECT \\ \mu_{LL} & \text{for a} = PERSIST \end{cases} \tag{5}$$

$$V(\tau - 1) = \max_a \begin{cases} \mu_{SS} & \text{for a} = DEFECT \\ \mu_E + \gamma \mu_{LL} & \text{for a} = PERSIST \end{cases} \tag{6}$$

In general, at any time step t we have,

$$V(t) = \max_a \begin{cases} \mu_{SS} & \text{for a} = DEFECT \\ \frac{\gamma^{\tau-t}-1}{\gamma-1}\mu_E + \gamma^{\tau-t}\mu_{LL} & \text{for a} = PERSIST \end{cases} \tag{7}$$

According to the above solution, the optimal strategy is highly dependent on the value of the discount factor, γ. Depending on γ, the optimal strategy is either to *DEFECT* at the very first time step or continue all the way to time step τ. This is because, either the discounting factor is so low, that the value of the discounted final reward is less than the value of the immediate reward for defecting. If this is not the case, then at no subsequent time step will the discounted value of the final reward be less than μ_{SS}, as the value of this discounted reward only grows as the reward is approached. Though this formalization leads to obtaining solutions analytically with ease, it happens to be a poor characterization of human behaviour. As can be recalled from the discussions in the above section, it is often observed that humans who choose to defect in a delayed gratification task, can do so by defecting after a certain time, even though their initial decision is to continue.

As alluded to earlier, to enable the use of Markov Decision Processes to model this problem, the perspective of *bounded rationality* on human cognition is used, where human behaviour can be considered optimal, subject to cognitive constraints. These cognitive constraints are introduced into the model by augmenting the state space and changing the reward functions. They are described below:

1. Individuals exhibit moment-to-moment fluctuations in *willpower* based on factors like current mood, hunger, etc. Willpower modulates the rewards in such a way that makes immediate rewards more tempting when willpower is low, and less tempting when it is high:

$$R(t, w; DEFECT) = \mu_{SS} - w_t \tag{8}$$

We consider willpower at any time step is correlated to the willpower at the previous time step and evolves according to a random walk process, with $w_t \sim \text{Gaussian}(w_{t-1}, \sigma^2)$. The state space now consists not only of the discrete time step t, but also the continuous variable w that represents the individuals' willpower level.

2. In the behavioural and neuroeconomics literature, its is suggested that effort carries a cost, that will also additively change the value of the rewards [7]. This is incorporated into the model by means of an effort cost, μ_E which is the immediate reward for waiting at any time step prior to the final time step, at which μ_{LL} is available, as shown below:

$$R(t, w; PERSIST) = \begin{cases} \mu_E & \text{for } t < \tau \\ \mu_{LL} & \text{for } t = \tau \end{cases} \tag{9}$$

The two cognitive constraints mentioned above are used to augment our model to sufficiently explain human behaviour. In subsequent sections and chapters we present algorithms that each provide reasonable fits to human data while overcoming the challenges of using a hybrid discrete-continuous state space, an uncontrolled Markov chain as well as non-finite transition function.

2.2 Formalizing The Iterated Delayed Gratification Task

To test our model against empirical results from human subjects, we need to design and run experiments that emulate the delayed gratification task. The marshmallow test is a nice experimental demonstration, but it is very difficult to collect such data because each participant can sit through the task only one time, and collect adequate amounts of data to the model would require hundreds of experimental participants. Further, individuals may defect in the marshmallow task simply because they care more about their time than the larger-later outcome. For this reason, we designed a variation on the one-shot delayed gratification task which is performed iteratively. This modification helps us collect enough data from one individual and hence from the population. It also removes the problems of accounting for unintended incentives by creating a fixed time task for the individual, whose rewards will now solely depend on the immediate and delayed incentives offered by the experiment. The experiment is designed as a web application and deployed on Amazon's Mechanical Turk platform. In this section, we briefly describe the setup and the finite state machine representation of the experiment that we choose to model.

We describe the iterative task as follows. The subject is required to choose between a large reward and immediately choosing a small reward. Upon receiving either the large or small reward, the individual is faced with a new delayed gratification task with the same setup. Each subsequent task that the individual is faced with is referred to as a *episode*. In each episode, the task can be thought of as a choice between waiting in a long queue to get a reward when you reach the front, or at any point to leave the long queue and claim a small but less appealing reward. The task is carried out for a fixed amount of clock time eliminating the incentive to defect in order to terminate the experiment.

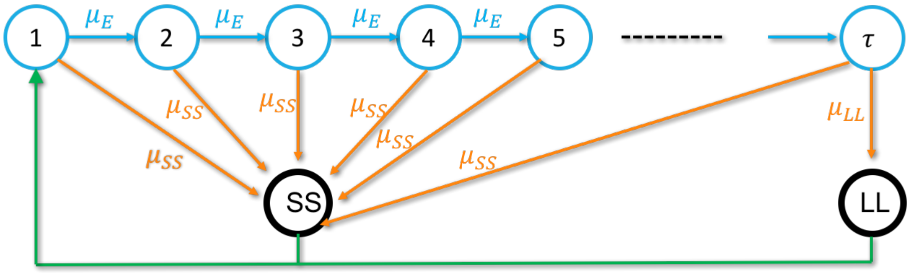


Figure 5: Finite state machine representation of the iterative *DGMDP*

Figure 5 shows the finite state machine representation for this simple iterative task. Though Figures 4 and 5 appear to be very similar, the green arrow that represents a repetition of episodes implies that finite-horizon planning isn't sufficient for such a setup. The green arrow is different from the others, as it doesn't represent a state transition, but a *short-circuit* between states **SS/LL** and **1** (first time step of next episode), i.e., being is **SS** or **LL** is as good as being in state **1** if the next episode.

We can generalize this task by varying the length of the long queue from episode to episode, to determine the dependency of the model as well as behaviour on the waiting time involved to obtain the large reward. The length of the longer line for an episode was chosen uniformly at random at the beginning of that episode.

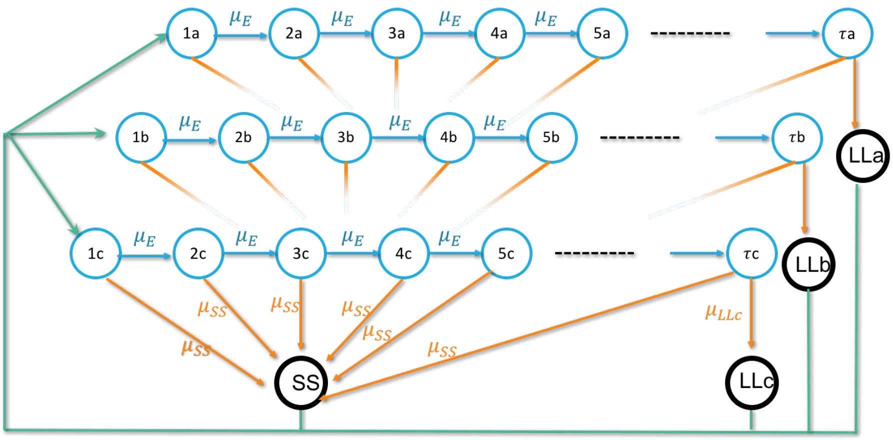


Figure 6: Finite state machine representation of the iterative *DGMDP* with multpile conditions

Figure 6 represents the experiment that was deployed on Amazon Mechanical Turk on subjects. As in the one-shot DGMDP presented in Section 2.1, the agent is assumed to be performing optimal planning subject to the two bounding constraints proposed earlier: (1) the moment-to-moment fluctuations in willpower and (2) the cost of effort it takes to persist towards the large reward. The figure shows several episodes stacked parallelly, with the faded arrows pointing to a *SS* node corresponding to its own episode.

The agent is also assumed to discount future rewards exponentially. The choice for assuming the agents temporal discounting as exponential is motivated by its compatibility to the MDP framework, which assumes exponential discounting for computational convenience. Although in the human intertemporal choice literature, there is evidence that humans and animals discount future rewards *hyperbolically* rather than exponentially [5], Kurth-Nelson and Redish [8] propose a solution where the hyperbolic function can be well approximated by a mixture of exponentials. Techniques used to solve MDPs, which have exponential discounting, can be extended to solve MDPs with a mixture of exponential discounting. Values can be propagated for different discounting rates (with a different level of decay), and can later be combined to determine the overall value function. Here, the proposed computational model can be readily extended in the same way to approximate hyperbolic discounting behaviour.

To summarize, we have formalized the one-shot and iterative delayed-gratification task with known horizon, as a Markov decision problem with parameters $\Theta_{task} \equiv \{\tau, \mu_{SS}, \mu_{LL}\}$ and a constrained rational agent parameterized by agent $\Theta_{agent} \equiv \{\gamma, \sigma, \beta, \theta, \mu_E\}$.

2.3 Experiment Design

To collect empirical data from human subjects in a delayed gratification task, a web-based application was built to simulate the experience of waiting in a long line for the long reward vs obtaining a small reward immediately, in repeated episodes. Participants were recruited on Amazon Mechanical Turk and were compensated for their time. The setup is described in more detail in [12] and in this section, the important details are summarized.

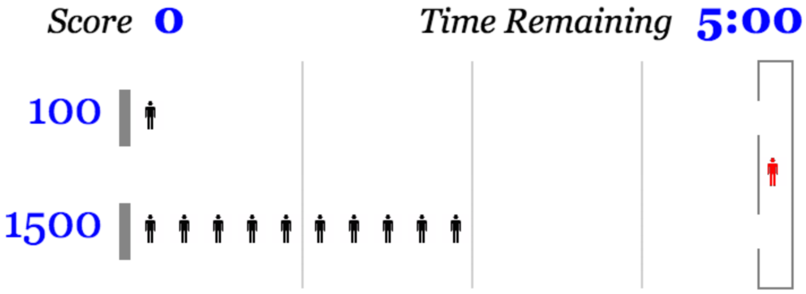


Figure 7: The queue-waiting game; The player (red icon) is in the vestibule, prior to choosing a queue. Queues advance right to left. Points awarded per queue are displayed left of the queue.

The board in Figure 7 represents the game board of the Mechanical Turk task. The upper queue is the short queue where the player will have to wait behind only one individual before they are *serviced*, gaining 100 points in reward. The lower queue is the long queue where the subject has to wait behind τ "people", before being serviced and rewarded with $100\tau\rho$ points. Here ρ represents the reward-rate ratio between the longer and shorter line. The game is updated at 2000 msec interval, at which the player's request is processed and the queue advances from left to right, when a "person" (stick-figure in black) at the front of both lines is serviced. The player starts every episode in the *vestibule* on the right, and has to select whether to enter the long or short line using the *up* or *down* arrow key. When the player (in red) enters the short line, they are immediately serviced, and when the long line is chosen, the player moves into the next-to-last position of the long line. This ensures that the wait to the reward involves exactly τ time steps. When in the long line, the player needs to hit the left arrow key at every time step to advance. If the player doesn't choose to advance and doesn't take any action, the person behind the player jumps ahead. At any position in the long line, the player can choose to defect to the short line upon which they are immediately serviced. Finally, when the reward is earned by the player, the points are flashed on the screen in big font, and the score on the top left corner is incremented by the corresponding number of points. A *cash-register* sound is also played, and a new episode begins, where the red player once again finds themselves in the vestibule on the right. In our experiment, the value of τ is uniformly drawn from the set $\{4, 6, 8, 10, 12, 14\}$ at the beginning of each episode.

Participants were paid \$0.80 for their time and were awarded a score-based bonus. They were required to perform at least one action every ten seconds, or else the experiment was terminated and their data rejected. In our analyses of behaviour, the first and last thirty seconds of data from their play was removed. At the beginning, the players were just getting accustomed to the game and hence weren't sure of the optimal strategy. Towards the end, it might have been optimal to just defect more since there wasn't sufficient time to wait in the long line and gain the rewards.

3 Solving The Markov Decision Problem

In this chapter, we will address the main contribution of this thesis, which is to present an efficient algorithm to solve the Markov decision problem described in chapter 2. First, we present an approximation that allows us to analytically solve for the optimal strategy, for the one-shot DGMDP (Figure 4). We then proceed to describe an algorithm to solve the more general, iterated MDP (Figure 6). For clarity, we list the attributes of the MDP in question, i.e., we formalize the DGMDP.

1. **State** $s = (t, w)$—a 2-tuple representing "time" t with respect to the length of the horizon, and the agent's willpower w. t is a discrete variable that doesn't represent the absolute time, but the time position in an episode relative to the end of the episode. w is a continuous variable that represents the individual's moment-to-moment fluctuation in willpower.

2. **Action** $a \in \{\mathit{DEFECT}, \mathit{PERSIST}\}$—binary action state where at every time step, the subject has the option to either *PERSIST* or *DEFECT*.

3. **Reward**—immediate reward dependent on the state and action variables

$$R(t, w; a) = \begin{cases} \mu_{SS} - w & \text{for } a = \mathit{DEFECT} \\ \mu_E & \text{for } a = \mathit{PERSIST}, t < \tau \\ \mu_{LL} & \text{for } a = \mathit{PERSIST}, t = \tau \end{cases} \tag{10}$$

   The reward for defecting in any given episode at any time step is modulated additively by the current willpower value of the individual. For extremely low willpower values $w \to -\infty$, the immediate reward for defecting is very high, which is likely to make it optimal for the individual to quit. For persisting, at any time step other than the final one, it is assumed that the agent incurs a cognitive cost for persisting, which we refer to as the *effort* cost. Therefore, μ_E is a negative value and a quantity that parametrizes the agent. At the final time step in an episode, if the agent decides to persist, they get the large reward μ_{LL}.

4. **Transition**

   willpower , w:

$$w' = \beta w + \theta + \sigma\epsilon \tag{11}$$

$$\epsilon \sim \text{Gaussian}(0, 1)$$

This is the most general form of the transition function used. The values of β and θ are varied to to obtain different transition models for the different DGMDPs.

time, t:

$$\begin{align} P(t+1|t, PERSIST) &= 1 \quad \text{for } t < \tau \tag{12} \\ P(t+1|t, DEFECT) &= 0 \quad \text{for } t < \tau \tag{13} \\ P(SS|t, PERSIST) &= 0 \quad \forall t \tag{14} \\ P(SS|t, DEFECT) &= 1 \quad \forall t \tag{15} \\ P(LL|t, PERSIST) &= 1 \quad \text{for } t = \tau \tag{16} \end{align}$$

5. **Discount Rate** , $\gamma \in [0, 1)$—exponentially diminishes the value of the future rewards. For infinite horizon problems, γ limits the planning horizon so that the objective doesn't go to infinity. In this case, γ is also a parameter that represents how myopic a human decision maker with respect to future rewards.

3.1 Solving The One Shot Delayed Gratification Task

We first present the one-shot case of the DGMDP in which the agent is presented with one decision task in which gratification is delayed (Figure 4). This task is akin to the marshmallow experiment described in Chapter 2. We solve this problem analytically using a piecewise linear approximation of the value function.

We consider willpower at any time step is correlated to the willpower at the previous time step and evolves according to a random walk process, with $w_t \sim \text{Gaussian}(w_{t-1}, \sigma^2)$. The transition function in this one-shot case for the value function is,

$$w_t = w_{t-1} + \sigma\epsilon \tag{17}$$

which is Equation (11) with $\beta = 1$ and $\theta = 0$. Here we use a simpler model for the transition of willpower because there are a finite number of steps for planning. Though the random walk model is not stationary, the resulting distribution of willpower can be characterized in terms of the number of steps to the end and the distribution of the noise added at each step. Therefore, behaviour can be characterized in the one-shot task without using β and θ. The initial willpower of the agent at step 1, $w_1 \sim \text{Gaussian}(0, \sigma_1^2)$.

Because of the relatively simple structure of the one-shot MDP, backward induction solution of the Bellman equation can be used for this environment to solve for the optimal solution. Here, we introduce an auxiliary function $Q(s; a)$ which is the state-action value function that defines the value associated with a state-action pair, regardless of whether the action a is optimal, with subsequent actions determined by the optimal policy. The optimal value function is therefore represented as follows:

$$\begin{aligned} V(t, w) &= \max_a \Big\{ Q(t, w; a) \Big\} \\ Q(t, w; a) &= R(t, w; a) + \gamma \max_a \Big\{ \mathbb{E}_{W_{t+1}|W_t = w} Q(t+1, W_{t+1}; a) \Big\} \end{aligned} \tag{18}$$

where, $a \in \Big\{ DEFECT, PERSIST \Big\}$. To solve the MDP using the backward induction method, we start by computing the value functions from the final time step, i.e., the time step closest to the large reward.

$$\begin{aligned} Q(\tau, w; DEFECT) &= \mu_{SS} - w_\tau \\ Q(\tau, w; PERSIST) &= \mu_{LL} \\ \therefore V(\tau, w) &= \begin{cases} \mu_{SS} - w_\tau, & \text{for } w_\tau < w_\tau^* \\ \mu_{LL} & \text{otherwise} \end{cases} \end{aligned} \tag{19}$$

In Equation (19), $w_\tau^* = \mu_{SS} - \mu_{LL}$ represents the threshold willpower value at which the two actions are valued equally.

At $\tau - 1$, the value function backup yields the following results:

$$\begin{aligned} Q(\tau-1, w; DEFECT) &= \mu_{SS} - w_{\tau-1} \\ Q(\tau-1, w; PERSIST) &= \mu_E + \gamma \Bigg[\Phi\bigg(\frac{w_\tau - w_{\tau-1}}{\sigma} \bigg) \mu_{SS} + \sigma \phi \bigg(\frac{w_\tau - w_{\tau-1}}{\sigma} \bigg) + \\ &\qquad \bigg(1 - \Phi\bigg(\frac{w_\tau - w_{\tau-1}}{\sigma} \bigg) \bigg) \mu_{LL} \Bigg] \end{aligned}$$

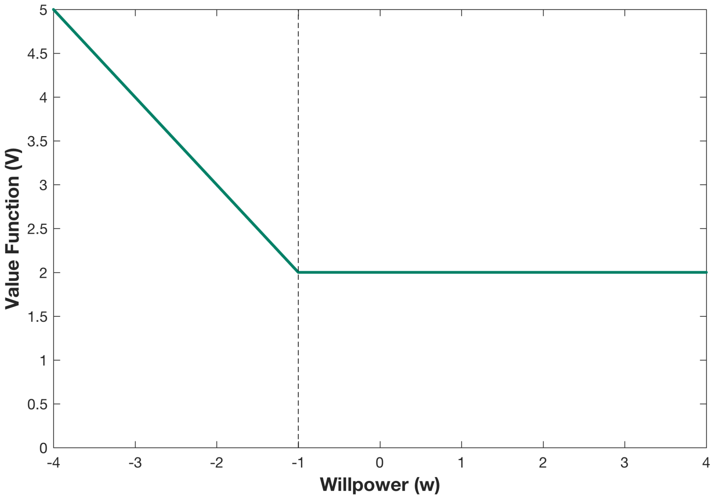


Figure 8: Value function at time step τ generated for $\mu_{SS} = 1$ and $\mu_{LL} = 2$; critical willpower $w^*_\tau = -1$

$$\therefore V(\tau-1, w) = \begin{cases} \mu_{SS} - w_{\tau-1}, & \text{for } w_{\tau-1} < w^*_{\tau-1} \\ \mu_E + \gamma\left[\Phi\left(\frac{w_\tau - w_{\tau-1}}{\sigma}\right)\mu_{SS} + \sigma\phi\left(\frac{w_\tau - w_{\tau-1}}{\sigma}\right) + \right. & \\ \left. \qquad \left(1 - \Phi\left(\frac{w_\tau - w_{\tau-1}}{\sigma}\right)\right)\mu_{LL}\right] & \text{otherwise} \end{cases} \tag{20}$$

where, $\Phi(.)$ and $\phi(.)$ are the cdf and pdf of the standard normal distribution, respectively. From Equation (20), we see that $w^*_{\tau-1}$ cannot be obtained analytically like w^*_τ. However, we can make a claim about the qualitative properties of the value function with respect to the state variable w[1]:

$$Q(\tau-1, w; PERSIST) \approx \mu_E + \gamma\big(\mu_{SS} - w_{\tau-1}\big) \qquad \text{for } w \to -\infty \tag{21}$$

$$Q(\tau-1, w; PERSIST) \approx \mu_E + \gamma\mu_{LL} \qquad \text{for } w \to +\infty \tag{22}$$

Based on Equation 21 we can see that as $w \to -\infty$, $\mu_{SS} - w$ grows faster than $\mu_E + \gamma\mu_{SS} - \gamma w$, since $\gamma < 1$. Therefore as $w \to -\infty$, we have $V(\tau-1, w) = Q(\tau-1, w; DEFECT)$. Similarly, from Equation (22), we see that as $w \to +\infty$, $\mu_{SS} - w \to -\infty$ which is less than $\mu_E + \gamma\mu_{LL}$. Therefore, for $w \to +\infty$, we have $V(\tau-1, w) = Q(\tau-1, w; PERSIST)$. The optimal value

[1] $\because$ for $w \to -\infty; \Phi\big(\frac{w'-w}{\sigma}\big) \approx 1$ & $\Phi\big(\frac{w'-w}{\sigma}\big) \approx w$
$\because$ for $w \to +\infty; \Phi\big(\frac{w'-w}{\sigma}\big) \approx 0$

functions at the extrema are as follows:

$$V(\tau-1,w)=\begin{cases}\mu_{SS}-w_{\tau-1} & \text{for } w_{\tau-1}\to-\infty\\ \\ \mu_E+\gamma\mu_{LL} & \text{for } w_{\tau-1}\to+\infty\end{cases} \tag{23}$$

We now have an understanding regarding the shape of the value function at step $\tau-1$, with a linear region with a negative slope for low values of w and a constant value being approached for higher values rendering a "hockey-stick" shape. Intuitively, we can expect such qualitative features for the value function at any time t. The equation for the linear region for $w_t < w_t^*$ is the same for all the t, since the reward for the defecting doesn't change. For $w_t \to \infty$, the value of the constant the curve approaches probably depends on t, as further behind in time, the large reward gets discounted more, and higher effort cost is incurred. Using the analytical expression for the value function $V(\tau-1,w)$, we can verify the claims made in Equations (21) and (22) qualitatively:

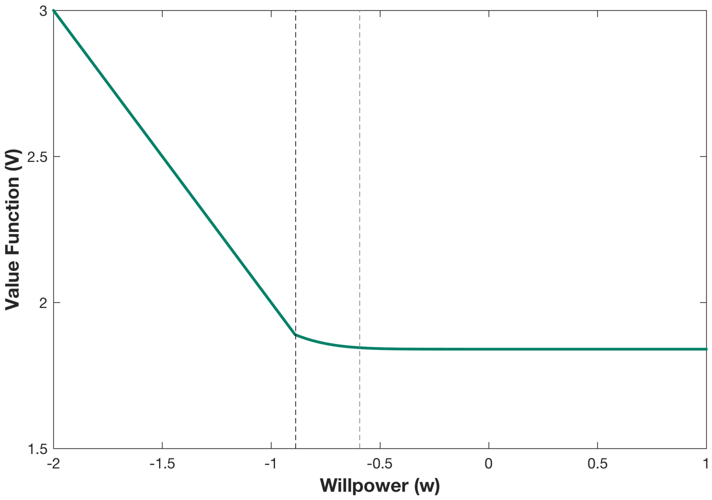


Figure 9: Value function generated for position $\tau-1$ analytically, for $\tau=8$ and $\mu_{SS}=1$ and $\mu_{LL}=2$. The black dashed lines indicate $w^*_{\tau-1}$ to the left of which we value function which is linear for low values of willpower. To the right of the grey dashed line we see that the value function approaches a constant. Between the two lines is the intermediate willpower values

To quantify the above claims regarding the shape of the value function, we present a method that solves the MDP using a piecewise linear approximation (PLA) for the value function.

The formal steps involved in performing value iteration using the PLA is as follows,

1. While propagating the value function from time step $t+1$ to time step t, we assume that the value function for the next time step has a k-PLA form and then compute the

expectation.

2. We then approximate the expectation by a 3-PLA curve, using any non-linear least squares method like the *Levenberg-Marquadt* method.
3. Once, we have the 3-PLA solution for the expectation, we perform the *max* operation analytically (also shown below), resulting in a k-PLA functional form, where $k = \{2, 3, 4\}$, depending on where the two curves intersect.
4. We then iterate this operation back to time step $t = 1$.

As listed in step 2 above, we need to analytically compute the expectation given a k-PLA form. Here, we present a general solution to find the expectation of any k-PLA function $J(w)$ over a Gaussian distributed random variable w, with mean μ and standard deviation σ with pdf represented by $\phi\big(\frac{w-\mu}{\sigma}\big)$, where $\phi(.)$ is the pdf of a standard normal distribution. Let the slope and intercept of each of the linear regions be represented by a_i^t, b_i^t, where $i = 1, ...k$ and let the end points of each of these linear segments be represented by α_j^t, where $j = 1, ...k+1$. For the sake of convenience, the superscript t is dropped in the following discussions, but the expressions described hold for all t.

$$\begin{aligned}
\therefore, \mathbb{E}[J(w)] &= \int_{-\infty}^{\infty} J(w)\phi\Big(\frac{w-\mu}{\sigma}\Big)dw \\
&= \int_{\alpha_1}^{\alpha_{k+1}} J(w)\phi\Big(\frac{w-\mu}{\sigma}\Big)dw \\
&= \sum_{i=1}^{k} \int_{\alpha_i}^{\alpha_{i+1}} \{a_i w + b_i\}\phi\Big(\frac{w-\mu}{\sigma}\Big)dw \\
&= \sum_{i=1}^{k} \Bigg[(a_i w + b_i)\Big[\Phi\Big(\frac{\alpha_{i+1}-\mu_{i+1}}{\sigma}\Big) - \Phi\Big(\frac{\alpha_i - \mu_i}{\sigma}\Big)\Big] \\
&\quad - a_i\sigma\Big[\phi\Big(\frac{\alpha_{i+1}-\mu_{i+1}}{\sigma}\Big) - \phi\Big(\frac{\alpha_i-\mu_i}{\sigma}\Big)\Big]\Bigg] \qquad (24)
\end{aligned}$$

Equation (24) gives us an expression to compute the expectation for any piecewise linear form of the value function. Based on our intuition from Equations (21) and (22), we know the the slope and intercept values for two out of the three linear regions in our k-PLA. If region 1 represents the left extreme, i.e., value function for low willpower, and region k represents the value function for high willpowers, then we have:

$$a_1 = -1 \qquad b_1 = \mu_{SS} \qquad a_k = 0 \qquad b_k = c_t \tag{25}$$

where, c_t is the value of steadfast persistence. At every backward induction step, it is computed as $c_{t-1} = \mu_E + \gamma c_t$, with $c_\tau = \mu_{LL}$. The final expression from c_t is therefore:

$$c_t = \frac{\gamma^{\tau-t} - 1}{\gamma - 1}\mu_E + \gamma^{\tau-t}\mu_{LL} \tag{26}$$

and values for α_i, α_{i+1}, μ_i and μ_{i+1} in Equation (24) are expressed as:

$$\alpha_{i+1} = \frac{\mu_{SS} - b_t}{a_t + 1}; \quad \mu_{i+1} = w(a_t + 1) \tag{27}$$

$$\alpha_i = \frac{c_t - b_t}{a_t}; \quad \mu_i = wa_t \tag{28}$$

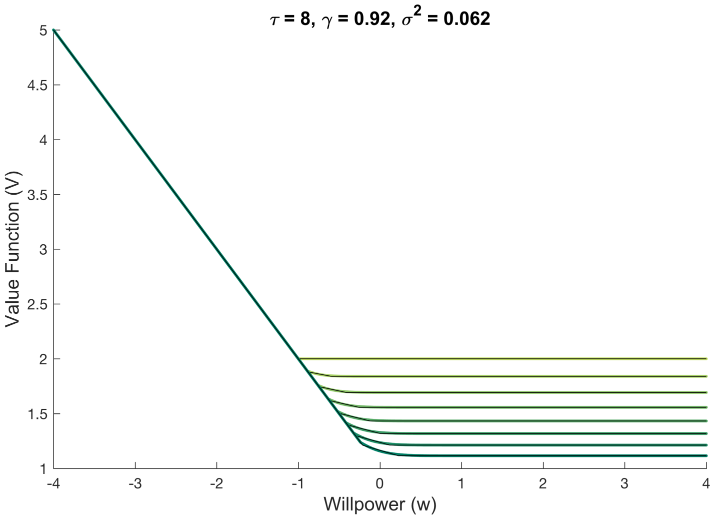


Figure 10: Value function generated using the modified value iteration for $\tau = 8$, $\mu_{SS} = 1$ and $\mu_{LL} = 2$. The colored lines are the exact computation of the value function plotted for different values of t from $t = 8$ (light yellow curve) to $t = 1$ (dark green curve).

In Figure 10, we generate the value function curves based on the k-PLA scheme. The graphs are generated for a task with $\tau = 8$ steps to getting the large reward. The value of the small and large rewards are $\mu_{SS} = 1$ and $\mu_{LL} = 2$, respectively. The colored curves represent the value function computed analytically, and the superimposed black curves represent the corresponding 3-PLA approximation at every time step. The lightest curve represents the value function at position 8 and the darkest represents position 1.

3.2 Solving The Iterated Delayed Gratification Task

In this section, we present methods to solve the iterated version of the DGMDP, represented in Figure 6. To recap, in this finite state machine, the agent is initially faced with a task in which a large reward is available at the end of a queue, in which he/she has to wait. At any of the positions in the long line, the agent is free to defect to a short queue in which a smaller reward is available immediately. Upon receiving a reward (larger later or smaller sooner), the agent is again at the beginning of the new episode for the same task, with possibly a new line length for the longer line. The goal here is to solve this iterated DGMDP using a modified approach to value iteration. For the iterated DGMDP, we have the following transition function for willpower:

$$w_{next} = \beta w_{curr} + \theta + \sigma\epsilon \tag{29}$$

$$\beta < 1 \quad \theta \neq 0$$

This model for willpower transition in Equation (29) is the autoregressive process model of order 1 ($AR(1)$), which is different from the random walk model defined in Section 3.1. The $AR(1)$ model has has a stationary distribution with the following mean (θ_0) and standard deviation (σ_0) [6]:

$$\theta_0 = \frac{\theta}{1-\beta}; \qquad \sigma_0^2 = \frac{\sigma^2}{1-\beta^2} \tag{30}$$

In the iterated case, we move from the simple random walk model to the $AR(1)$ model for willpower transition. The $AR(1)$ process is wide-sense stationary resulting in a theory of willpower which is stationary. The use of a stationary model is convenient mathematically and can also be used to better explain some cognitive phenomena. For example, we can characterize individuals as having high or low mean willpower (θ_0). Without the decay parameter β, a distribution of willpower would be dependent on which episode the individual is in, i.e., it would be *age*-dependent.

Before presenting the efficient value iteration method to solve the iterated DGMDP (Figure 6), we present results from a naive solution based on completely discretizing the continuous state and computing the pointwise value function:

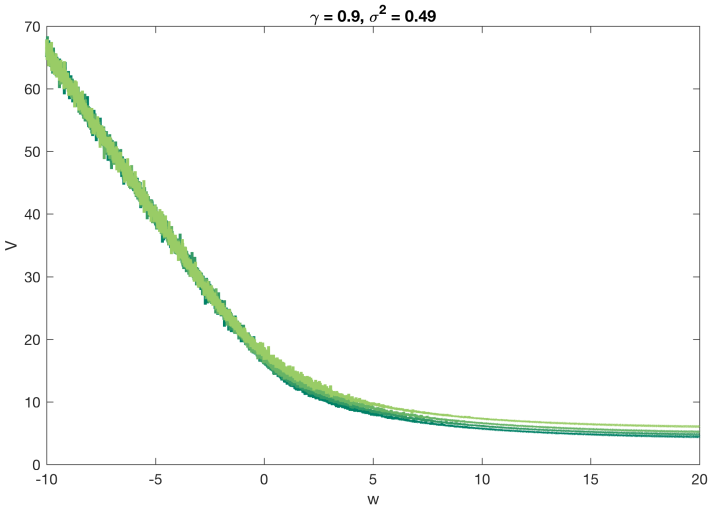


Figure 11: Value function generated using the discretization method with $\tau = 4$ and resolution of $10,000$ points; the colored lines are the exact computation of the value function plotted for different values of t from $t = 4$ (light green curve) to $t = 1$ (dark green curve)

Figure 11 plots the value function with respect to w for different positions from the reward, t, using the discretization method. The discretization is performed in accordance with the transition probability distribution, i.e., willpower points are sampled based on the distribution, as opposed to a uniform discretization scheme. To reduce approximation error, the resolution of discretization can be increased, but this leads to a polynomial increase in computation time. Another problem associated with this method is its incompatibility with non-finite state spaces. To perform discretization, a finite region of the w is considered, thereby truncating the variable and introducing error while computing the value function.

In Figure 11, we see that some of the qualitative attributes of the value function have been preserved from the one-shot DGMDP case (Figure 10). For instance, for smaller w values ($w \to -\infty$), the curve appears to retain a linear shape, and for $w \to \infty$, a constant value is approached. This indicates that an analytical can be computed for the value function, at the extrema.

Let us analyze what it means to the agent to have willpower that lies in these extrema, to obtain our analytical expression. From the agent's perspective, when their willpower is extremely low, it is optimal to defect repeatedly in consecutive episodes. Using the value iteration algorithm from Figure 2, we can perform a series of value iteration steps to understand the estimation of $V_i(t, w)$, the value function in the i^{th} iteration for position t and willpower w

as follows:

$$
\begin{aligned}
V_0(1,w) &= 0 && (31)\\
V_1(1,w) &= R(t,w) + \gamma \mathbb{E}_{w_{next}}[V_0(1,w_{next})]\\
&= (\mu_{SS} - w) + \gamma \int_{w_{next}} 0 dw_{next}^{-}\\
&= \mu_{SS} - w && (32)\\
V_2(1,w) &= R(t,w) + \gamma \mathbb{E}_{w_{next}}[V_1(1,w_{next})]\\
&= (\mu_{SS} - w) + \gamma \int_{w_{next}} (\mu_{SS} - w_{next}) dw_{next} && (33)\\
V_2(1,w) &= (\mu_{SS} - w) + \gamma \int_{-\infty}^{\infty} (\mu_{SS} - w_{next}) dw_{next}\\
&= (\mu_{SS} - w) + \gamma[\mu_{SS} - (\theta + \beta w)]\\
&= (1+\gamma)\mu_{SS} - (1+\beta\gamma)w - \gamma\theta && (34)
\end{aligned}
$$

Repeated defection lands the agent at state 1 at subsequent episodes and hence, V is computed only for $t = 1$. w_{next} and w are related according to Equation (29). Based on the above steps of value iteration we can present an analytical solution for the value function at any iteration for $w \to -\infty$:

$$V_n(1,w) = \frac{1-\gamma^n}{1-\gamma}\mu_{SS} - \frac{1-(\beta\gamma)^n}{1-(\beta\gamma)} w - \sum_{i=1}^{n-1} \left[\gamma^i \frac{1-(\gamma\beta)^i}{1-(\gamma\beta)} \right] \tag{35}$$

Once the algorithm converges, the optimal value function $V_\infty(1,w)$, for $w \to -\infty$ is:

$$V_\infty(1,w) = \frac{1}{1-\gamma}\mu_{SS} - \frac{1}{1-\beta\gamma} w - \frac{\gamma}{(1-\gamma)(1-\gamma^2\beta)} \tag{36}$$

Similarly, for the other extremum of the willpower space ($w \to \infty$), we can prescribe a method to compute the value function. When the willpower of the agent is extremely high, the optimal policy would be to persist to the end of the long line in every episode. As in the previous case, value function is initialized to zero. We solve for this “piece” of the value function inductively instead of analytically as follows:

$$V_0(t,w) = 0$$

$$V_1(t,w) \equiv c_1(t) = \begin{cases} \mu_E \quad \text{if } t \neq \tau \\ \mu_{LL} \quad \text{if } t = \tau \end{cases}$$

$$V_n(t,w) \equiv c_n(t) = \begin{cases} \mu_E + \gamma\big[c_{n-1}(t+1)\big] & \text{if } t \neq \tau \\ \mu_{LL} + \gamma\, \mathbb{E}_{t_{next}}\big[c_{n-1}(t_{next})\big] & \text{if } t = \tau \end{cases} \tag{37}$$

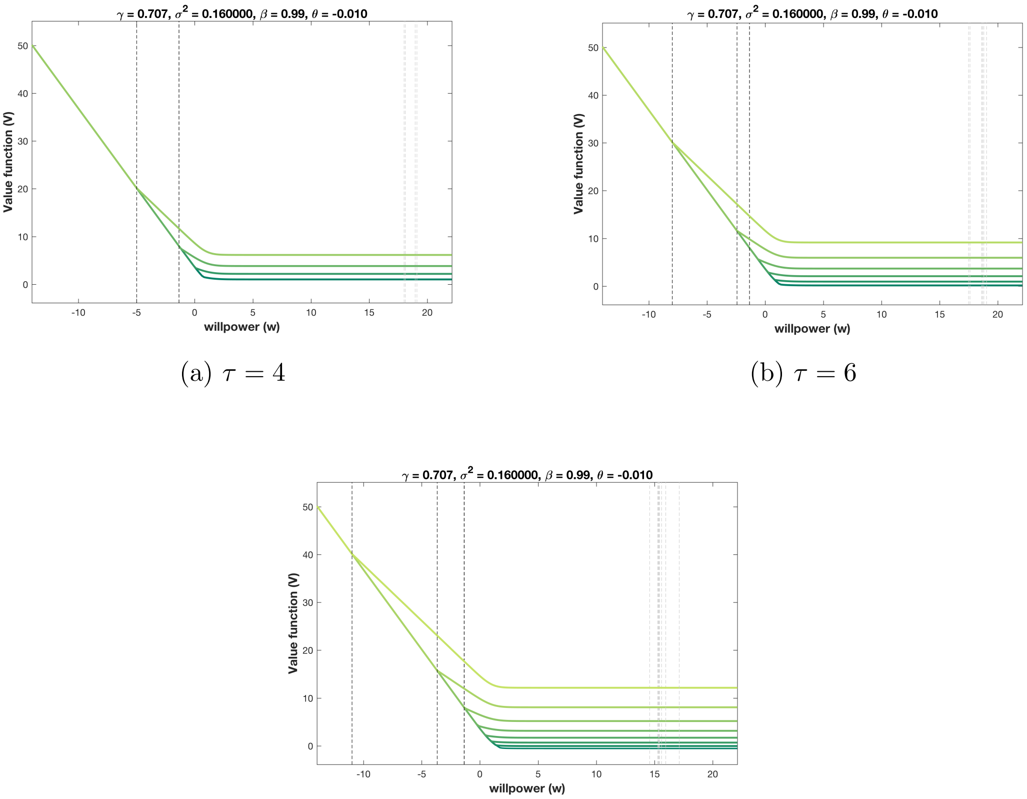
Therefore, the value function on the right extrema is computed by updating $c_n(t)$ at every iteration n, according to Equation (37). This is analogous to Equation (26) from the one-shot DGMDP, where the agent persists all the way to the end of the horizon.

We now look at performing value iteration efficiently by leveraging the analytical expression and update rule in Equations (35) and (37) respectively. The basic idea is as follows: for large willpower values, we can approximate the value function by the constant corresponding to the value of the $w \to \infty$ extremum. For small willpower values, we can approximate the value function as if the agent is always defecting. The remaining willpower region in between is considered to be piecewise constant. This region is bounded by two values which we call $w_{Bmin}(t)$ and $w_{Bmax}(t)$, on the left and right respectively. For willpower values less than $w_{Bmin}(t)$, the value function is defined by the analytical expression in Equation (35) and for values greater than $w_{Bmax}(t)$, it is computed according to Equation (37). Between $w_{Bmin}(t)$ and $w_{Bmax}(t)$, the value function is computed at each of the discrete points using the regular value iteration update equation:

$$V_i(t,w) = \max_a \Big\{ R(t,w;a) + \gamma \mathbb{E}_{t_{next},w_{next}}\big[V_{i-1}(t_{next}, w_{next})\big]\Big\} \tag{38}$$

To compute the expectation over willpower, space is sampled according to the transition distribution for possible values at the next step and the pointwise value function is summed up and divided by the number of samples. The values lying below the next time step's $w_{Bmin}(t)$ and $w_{Bmax}(t)$ values take on the analytical and inductive expressions. For points in between, the value function at the given points takes on takes on the value computed for the piecewise constant region to which it belongs. For the expectation over time, the average is computed for the value functions from next possible line length conditions that the agent will see.

The $w_{Bmin}(t)$ and $w_{Bmax}(t)$ values are initialized to $-\infty$. At iteration 1, the value of both these variables is equivalent to where the linear region (Equation (35)) and the constant region (Equation(37)) meet. This is because at the first iteration the value function is composed of only these two regions. After that, the values are updated to allow the middle region to grow. This update is based on the transition function for willpower. A set of points are generated according to the cumulative distribution function transition probability $Gaussian(\beta * w + \theta, \sigma^2)$, where w represents the current willpower value. After computing, the value function at all the points between $w_{Bmin}(t)$ and $w_{Bmax}(t)$, the pointwise value function at the extremes is compared to the analytical expressions that they approach on either side. If on any side, the pointwise estimate of the value function is within some tolerance of the value computed analytically, the values of $w_{Bmin}(t)$ or $w_{Bmax}(t)$ (depending on which extremum) are moved inwards to shrink the region that relies on the pointwise approximation.



(a) $\tau = 4$

(b) $\tau = 6$

(c) $\tau = 8$

Figure 12: Value function curves generated using the modified approximation of value iteration with three line length conditions appearing in subsequent episodes in random order. The curves represent the value function for different τ and with respect to w. The black dashed lines on the left indicate the $w_{Bmin}(t)$ values of willpower below which the curve is approximated by the linear region. The lighter dashed lines on the right are $w_{Bmax}(t)$ above which the value function is approximated by a constant

Figure 12 shows the value function computed according to the modified value iteration approach. The value function curves obtained from this method are much less noisy than the ones from the fully discretized approach. The more yellow curves represent value function at time step closer to the reward whereas the more green curves represent value further away. The qualitative features of these value functions also match our intuition for the shape for the one-shot and iterative case. The values of w_{Bmin} and w_{Bmax} for different t are annotated by the black and grey dashed lines, respectively.

3.3 Hazard Function To Characterize Behaviour

In order to effectively characterize behaviour using our model and meaningfully compare it to the empirical evidence from human subjects, we compute the *hazard* probability. The hazard probability or hazard rate is a quantity from the survival analysis literature that is used to represent the probability of 'death'—which refers to the occurrence of the event of interest—given survival or lack of occurrence of such an event upto that point. In our case, the event of interest would be the agent defecting to obtain the immediate reward. Therefore, the hazard probability in our situation would be the probability that the agent will defect at any given time, having continued or waited for the large reward until such time. . If D denotes the time step of defection, then the hazard probability is represented as follows:

$$h_t \equiv P(D = t | D \geq t) \equiv P(W_t < w_t^* | W_1 \geq w_1^* ..., W_{t-1} \geq w_{t-1}^*) \tag{39}$$

where w^* is the willpower threshold that yields action indifference, i.e., $Q(t, w; DEFECT) = Q(t, w; PERSIST)$. The posterior distribution for willpower at each non-defection step is represented using quantile based samples [12].

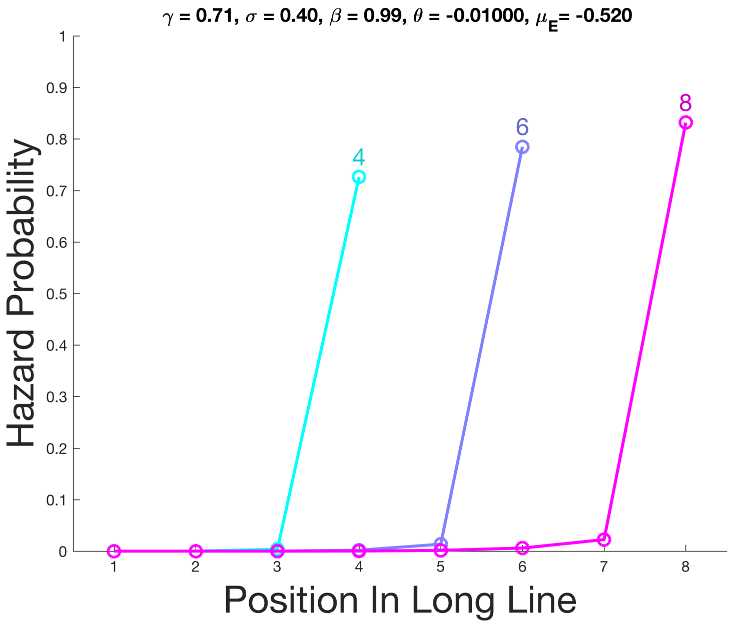


Figure 13: Hazard curves generated for iterated DGMDP with line lengths $\{4, 6, 8\}$ with respect to line position. Line position of τ represents the farthest position from the reward, i.e, time step 1.

Figure 13 shows the hazard curves for iterated DGMDP with line lengths $\{4, 6, 8\}$ with respect to the line position. Line position of τ corresponds to the time step 1, which is the back of the queue and farthest from the reward. The graph shows that the hazard rate at position τ is the highest, whereas closest to the reward at position one, the hazard rate drops to zero. As the agent gets closer to the large reward, the model predicts that they are less likely to defect in favour of the small reward. To generate the hazard curves, only one episode is simulated for the iterated case, since the policy obtained is stationary with respect to episodes. The prior at step 1, W_1 and the added noise term $(\sigma\epsilon)$ are approximated with discrete, equal probability $q-$quantiles. We reject willpower values that lie below the willpower threshold, as those corresponding to the act of defecting, and propagate the rest forward based on the $AR(1)$ equation for the iterated case and the random walk equation for the one-shot case. The forward propagation at each step results in q^2 samples, which is thinned back to $q-$quantiles at each step. The value of q used is 1000 which produces results identical to higher values.

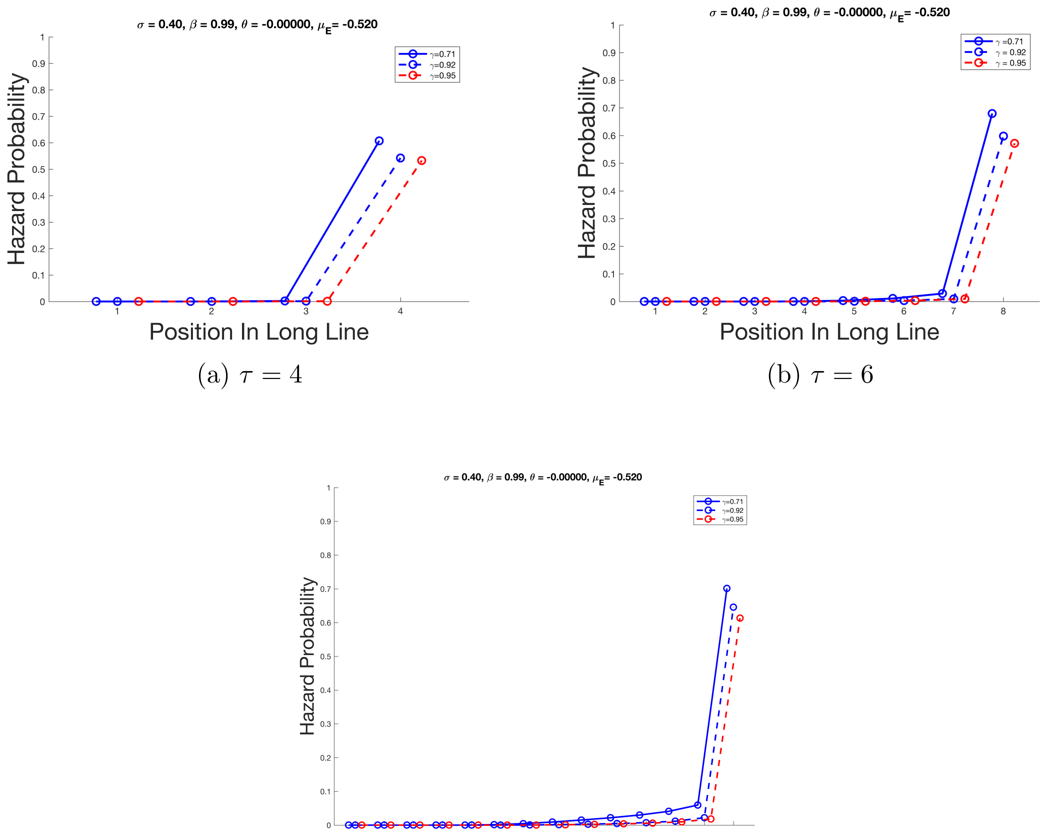
4 Model Simulation And Parameter Exploration

In this chapter, we characterize the optimal policy by means of the hazard function mentioned in Section 3.3. First, the model output is characterized as a function of parameters in Θ_{agent}, Θ_{task}.

4.1 Characterizing Behaviour With Respect To Agent Parameters

First, we show the effect of the change in different parameters of the agents Θ_{agent} on the hazard rates and characterize their effects on human behavior, according to our model.

1. Discount rate γ:



(a) $\tau = 4$

(b) $\tau = 6$

(c) $\tau = 14$

Figure 14: Hazard curves plotted for different line lengths τ with varying γ: 0.71 (solid blue curve), 0.91 (dashed blue line) and 0.95 (dashed red line); hazard curves plotted for the same line length are staggered for clarity while superimposing the different graphs

In Figure 14, we have plotted the hazard curves for three different line lengths 4, 8 and 14. Each of the curves in these plots was generated with repeated episodes with only one line length condition. The values of γ chosen were values with half-lives from $n_{HL} = 2, 8, 14$, i.e., after n steps, γ drops to 0.5. The final reward in all three cases, at the end of the episode, was equivalent to $\mu_{LL} = 1.5 * \mu_{SS} * \tau$.

In Figure 14a,

$\tau = 4 \quad \mu_{LL} = 6$

At position 4 which is farthest from the reward, the value of the final reward is diminished as follows for various γ values:

$$\text{for } \gamma = 0.7071, n_{HL} = 2, \mu_{LL}^{discounted} = 1.52$$

$$\text{for } \gamma = 0.9170, n_{HL} = 8, \mu_{LL}^{discounted} = 4.24$$

$$\text{for } \gamma = 0.9517, n_{HL} = 14, \mu_{LL}^{discounted} = 4.92$$

Hence, we see a much higher defect rate for $\gamma = 0.71$ as opposed to $\gamma = 0.91$ and $\gamma = 0.95$, farther away from the reward. Also, we see that the defect rates for $\gamma = 0.91$ and $\gamma = 0.95$ are very similar.

In Figure 14b,

$\tau = 8 \quad \mu_{LL} = 12$

At position 4 which is farthest from the reward, the value of the final reward is diminished as follows for various γ values:

$$\text{for } \gamma = 0.7071, n_{HL} = 2, \mu_{LL}^{discounted} = 0.75$$

$$\text{for } \gamma = 0.9170, n_{HL} = 8, \mu_{LL}^{discounted} = 6.00$$

$$\text{for } \gamma = 0.9517, n_{HL} = 14, \mu_{LL}^{discounted} = 8.07$$

Here also, we see a much higher defect rate for $\gamma = 0.71$ as opposed to $\gamma = 0.91$ and $\gamma = 0.95$, farther away from the reward. However, we start to see a slight difference in defect rate for $\gamma = 0.91$ and $\gamma = 0.95$ as can be reflected in the differing values in the $\mu_{LL}^{discounted}$ values between the two. However, they're closer for these two values of γ than they are to $\gamma = 0.71$.

In Figure 14c,

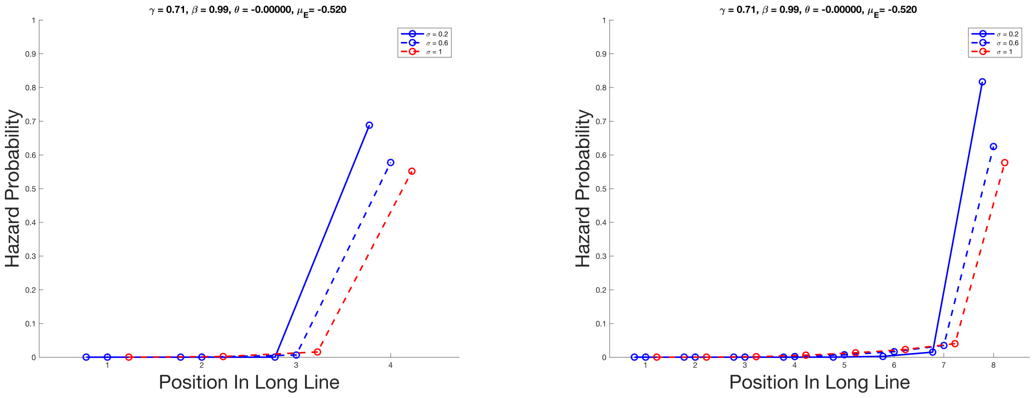
$\tau = 14 \quad \mu_{LL} = 21$

At position 4 which is farthest from the reward, the value of the final reward is diminished as follows for various γ values:

$$\text{for } \gamma = 0.7071, n_{HL} = 2, \mu_{LL}^{discounted} = 0.16$$
$$\text{for } \gamma = 0.9170, n_{HL} = 8, \mu_{LL}^{discounted} = 6.24$$
$$\text{for } \gamma = 0.9517, n_{HL} = 14, \mu_{LL}^{discounted} = 10.5$$

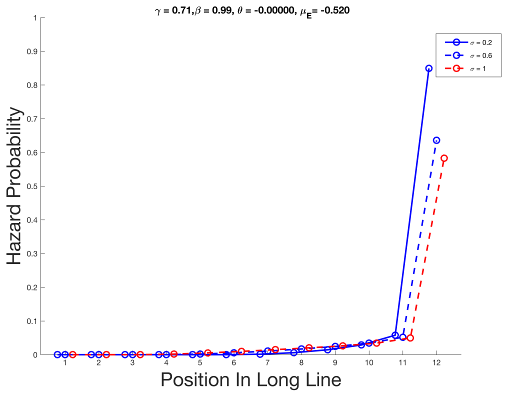
Once again, we see a much higher defect rate for $\gamma = 0.71$ as opposed to $\gamma = 0.91$ and $\gamma = 0.95$ at line positions farther away from the large reward. As expected, the difference in defect rates for $\gamma = 0.91$ and $\gamma = 0.95$ are more pronounced than in the previous two cases since we see a bigger difference between the $\mu_{LL}^{discounted}$ values. Farthest from the reward at position $\tau = 14$, the reward value is diminished to less than 0.5 for $\gamma = 0.71$, explaining the high defection rate at this position.

2. Variance of noise σ^2 This represents the variance of the noise added at every step of the $AR(1)$ process.



(a) $\tau = 4$

(b) $\tau = 8$



(c) $\tau = 12$

Figure 15: Hazard curves plotted for different line lengths τ with varying σ: 0.2 (solid blue curve), 0.6 (dashed blue line) and 1 (dashed red line); hazard curves plotted for the same line length are staggered for clarity while superimposing the different graphs

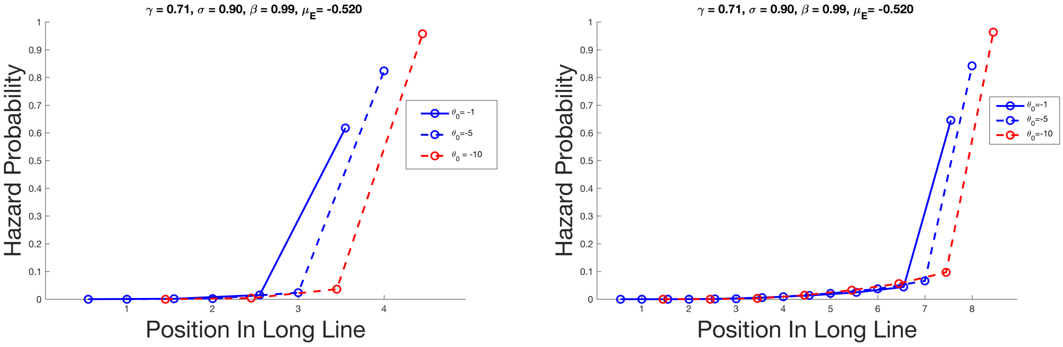
In Figure 15, we can see that with high variance, though the initial defect rate is high, the hazard rate drops almost to zero for any of the intermediate time steps showing that the behaviour is more deterministic with small variance values. With large variance values, the initial defect rate drops, but the subsequent defect rate also increases, making for similarity in graphs irrespective of Θ_{task} parameters for these values.

3. The AR(1) stationary mean, θ_0:

This parameter represents the mean of the stationary distribution of the AR(1) transition model. The mean is related to the drift rate parameter from Equation (11) as follows:

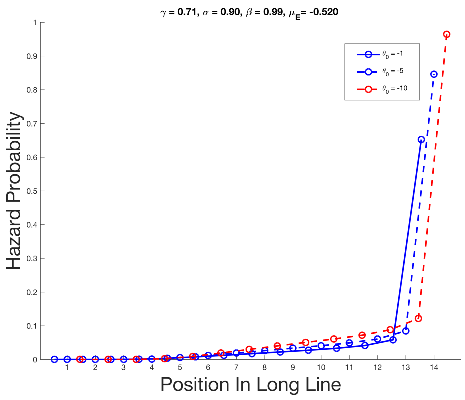
$$\theta_0 = \frac{\theta}{1-\beta}$$

where β represents the decay parameter of the AR(1) process.



(a) $\tau = 4$

(b) $\tau = 8$

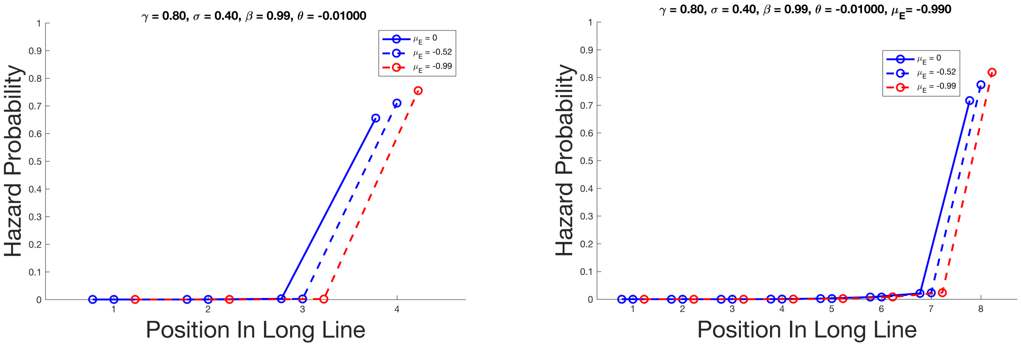


(c) $\tau = 14$

Figure 16: Hazard curves plotted for different line lengths τ with varying θ_0: -1 (solid blue curve), -5 (dashed blue line) and -10 (dashed red line); hazard curves plotted for the same line length are staggered for clarity while superimposing the different graphs

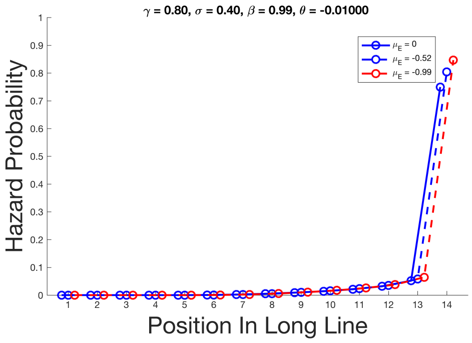
The variation of hazard curves is shown with respect to different values of the stationary mean. As the mean becomes bigger and more negative, the hazard rate for all the positions increases. This happens for different values of τ. This intuitively is reasonable since, a negative mean implies that the willpower reverts back to negative values making it likely to fall below the threshold or indifference point, hence increasing the hazard rate.

4. Effort Cost μ_E: The effort cost is the cost incurred by the agent for persisting towards the large reward. μ_E is a negative quantity since it represents a cost, but has to be cast as a reward to make it compatible with the MDP framework considered here. The following are some of the hazard curves, plotted for different values of effort cost.



(a) $\tau = 4$

(b) $\tau = 8$



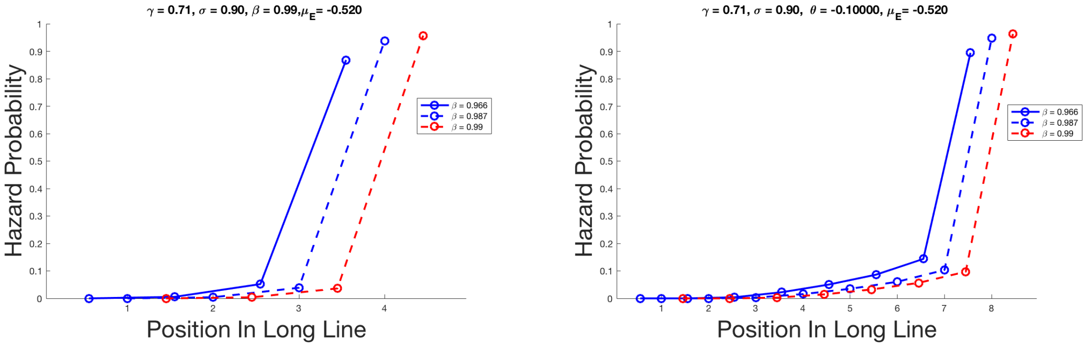
(c) $\tau = 14$

Figure 17: Hazard curves plotted for different line lengths τ with varying μ_E: 0 (solid blue curve), -0.52 (dashed blue line) and -0.99 (dashed red line); hazard curves plotted for the same line length are staggered for clarity while superimposing the different graphs

From Figure 17 we can characterize the hazard curves dependence on effort cost as follows: the higher the effort cost (in magnitude), the higher the defection rate, especially at the end of the line. This seems like a reasonable prediction—if the agent perceives the wait to be more effortful, then they are less likely to persist towards the large reward in the long line.

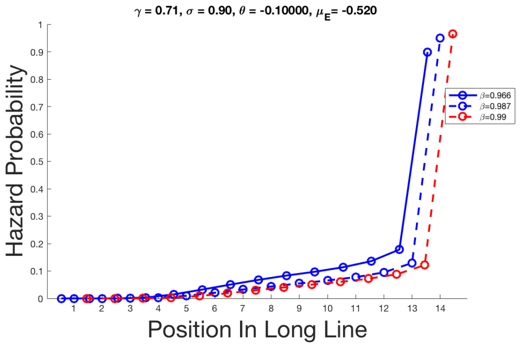
5. Decay parameter β

The decay parameter in the $AR(1)$ process indicates how much correlation there is between willpower is between successive stages.



(a) $\tau = 4$

(b) $\tau = 8$



(c) $\tau = 14$

Figure 18: Hazard curves plotted for different line lengths τ with varying β: 0.966 (solid blue curve), 0.987 (dashed blue line) and 0.99 (dashed red line); hazard curves plotted for the same line length are staggered for clarity while superimposing the different graphs τ

Figure 18 shows the variation in hazard value with different value of decay rate β for different line length values.

4.2 Validation Using Synthetic Data

To validate our approach, our model was fit to synthetically generated hazard curves to verify whether the parameters used to generate the graphs could be inferred. The results are as follows:

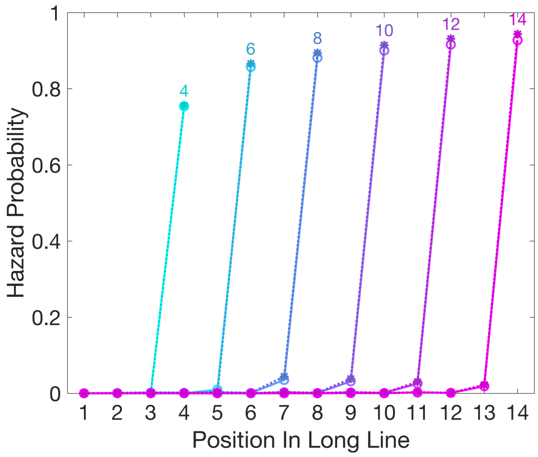


Figure 19: Hazard curves synthetically generated are represented by the dotted lines. The model hazard curves are the solid lines.

The parameters used in the generative model and the recovered parameters to generate the hazard curves in Figure 19 are as follows:

Parameter	Generated	Inferred
γ	0.9568	0.9599
σ^2	0.0454	0.0406
β	0.99	0.99
θ_0	-0.01	-0.008
μ_E	-0.5208	-0.5416

Table 1: Parameters of the generative model and the inferred parameters presented side-by-side for comparison for line lengths $\{4, 6, 8, 10, 12, 14\}$

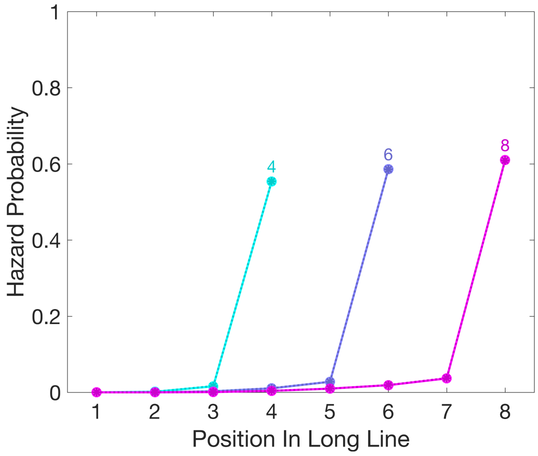


Figure 20: Hazard curve synthetically generated are represented by the dotted lines. The model hazard curves are the solid lines.

The parameters used in the generative model and the recovered parameters to generate the hazard curves in Figure 20 are as follows:

Parameter	Generated	Inferred
γ	0.7071	0.7058
σ^2	0.8100	0.8898
β	0.99	0.99
θ_0	-0.00100	-0.00039
μ_E	-0.5200	-0.5708

Table 2: Parameters of the generative model and the inferred parameters presented side-by-side for comparison for line lengths $\{4, 6, 8\}$

4.3 Predicting Human Behaviour

To justify the modeling approach for delayed gratification, the model hazard curves are fit to empirical hazard curves obtained from human data. The model free parameters Θ_{agent} are varied to obtain the best match between the two sets of curves. The following hazard curve shows fits to human data as well as the parameters returned from the optimization procedure:

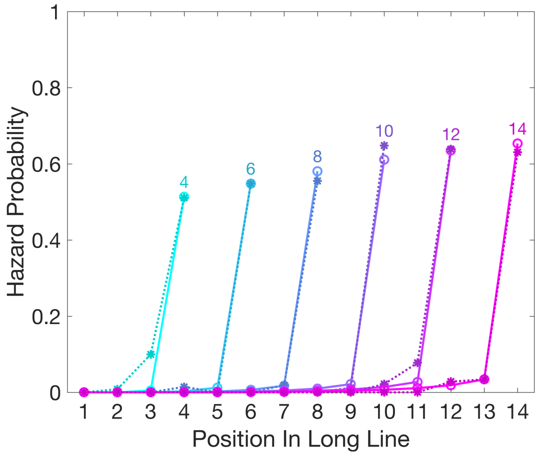


Figure 21: Fits from the model to the human data. The model hazard curves are the solid lines and the human hazard are the dotted lines.

As we see in Figure 21, there are five line length conditions which are repeatedly presented to the subject for a fixed period of 5 minutes in random order. The payoff at the end of the long line, $\mu_{LL} = 1.5 * \mu_{SS} * \tau$, in a given episode. μ_{SS} is the payoff associated with the short line and is set at 100. The model was used to simulate the hazard curves with Θ_{task} being set correspondingly to the experiment. The parameters that were optimized over were Θ_{agent}.

The least-squares *post hoc* fits for the five agent parameters are as follows:

Parameter	Inferred
γ	0.7665
σ^2	48.03
β	0.99
θ_0	-0.05
μ_E	-5.87

Table 3: Parameters of the generative model and the inferred parameters presented side-by-side for comparison for line lengths $\{4, 6, 8\}$

5 Conclusion And Future Work

In this thesis, we discussed a number of issues associated with solving Markov decision problems with continuous state variables, as well as solutions to address these issues. All simulations and results were presented in the context of a computational model of human behaviour in delayed gratification tasks. Specifically our contributions include:

1. Presenting an efficient value iteration based algorithm to solve a specific type of MDPs that contain a continuous and discrete state component, and the continuous component evolves according to a stochastic process unrelated to the agent's actions. The method leverages domain information to improve efficiency and reduce approximation error.

2. Presenting an analytical approximation to solve for the optimal policy for a specific case of the delayed gratification MDP, when the task is performed just once.

Hazard curves were generated to characterize behaviour as predicted by the model. The model predictions were justified based on fits to synthetic as well as empirical hazard curves. The fits to the empirical hazard curves provide some validation for the modelling approach.

The modelling is approach can be applied to a fairly restrictive set of decision problems. Some assumptions like the piecewise linearity of rewards can be expanded to include a wider class of reward structures to allow for modelling a larger set of decision problems. As stated before, this work has been presented in the context of a psychological computational model to explain behaviour. The application of this method to problems in optimal control is yet to be explored.

Situations where the method we have developed might be useful include any decision environment in which planning is required by the agent whose actions do not affect a particular variable, for example, a solar powered automatic lawnmower that needs to plan how to cover as much area as it can, but has no control over the weather or how much sunlight it will receive. This is a situation to which our method can be applied after making similar, but with reasonable assumptions made about the reward and transition structures.

The long term goal of this project is to use this computational model of behaviour to generate incentives to human decision makers to stick to long term goals [12]. The model can easily be augmented to include bonus structures by adding time-dependent rewards for persisting in the long line, in addition to the effort cost. Once an effective incentive structure is obtained, the model can be used to determine the improvement in behaviour.

References

[1] Richard Bellman. *Dynamic Programming.* Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

[2] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 2nd edition, 2000.

[3] Justin A Boyan and Michael L Littman. Exact solutions to time-dependent mdps. In *NIPS*, pages 1026–1032, 2000.

[4] Zhengzhu Feng, Richard Dearden, Nicolas Meuleau, and Richard Washington. Dynamic programming for structured continuous markov decision problems. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 154–161. AUAI Press, 2004.

[5] Shane Frederick, George Loewenstein, and Ted O'donoghue. Time discounting and time preference: A critical review. *Journal of economic literature*, 40(2):351–401, 2002.

[6] James Douglas Hamilton. *Time series analysis*, volume 2. Princeton university press Princeton, 1994.

[7] Ran Kivetz. The effects of effort and intrinsic motivation on risky choice. *Marketing Science*, 22(4):477–502, 2003.

[8] Zeb Kurth-Nelson and A David Redish. A reinforcement learning model of precommitment in decision making. *Frontiers in behavioral neuroscience*, 4:184, 2010.

[9] Lihong Li and Michael L. Littman. Lazy approximation for solving continuous finite-horizon mdps. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 3*, AAAI'05, pages 1175–1180. AAAI Press, 2005.

[10] Joseph T McGuire and Joseph W Kable. Rational temporal predictions can underlie apparent failures to delay gratification. *Psychological review*, 120(2):395, 2013.

[11] Walter Mischel, Yuichi Shoda, and Monica L Rodriguez. Delay of gratification in children. *Science*, 244(4907):933, 1989.

[12] Michael C Mozer, Shruthi Sukumar, Camden Elliott-Williams, Shabnam Hakimi, and Adrian F Ward. Overcoming temptation: Incentive design for intertemporal choice.

[13] Rémi Munos and Andrew Moore. Variable resolution discretization in optimal control. *Machine learning*, 49(2):291–323, 2002.

[14] Scott Sanner, Karina Valdivia Delgado, and Leliane Nunes De Barros. Symbolic dynamic programming for discrete and continuous state mdps. *arXiv preprint arXiv:1202.3762*, 2012.

[15] Satinder P. Singh and Richard C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.